

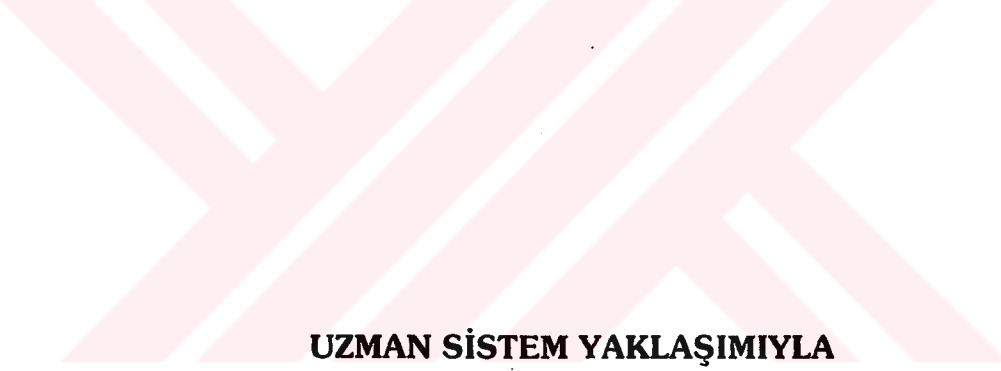
57128

T.C.  
SELÇUK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

UZMAN SİSTEM  
YAKLAŞIMIYLA  
PRİZMATİK PARÇALAR İÇİN  
İŞLEM PLANLAMA SİSTEMİ

Süleyman YALDIZ  
DOKTORA TEZİ  
MAKİNA MÜHENDİSLİĞİ  
ANABİLİM DALI  
KONYA, 1996

**T.C.  
SELÇUK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**



**UZMAN SİSTEM YAKLAŞIMIYLA  
PRİZMATİK PARÇALAR İÇİN  
İŞLEM PLANLAMA SİSTEMİ**

**Süleyman YALDIZ  
DOKTORA TEZİ**

**MAKİNA MÜHENDİSLİĞİ  
ANABİLİM DALI**

**KONYA, 1996**

T.C.  
SELÇUK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

UZMAN SİSTEM YAKLAŞIMIYLA  
PRİZMATİK PARÇALAR İÇİN  
İŞLEM PLANLAMA SİSTEMİ

Süleyman YALDIZ

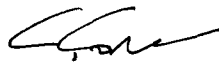
DOKTORA TEZİ

MAKİNA MÜHENDİSLİĞİ  
ANABİLİM DALI

Bu tez 20 /12/1996 tarihinde aşağıdaki jüri tarafından 100. (...402...) not takdir edilerek oybirliği/oyçokluğu ile kabul edilmiştir.



Prof. Dr. Ali ÜNÜVAR  
(Danışman)



Doç. Dr. M. Cemal ÇAKIR  
(Üye)



Doç. Dr. Ahmet AVCI  
(Üye)

## ÖZET

### Doktora Tezi

## UZMAN SİSTEM YAKLAŞIMIYLA PRİZMATİK PARÇALAR İÇİN İŞLEM PLANLAMA SİSTEMİ

**Süleyman YALDIZ**  
**Selçuk Üniversitesi**  
**Fen Bilimleri Enstitüsü**  
**Makina Mühendisliği Anabilim Dalı**

**Danışman : Prof. Dr. Ali ÜNÜVAR**

**1996, Sayfa : 170**

**Jüri : Prof. Dr. Ali ÜNÜVAR**

**Doç. Dr. M. Cemal ÇAKIR**

**Doç. Dr. Ahmet AVCI**

Uzmanlar, endüstride tasarım bilgileri ve imalat tekniklerini birlikte organize ederek bir ürünü meydana getirmek için yıllardır çalışmaktadırlar. Geleneksel olarak bir ürünü tasarım özelliklerine göre üretmek için bu alanda tecrübeli bir kişiye duyulan ihtiyaç halen devam etmektedir. Bununla beraber, teknolojiye üretim tekniklerinde meydana gelen hızlı gelişmeler, bu tür imalat problemlerini çözmede tecrübeli bir insana bağımlı olmaktan kurtulmayı gerektirmektedir. Çünkü, herhangi bir alanda bir kişinin uzmanlaşması bir hayli zaman almaktadır. Son yıllarda tasarımın ürüne dönüşmesinde planlayıcı olarak, bir insan yerine, aynı görevleri daha kısa zamanda ve optimum şartlarda yerine getirmekte kullanılan çok değişik yazılımlar geliştirilmiştir. Bu çalışmada prizmatik parçaların işlem planlarını otomatik olarak yerine getirmek üzere, uzman sistem yaklaşımı kullanan üretken bir Bilgisayar Destekli İşlem Planlama (BDİP) sistemi geliştirilmiştir.



Sistemde üç boyutlu (3B) parça verilerini temsil etmek için Sınır Temsiline (ST) dayalı Ürün Model Verileri Değişimi için STEP standardı kullanılmıştır. Sistem stok seçimi, taslak ve bitmiş parça tasarımı, referans yüzeyi seçimi, operasyon seçimi ve sıralama, bağlama ve araparça oluşturma modüllerinden meydana gelir. Sistemin çıktısı ise, çalışmada ele alınan örnek parçalar için işlem planlama, kesici takım-takım tezgahı seçimi ve bağlama işlemi kartlarından oluşur.

**Anahtar Kelimeler** : Bilgisayar Destekli İşlem Planlama , Uzman Sistem Yaklaşımı, Özellik Tabanlı Tasarım, STEP Standard



## **ABSTRACT**

**PhD Thesis**

### **PROCESS PLANNING SYSTEM FOR PRISMATIC PARTS BY EXPERT SYSTEM APPROACH**

**Süleyman YALDIZ**

**Selçuk University**

**Graduate School of Natural and Applied Sciences**

**Department of Mechanical Engineering**

**Supervisor : Prof. Dr. Ali ÜNÜVAR**

**1996, Page : 170**

**Jury : Prof. Dr. Ali ÜNÜVAR**

**Assoc. Prof. Dr. M. Cemal ÇAKIR**

**Assoc. Prof. Dr. Ahmet AVCI**

The experts have been trying to produce a product for many years by organizing the industrial design information and production techniques together. In order to produce a product according to the traditional methods a human expert is needed continuously. However, rapidly developing progressions in the technological production techniques need not to be depended on a person who is experienced in solving these kinds of productions' problems. Because, becoming a human expert in any field takes too much time. In recent years in order to convert of a design into a product instead of a human who is as a planner different software has been developed in optimum conditions.

In this thesis, a generative Computer Aided Process Planning system (CAPP) has been developed to carry out the process planning for prismatic parts by using expert system approach automatically.

In this system, STEP standard has been used to represent 3D (three dimensional) parts for the Product Model Data Exchange (PMDE) depending on the Boundary Representation (B-rep).

The system consists of blank and finished part design, reference surface selection, operation selection and sequencing, cutter and machine tool selection,

calculation of machining parameters and fixturing and intermediate part generating modules. Output of the system for sample parts handling in this study include process planning, cutter and machine tool selection and fixturing charts.

**Key Words :** Computer Aided Process Planning, Expert System Approach, Feature Based Design, STEP Standard



## TEŐEKKÜR

Bu arařtırmanın Doktora Tezi olarak planlanıp, y¼r¼t¼lmesi ve sonularının deęerlendirilmesi sırasında daima yardımlarını ve teŐvięini g¼rd¼ę¼m ve birlikte alıŐtıęım danıŐman hocam sayın Prof. Dr. Ali ¼N¼VAR'a ve bazı d¼k¼manların temininde bana yardımcı olan Konya TSE B¼lge M¼d¼r¼ sayın Yrd. Do. Dr. Fatih BOTSALI'ya teŐekk¼r¼ bir bor bilirim. Ayrıca T¼bitak Marmara AraŐtırma Merkezi Yapay Zeka B¼l¼m¼ uzmanlarından sayın Do. Dr. Ercan ¼ZTEMEL ve alıŐma arkadaŐlarına ve araŐtırma safhasında bana yol g¼steren G.¼. Teknik Eęitim Fak¼ltesi ¼đretim ¼yesi sayın Do. Dr. Mahmut G¼LESİN'e teŐekk¼r ederim.



## İÇİNDEKİLER

	<b><u>Sayfa No</u></b>
ÖZET.....	i
ABSTRACT .....	iii
TEŞEKKÜR.....	v
ŞEKİL LİSTESİ .....	x
KISALTMA LİSTESİ.....	xii
1. GİRİŞ.....	1
1.1. Bilgisayar Destekli Tasarım ve Üretimin Otomasyondaki Rolü.....	1
1.2. İmalatta Bilgisayar Destekli Tasarım ve Üretim Sistemlerinin Bütünleştirilmesi.....	3
1.3. Araştırmanın Amaçları ve Kapsamı.....	6
2. TARİHÇE VE KAYNAK ARAŞTIRMASI.....	8
2.1. Manuel (Elle) İşlem Planlama .....	8
2.2. Bilgisayar Destekli İşlem Planlama Sistemleri .....	8
2.2.1. Değişken işlem planlama sistemi.....	10
2.2.2. Üretken işlem planlama sistemi .....	11
2.2.3. Bilgi tabanlı üretken işlem planlama sistemi .....	12
2.3. Parça Tasvir Etme Metotları .....	12
2.4. İşlem Planlama Üzerine Yapılan Çalışmalar .....	13
2.4.1. Geleneksel yaklaşımla hazırlanmış İşlem planlama sistemleri .....	13
2.4.2. İşlem planlamada uzman sistemler .....	16
3. YAPAY ZEKA VE UZMAN SİSTEMLER .....	22
3.1. Yapay Zeka .....	22
3.2. Uzman Sistemler .....	23
3.2.1. Bilginin anlamı.....	27
3.2.2. Bilgi temsil etme teknikleri.....	28
3.2.2.1. Predicate (ilişki) mantığı.....	29
3.2.2.2. Üretim kuralları.....	29
3.2.2.3. Semantik ağlar.....	33
3.2.2.4. Çerçevesel (Çatılar).....	37
3.3. Uzman Sistemlerde Sonuç Çıkarma.....	41
3.3.1. Ağaçlar, kafesler ve grafikler.....	42
3.3.2. VE-VEYA (AND-OR) ağaçları ve amaçlar .....	44
3.4. Sonuç Çıkarma Teknikleri .....	46

3.4.1. İleriye zincirleme .....	47
3.4.2. Geriye zincirleme .....	49
3.5. Temel Araştırma Teknikleri.....	50
3.5.1. Derinlik ilk araştırma tekniği .....	52
3.5.2. Genişlik ilk araştırma tekniği.....	52
4. GELİŞTİRİLEN SİSTEMİN MODELLENMESİ.....	54
4.1. STEP Standardı ve US Yaklaşımı Kullanan İşlem Planlama Sistemi .....	55
5. BİLGİSAYAR ORTAMINDA PARÇA MODELLEME.....	59
5.1. Üç Boyutlu Cisimleri Temsil Etme Teknikleri.....	59
5.1.1. Konstrüktif katı geometri (KKG).....	59
5.1.2. Sınır temsili (ST) .....	60
5.2. Özellik Tanıma ve Özellik Tabanlı Tasarım.....	61
5.2.1. Özellik tanıma .....	61
5.2.2. Özellik tabanlı tasarım .....	62
5.3. BDT Verileri Değişirme Standartları.....	62
5.4. STEP Standardın Genel Hatları.....	64
5.4.1. Uygulama katmanı.....	65
5.4.1.1. Şekil modelleri .....	65
5.4.1.2. Geometri.....	67
5.4.1.3. Topoloji.....	69
5.4.1.4. Şekil özellikleri .....	71
5.4.1.5. Boyutlar ve toleranslar .....	73
5.4.2. Mantıksal katman.....	76
5.4.2.1. EXPRESS dili .....	77
5.4.3. Fiziksel katman .....	78
5.5. STEP Dosyasının Prolog Cümleciklerine Dönüştürülmesi.....	78
5.5.1. Prolog dizini.....	78
6. STOK SEÇİMİ, TASLAK VE BİTMİŞ PARÇA MODELLEME.....	81
6.1. Komşu Yüzey Grafiği (KYG).....	81
6.1.1. KYG'de yüzey temsil etme .....	84
6.1.2 KYG'de delik temsil etme .....	87
6.2. Stok Seçme ve Taslak Parça Oluşturma Modülü .....	88
6.2.1. Stok boyutlarının belirlenmesi .....	89
6.2.2. İdeal boyutun seçimi .....	90
6.2.3. Taslak parça oluşturma .....	91
6.2.4. Dosyaya taslak modeli yazma .....	91
6.3. KYG'de Bitmiş Parçayı Modelleme .....	93

6.3.1. Yüzeyler arasındaki açılar hesabı .....	94
6.3.2. İçbükeylik-dışbükeylik ilişkisinin belirlenmesi.....	94
6.3.3. Özelliklerin işlenmesi .....	96
6.3.3.1. Kademe özelliği işleme .....	96
6.3.3.2. Kanal özelliği işleme.....	97
6.3.4. Son modeli dosyaya yazma.....	98
<b>7. UZMAN SİSTEM, REFERANS YÜZEYİ SEÇİMİ VE</b>	
<b>OPERASYON SIRALAMA.....</b>	<b>101</b>
7.1. Uzman Sistem.....	101
7.1.1. Bilgi tabanı .....	101
7.1.2. Kuralları yapılandırmak için VE/ VEYA grafiği .....	104
7.1.3. Bilgi tabanının prolog cümleciklerine çevrilmesi .....	107
7.1.4. Sonuç çıkarma mekanizması .....	107
7.1.5. Kullanıcı arayüzü .....	109
7.1.6. Bilgi edinme ve sürdürme .....	109
7.2. Referans Yüzeyi Seçimi.....	109
7.3. Operasyonların Sıralanması .....	110
<b>8. OPERASYON, TAKIM VE TAKIM TEZGAHI SEÇİMİ .....</b>	<b>115</b>
8.1. İmalat Operasyonu Seçimi .....	115
8.1.1. Operasyonları seçmede kullanılan bilgi tabanı .....	115
8.1.1.1. Frezeleme operasyonları .....	119
8.1.1.2. Delme operasyonları.....	120
8.1.1.3. Raybalama operasyonları .....	123
8.1.1.4. Delik işleme operasyonu .....	124
8.1.1.5. Taşlama operasyonları.....	124
8.1.1.6. Honlama operasyonları .....	124
8.1.1.7. Klavuz çekme, silindirik ve konik havşa açma, oturma yüzeyi işleme.....	125
8.1.2. İşlem seçiminde sonuç çıkarma .....	125
8.2. Takım Seçimi.....	126
8.2.1. Veri tabanında kesme takımlarını temsil etme .....	126
8.2.2. Kesme takımı seçme prosedürü .....	127
8.3. Takım Tezgahı Seçimi .....	132
8.3.1. Veri tabanında takım tezgahlarını temsil etme .....	134
8.3.2. Takım tezgahı seçme prosedürü .....	135
8.3.2.1. Tezgah tipini belirleme .....	135
8.3.2.2. İşleme parametrelerinin hesaplanması.....	136
8.3.2.3. Takım tezgahının belirlenmesi.....	141

9. BAĞLAMA İŞLEMİ VE ARAPARÇA OLUŞTURMA .....	143
9.1. Bağlama İşleminin Yapılması.....	143
9.1.1. 6 Nokta metodu.....	144
9.1.2. Bağlama prosedürü .....	146
9.1.2.1. Yüzey frezeleme için bağlama işlemi.....	146
9.1.2.2. Kademe frezeleme için bağlama işlemi .....	148
9.1.2.3. Delme için bağlama işlemi.....	150
9.1.2.4. Kanal frezeleme için bağlama işlemi .....	150
9.2. Araparça Modelleri Oluşturma.....	151
10. ARAŞTIRMANIN DEĞERLENDİRİLMESİ VE TARTIŞMA.....	155
11. SONUÇ VE ÖNERİLER .....	160
12. KAYNAKLAR.....	162

## EKLER





## ŞEKİL LİSTESİ

<b><u>Sekil No</u></b>	<b><u>Sayfa No</u></b>
1.1. Bütünleşik BDT/BDİP/BDÜ sistemleri .....	5
2.1. Geleneksel işlem planlama prosedürü .....	9
3.1. Bilginin hiyerarşisi .....	28
3.2. Araba arıza teşhisinde sorgulama başlangıç durumu .....	31
3.3. Kural 1 çalıştırıldıktan sonraki durum .....	31
3.4. Kural 4 çalıştırıldıktan sonraki durum .....	32
3.5. Araba arıza teşhisi AND/OR grafiği .....	32
3.6. İki tip ağ yapısı .....	34
3.7. IS-A ve A-Kind-Of ilişkilerini gösteren semantik ağ .....	35
3.8. NND tablosu .....	37
3.9. Kademe özelliğinin semantik ağla temsili .....	37
3.10. Otomobil çerçevesi .....	38
3.11. Özellik için genel çerçeve .....	39
3.12. Otomobil özelliğinin genel alt çerçevesi .....	40
3.13. Bir araba çerçevesi örneği .....	40
3.14. Takım tezgahı çerçevesi .....	41
3.15. Fil çerçevesi .....	41
3.16. İkili ağaç yapısı .....	43
3.17. Tezgah seçimi yapan karar ağacı .....	44
3.18. Bir aracın nasıl alınabileceğini gösteren VE-VEYA yapısı .....	45
3.19. Evden işe gitme alternatiflerini gösteren VE-VEYA yapısı .....	46
3.20. Uzman sistemlerde sonuç çıkarma .....	47
3.21. İleriye zincirlemede sonuç çıkarma .....	48
3.22. Yolculuk seçiminde problem uzayı .....	51
3.23. Geriye iz sürme ve derinlik ilk araştırma stratejisi .....	52
3.24. Genişlik ilk araştırma stratejisi .....	53
4.1. Geliştirilen sistemin yapısı .....	56
5.1. Bir parçanın KKG ile temsil edilmesi .....	59
5.2. Bir parçanın ST ile temsil edilmesi .....	60
5.3. STEP standartın yapısı .....	66
5.4. STEP'te geometrinin mimarisi .....	69
5.5. STEP'te topoloji sınıflandırma yapısı .....	70
5.6. STEP standartta şekil özelliklerinin sınıflandırılması .....	72
5.7. Tolerans ve boyut sınıflandırma yapısı .....	75
6.1. Yüzeyler arasındaki içbükeylik-dışbükeylik ilişkisi .....	82

6.2. KYG için örnek parça.....	83
6.3. KYG'de parça temsil etme.....	83
6.4. L Kademe özelliklerini oluşturmak için parçaya yerleştirme ve işleme.....	85
6.5. Parçaya özellikleri yerleştirme öncesi ve sonrası yüzeylerin durumu.....	86
6.6. Örnek parçanın Y1 yüzeyinin KYG'de temsil edilmesi.....	88
6.7. KYG'de Y1 yüzeyinin kenar çevrimi ve koordinatları.....	88
6.8. KYG'de Y5'in temsil edilmesi.....	89
6.9. Stok seçiminde Y ve Z boyutlarının artırılması.....	91
6.10. Örnek parça için oluşturulan taslak parça.....	92
6.11. Örnek parçayı kapsayan taslak parça.....	92
6.12. Yüzeylerin içbükeylik ve dışbükeylik ilişkilerinin belirlenmesi.....	95
6.13 L kademe özelliğinin yerleştirilmesi, uyumu ve işlenmesi.....	96
6.14. U şekil özelliğinin yerleştirilmesi ve işlenmesi.....	97
6.15. Son modeli dosyaya yazma.....	99
6.16. Kademe özellikleri yerleştikten sonra meydana gelen yüzeyler.....	100
7.1. Uzman sistemin grafik olarak gösterimi.....	102
7.2. Kuralların diğer kuralları referans alması.....	103
7.3. Amaç1 ve amaç2 için VE/ VEYA grafiği.....	105
7.4. Üretim kurallarının VE/VEYA grafiği.....	106
7.5. Referans yüzeyi seçme modülü.....	111
7.6. Referans yüzeyi seçme işlemi.....	112
7.7. Operasyonların sıralanması.....	113
8.1. İmalat operasyonları seçme modülü.....	116
8.2. Kesici takım seçme modülü.....	128
8.3. Kesme takımının geometrisi.....	131
8.4. Takım tezgahı seçme modülü.....	136
8.5. Taylor'un takım-ömür bağıntısı.....	138
8.6. Takım ömrü (tr) 60 sn için kesme hızı (vr) değerleri.....	139
9.1. Tespit edilmemiş bir bloğun uzaydaki serbestlik derecesi.....	145
9.2. Bir parçayı 6 nokta prensibine göre yerleştirilme ve bağlama.....	145
9.3. Birinci ve ikinci yerleştirme yüzeylerinin belirlenmesi.....	148
9.4. Yüzey frezeleme için bağlama yüzeylerinin belirlenmesi.....	149
9.5. Kademe frezeleme için bağlama işlemi.....	150
9.6. Sol kademenin alternatif bağlama yüzeyi olarak kullanılması.....	151
9.7. Delik ve kanal frezeleme için bağlama işlemi.....	152
9.8. Y5 yüzeyinin işlenmeden önce ve işledikten sonraki durumları.....	153
9.9. Sağ kademe özelliğini işlemeden önce ve işledikten sonra oluşan araparça model.....	154

<b>Kısaltma</b>	<b>Türkçe karşılığı</b>	<b>Kısaltma</b>	<b>İngilizce karşılığı</b>
BBÜ	Bilgisayarla Birleşik Üretim	CIM	Computer Integrated Manufacturing
BDİP	Bilgisayar Destekli İşlem Planlama	CAPP	Computer Aided Process Planning
BDT	Bilgisayar Destekli Tasarım	CAD	Computer Aided Design
BDÜ	Bilgisayar Destekli Üretim	CAM	Computer Aided Manufacturing
BSD	Bilgisayarlı Sayısal Denetim	CNC	Computerized Numerical Control
DSD	Doğrudan Sayısal Denetim	DNC	Direct Numerical Control
EÜS	Esnek Üretim Sistemleri	FMS	Flexible Manufacturing Systems
GT	Grup Teknolojisi	GT	Group Technology
KKG	Konstrüktif Katı Geometri	CSG	Constructive Solid Geometry
KKVY	Kanatlı Kenar Veri Yapısı	WEDS	Winged Edge Data Structure
KYG	Komşu Yüzey Grafiği		
LKS	Lokal Koordinat Sistemi	LCS	Local Coordinate System
MMK	Mimarlık, Mühendislik ve Konstrüksiyon	AEC	Architecture, Engineering and Construction
NND	Nesne-Nitelik-Değer	OAV	Object-Attribute-Value
SD	Sayısal Denetim	NC	Numerical Control
SEM	Sonlu Eleman Modelleme	FEM	Finite Element Modeling
ST	Sınır Temsili	B-rep	Boundary Representation
US	Uzman Sistem	ES	Expert System
YHÇ	Yüksek Hız Çeliği	HSS	High Speed Steel
YZ	Yapay Zeka	AI	Artificial Intelligence
2-B	2 Boyutlu	2-D	Two Dimensional
3-B	3 Boyutlu	3-D	Three Dimensional
	Başlangıç Grafikleri	IGES	Initial Graphics Exchange Specification
	Değiştirme Özelliği		
	Ürün Verileri Tanımlama Arayüzü	PDDI	Product Data Definition Interface
	Ürün Verileri Değiştirme Standardı/ Ürün Model Verilerini Değiştirme İçin STEP Standardı	PDES/ STEP	Product Data Exchange Standard/ STEP-Standard For The Exchange Of Product Model Data

## 1. GİRİŞ

### 1.1. Bilgisayar Destekli Tasarım ve Üretimin Otomasyondaki Rolü

Endüstrideki teknolojik gelişmeler ve üretici firmalar arasındaki rekabet tüm dünyada hızlı bir şekilde artmaktadır. Endüstriyel alandaki üretici firmalar, işletme verimliliğini ve bilgi yönetimini geliştirmek, sürdürmek, büyümek ve karlılık marjlarını korumak için fiyat, kalite ve teslim süreleri açısından sürekli rekabet etmek zorundadırlar. Endüstrideki bu gelişmeler firmaları tasarım, imalat ve yönetim alanında geleneksel yaklaşımlara karşı yeni alternatifler aramaya zorlamıştır.

Endüstrideki hızlı gelişmelerin en önemlilerinden birisi de bilgisayar teknolojisinin imalat organizasyonlarındaki uygulamasıdır. Zamanla donanım ve yazılımda meydana gelen gelişmeler, bilgisayarları karmaşık imalat problemlerini çözmede ve performansı arttırmada vazgeçilmez birer yardımcı araç haline getirmiştir. Bazı firmalar, bilgisayarları imalatın değişik kademelerinde uzun zamandan beri kullanmaktadırlar. İmalatta bilgisayarların kullanım alanını tasarım ve çizim, tezgah kontrolü, imalat kontrolüyle satış ve sipariş işlemlerinin kontrolü olarak sıralamak mümkündür. Tüm bu alanlarda daha önce elle yapılan faaliyetleri, bugün bilgisayarlar daha verimli ve etkin bir şekilde yerine getirmektedir. Bununla beraber halen imalat sistemleri tipik otomasyon hücrelerinden oluşmaktadır. Bu durumda, imalat işlemlerinin birbirlerinden bağımsız otomasyonu, imalat problemlerinin global optimizasyonuna çözüm getirmez [1].

Endüstride meydana gelen tüm değişimler sistematik olarak incelendiğinde imalat sahasındaki gelişmelerin, mekanizasyon kademesi ve otomasyon kademesi olmak üzere iki alanda gerçekleştiği görülmektedir. Mekanizasyon kademesinde imalat işlemleri döküm, dövme, tornalama, frezeleme, delme ve kesme gibi değişik işlemlere bölünerek bunlara ait takımlar ve işlemler mekanize edilmiştir. Aynı zamanda imalat sanayiinde çalışan işçiler de, bu alanların birisinde uzmanlaşmaya yönelmişlerdir. Uzmanlaşmada son aşama, imalattan tasarımın ayrılması olmuştur. Böylece tasarımcı, bir ürünün tasarımını yaparken imalat alanında uzmanlaşmış bir ekip de, bu çizimlerle ilişkilendirilen ürünü imal etmektedir. Mekanizasyon kademesi, kütle üretimini

gerçekleştirmede yeterli olmasına rağmen, esneklik ve bütünleşmeden yoksundur [2].

İmalatın gelişmesinde ikinci aşama, otomasyon kademesi olarak adlandırılır. 1975' te transfer hatlarının kullanılması ile kütle üretimi, 1976'da da Esnek Üretim Sistemleriyle (EÜS) parti üretimi otomatik hale getirilmiştir. 1979'dan sonra, 1985'li yıllarda Bilgisayar Destekli Tasarım (BDT) ile Bilgisayar Destekli Üretim (BDÜ)'in bütünleşmesinde yaygın olarak kullanılan BDT ile tasarım ve çizim işlemlerinin yapılmasına yaygın olarak başlanılmıştır. Otomasyon kademesinin amacı, minimum seviyede insan gücünün kullanıldığı tamamen otomatik imalat sistemlerinin geliştirilmesidir. Bu alandaki çabalar, böyle bir düşünceye bağlı olarak sürdürülmektedir. Bununla beraber, bütünleşme henüz tam olarak başarılamamıştır. Tamamen otomatik hale getirilmiş bir fabrika, kütle üretimi ve hızlı imalatla birlikte esneklik ve tam bütünleşmeyi de sağlayacaktır.

İmalatta otomasyon, bilgisayarların bu alanda etkin bir şekilde kullanılmasıyla önemli ölçüde gelişmiştir. Böylece imalat kontrolü, malzeme nakli, planlama ve diğer faaliyetlerde de çeşitli gelişmeler sağlanmıştır [3]. İmalat kontrolünde bilgisayarın kullanımı, bilgisayar destekli Sayısal Denetim (SD) kodu üretmek üzere sayısal kontrolü geliştirmiştir. Günümüzde bazı BDT/BDÜ sistemleriyle bir BDT veri tabanında tasarlanan parçadan, doğrudan SD kodu üretmek mümkün olmaktadır [4]. SD tezgahlarının yerini Bilgisayarlı Sayısal Denetimli (BSD) takım tezgahlarının almasıyla, hemen hemen her talaş kaldırma işlemi verimli bir şekilde, yüksek bir tamlık derecesinde ve tekrarlanabilirlik kabiliyetiyle otomatik hale getirilmiştir [5].

BDT, hesaplama, analiz, modelleme ve test işlemlerini içeren tasarım işlemine yardım sağlamak üzere bilgisayarların kullanımı olarak tanımlanabilir. Başlangıçta BDT sistemleri, öncelikle çizim amacıyla kullanılmıştır. Bununla beraber günümüzde BDT sistemleri, sonlu eleman modelleme, statik, dinamik, değişken sistemlerin ısıl gerilme analizi ve titreşim analizi içeren birçok mühendislik problemlerinin analizi için de kullanılmaktadır [6].

SD takım tezgahlarının gelişimi, BDÜ sistemlerinin başlangıcı olarak kabul edilir. BDÜ, herhangi bir imalat işlemine destek sağlamak veya onun performansını yükseltmek amacıyla bilgisayarların kullanılması olarak tanımlanabilir [7]. Günümüzde BDÜ; BSD, Doğrudan Sayısal Denetim (DSD), EÜS, Otomatik Malzeme Nakli, muayene ve bilgisayar kontrollü montaj sistemleri gibi fonksiyonların bir çoğunu kapsamaktadır [6].

BSD takım tezgahı, SD fonksiyonlarını icra etmek üzere, kendine özgü bir bilgisayar kullanan SD sistemidir [8]. DSD, bir imalat sistemi olarak



tanımlanabilir. Bu sistemde takım tezgahlarının bir kısmı eşzamanlı olarak merkezi bir bilgisayar tarafından kontrol edilir ve parça programı bir bilgisayardan herhangi bir takım tezgahına doğrudan iletilir [4].

EÜS, çeşitli mamülleri otomatik olarak üreten programlanabilir bir imalat sistemidir. EÜS, aynı zamanda BSD takım tezgahları, otomatik malzeme nakil sistemleri, robotlar ve bunları kontrol eden sistemlerin bileşimidir [7].

## **1.2. İmalatta Bilgisayar Destekli Tasarım ve Üretim Sistemlerinin Bütünleştirilmesi**

Bilgisayar teknolojisindeki gelişmelere paralel olarak BDT/BDÜ sistemlerinde sayısal kontrollü ekipmanlar ve robotların kullanımı, tecrübeye dayalı el becerilerinin bir çoğunu otomatik hale getirmiştir. Bunun sonucunda da hazırlık zamanında azalma, ürünlerde gelişme, imalatın tamlığı ve esnekliğinde artışlar kaydedilmiştir [5].

Bütünleştirilmiş BDT/BDÜ sistemlerinin herhangi bir şirkette önemli faydalar sağlayabileceği düşünülmüştür. Bu sebeple BDT/BDÜ sistemlerinin bütünleşmesi, fabrika otomasyonunu sağlamada önemli bir amaç olmuştur. Böylece Bilgisayarla Bütünleşik Üretim (BBÜ) sistemi, geliştirilerek BDT/BDÜ sistemlerinin tam olarak bütünleşmesini sağlamak amacıyla kullanılmıştır [9].

BDT ve BDÜ sistemlerinin tam olarak bütünleştirilmesindeki en büyük engel, komple bir BDT, Bilgisayar Destekli İşlem Planlama (BDİP) ve BDÜ arayüzü ve tam otomatik bir BDİP sisteminin olmayışıdır.

Tüm BDT sistemleri, kendi veri tabanlarında, kendilerine özgü matematiksel temsil tekniklerine sahip olmaları nedeniyle, bu sistemler birbirleriyle doğrudan iletişim kuramazlar [10]. BDT sistemleri arasındaki iletişimi sağlamak için, Başlangıç Grafikleri Değiştirme Özelliği (IGES), SET, VDA-FS gibi bazı tabii standart formatlar geliştirilmiştir [11]. Bu standartlar içerisinde en iyi bilinen IGES, BDT sistemleri arasında veri transferi yapmak üzere yaygın bir şekilde kullanılmaktadır [10]. Bununla beraber bu tip standartlar, BDT ve BDİP/BDÜ sistemleri arasındaki iletişim problemlerini tam olarak çözememiştir. Grafik standartları, BDT ve BDÜ sistemleri arasında veri transfer etmekten ziyade, orijinal olarak BDT sistemleri arasında geometrik verileri transfer etmek üzere geliştirilmiştir. Grafik standartların en büyük eksikliği, sadece şekil özelliği tanımlamalarıdır. Halbuki BDİP sistemleri- şekil özellikleri yanında- toleranslar, yüzey pürüzlülüğü, boyutlar ve malzeme bilgisi gibi diğer teknolojik bilgilere de ihtiyaç duyarlar. Bundan dolayı grafik standartlar, komple bir ürünü tanımlamada yetersizdir. Mevcut standartların ihtiyaçlara cevap vermedeki

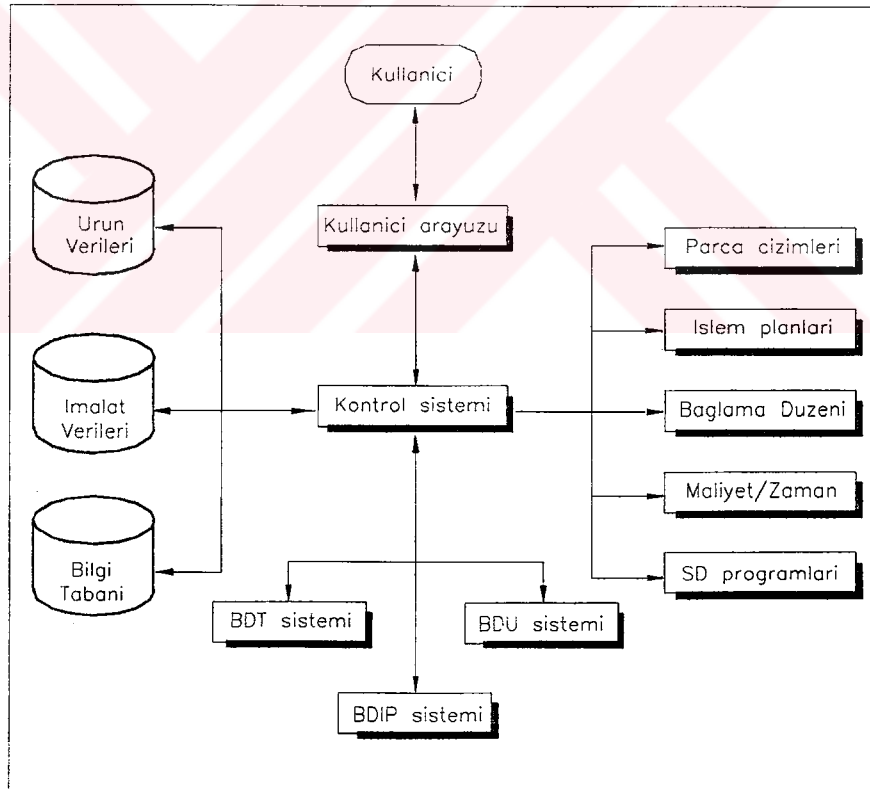
yetersizliğinin anlaşılmasıyla, IGES/PDES organizasyonu tarafından yeni bir standart olan Ürün Verileri Değiştirme Özelliği (PDES) tasarlanmıştır. Bu yeni standart halen **Ürün Model Verileri Değiştirme Standardı** (STEP) adıyla geliştirilmektedir. STEP'in 1.0 versiyonu 1988' de yayınlanmış olup, ISO tarafından uluslararası grafik verileri standardının ilk taslak teklifi olarak kabul edilmiştir [12]. STEP, komple bir ürünün tanımlanması için gerekli olan geometri, topoloji, şekil özelliği, tolerans, yüzey pürüzlülüğü ve malzeme gibi geometrik ve teknolojik bilgileri birlikte içermektedir. Bu yüzden tam olarak ürün verilerini temsil etme yeteneğine sahip, tabii bir grafik standardıdır. Sonuç olarak STEP standardı kullanılarak farklı BDT sistemleri ile BDT ve BDİP/BDÜ sistemleri arasında veri transferi sağlanabilecektir [13].

Mevcut BDT sistemlerinin çoğu, bir parçanın tüm geometrik bilgilerini içerir. Bu özelliklerinden dolayı BDT sistemleri, yüzeyler, noktalar, kavisler ve temel geometrik şekiller gibi düşük seviyeli bilgilerle mekanik parçaları tanımlar. Fakat BDİP/BDÜ sistemleri, cep, delik, kanal gibi şekil özellikleri yanında; tolerans, yüzey pürüzlülüğü, sertlik ve malzeme gibi teknolojik bilgilere de ihtiyaç duyar. Bu yüzden BDT veri tabanlarını, doğrudan BDÜ sistemlerinde ve işlem planlamada kullanmak uygun değildir. Bu bakımdan BDT, BDÜ ve BDİP sistemleri tarafından ortak kullanılacak bir veri tabanı gereklidir [14]. BDT ve BDÜ sistemlerini arayüz etmek için geliştirilen yaklaşımlardan birisi de **Özellik Tanıma**'dır. Özellik Tanıma, bir BDT modelini analiz eder ve şekil özelliklerini BDT modeli içinde belirler [15]. Temel geometrik şekillere sahip varlıklarla açıklanan bir BDT modelden, şekil özelliklerini çıkarmak üzere, birçok özellik tanıma sistemi geliştirilmiştir [16]. Çıkarılan şekil özellikleri, BDİP/BDÜ sistemlerine veri girişi olarak kullanılabilir. Bununla birlikte bu tür yaklaşımlar bazı kısıtlara sahiptir. Bunlar, şekil özelliklerinin sadece sınırlı bir miktarını tanıma ve bir şekil özelliğinin de sadece geometrik ve topolojik bilgiye sahip olmasıdır. Özellik tanıma yaklaşımı, teknolojik bilgi desteği sağlamaz; çünkü BDT modelleri, bu tür bilgileri içermez.

Özellik tanıma dışında diğer bir yaklaşım, **Özellik Tabanlı Tasarım**'dir. Bu yaklaşım özellik tanıma prosedürünü ortadan kaldırır. Son yıllarda BDT sistemlerindeki özellik tabanlı tasarım yaklaşımı, büyük bir gelişme göstererek BDT/BDÜ sistemlerinin bütünleşmesinde önemli bir destek sağlamıştır [17]. Parça özellik kavramını kullanmak suretiyle, bir parçanın karakteristikleri ile tasarım ve imalatı için ihtiyaç duyulan bilginin tümü, tasarım aşamasında belirlenebilir ve birbirleriyle ilişkisi olan parça özelliklerinin bir kümesiyle açıklanabilir. Ürün tanımlama için ortak erişilebilen ve kolayca yorumlanabilen bu tip verileri kullanarak otomatize edilmiş BDİP sistemi ile, tam olarak BBÜ

başarılabilir [18]. Parçalar, özellik tabanlı tasarım yaklaşımını kullanarak tanımlandıktan sonra, STEP gibi belirli bir formatta ortak veri tabanında depolanmalıdır. Böylece BDT, BDİP ve BDÜ sistemleri ürün verilerine doğrudan erişebilir.

BDİP sistemi, BDT ile BDÜ arasında köprü kuran önemli bir faaliyet olup, mekanik parçaları üretmek için, tasarım bilgilerini imalat talimatlarına dönüştürür (Şekil 1.1), [17]. Endüstride işlem planlama görevleri, işleme operasyonlarını dikkate alarak, bilgi ve tecrübesini manüel olarak kullanan tecrübeli bir işlem planlayıcı tarafından yerine getirilir. Burada geliştirilen planın kalitesi, planlayıcının tecrübesine ve tercihlerine bağlıdır. Ayrıca işlem planlamada planlayıcının tecrübesi, önemli bir birikimi gerektirdiğinden, planlayıcının işyerinden ayrılması durumunda onun yerini doldurmak bir hayli güç olacaktır [19].



Şekil 1.1. Bütünleşik BDT/BDİP/BDÜ sistemleri



Bugüne kadar birçok BDİP sistemi geliştirilmiştir. Bu sistemlerde bilgisayar, işlem planları üretimde planlayıcıya sadece yardım etmek üzere kullanılmıştır. Tüm bu çalışmaların yanında tam olarak otomatize edilmiş bir BDİP sistemi, henüz mevcut değildir. Gerçek endüstri çevrelerinde tam olarak otomatize edilmiş bir BDİP sisteminin önemli faydalar sağlayacağı düşünülmektedir [20].

BDİP teknikleri iyi bir şekilde geliştirilmediği takdirde, BBÜ başarısızdır [21]. İşlem planlamadaki karmaşık karar verme, işlem planlamayı otomatik hale getirmede önemli bir engeldir. Burada uzman bilgi, tecrübe ve akıllı sorgulamayı gerektiren görevlerin birçoğu, planlayıcı tarafından yerine getirilir [22]. Otomatik sistemlerin yerine getirilmesinde diğer önemli güçlükler, planlama mantığının elde edilmesi ve sezgisel bilgi ile ilgilidir [23].

Algoritmik programlama teknikleri, işlem planlamanın otomatizasyonu için elverişsiz olarak görülmektedir. Çünkü işlem planlama problemleri, sezgisel olarak insan kararı, tecrübe ve ampirik kurallara dayalı olarak çözülür [24]. Fabrika ortamında zamanla imalat işlemlerinde bazı değişikliklerin ve gelişmelerin olması doğal bir sonuçtur. Algoritmik programlama tekniğine göre hazırlanan BDİP sistemleri, yapıları gereği yeterli esnekliğe sahip olmadığından, imalat işlemlerinde meydana gelen değişikliklerin BDİP sisteme eklenmesi, yazılımın tüm yapısını etkiler [7]. Tüm bu olumsuzlukları aşmada **Yapay Zeka** (YZ)'nin işlem planlama faaliyetlerini otomatize etmede önemli bir yaklaşım olduğu düşünülmektedir. YZ' nin bir dalı olan **Uzman Sistem** (US) veya **Akıllı Bilgi Tabanlı Sistem**, bilgisayar ortamında mantık, sezgisel bilgi ve tecrübeyi elde etmek için, sorgulama faaliyetlerini otomatize etme yeteneğine sahiptir. Bir uzman sistem, belirli bir alan bilgisini özel bir tarzda temsil ederek depolar. Onun için uzman sistemlerde, programlarda herhangi bir değişiklik yapmaksızın, bilgi tabanındaki bilgiyi değiştirmek, silmek ve ilave etmek mümkündür.

### 1.3. Araştırmanın Amaçları ve Kapsamı

Bu araştırmanın amacı, US yaklaşımı kullanarak, prizmatik parçalar için bağlama işlemi ve işlem planlamada gereken sorgulama faaliyetlerini otomatize etmektir. Burada US yaklaşımı kullanılarak gerekli sonuçları çıkartan tam otomatik bir işlem planlama sistemi geliştirilecektir.

Bilginin en iyi şekilde temsil edilmesi, işlem planlamada önemli bir rol oynar. Bu bakımdan, Türkçe yazılımda bilgi temsil etme yeteneğine sahip bir tekniğin geliştirilmesi hedef alınmıştır.

Her imalat operasyonu, veri tabanındaki parça modelini etkiler. İşlem planlama ve bağlama işlemi hakkında güvenilir kararlar vermek için, her işlemden sonra ara parça modeller üretilmelidir. Bu yüzden taslak, bitmiş ve ara parça modellerini temsil etmek için özel bir temsil tekniği geliştirilecektir.

Endüstride üretilen parçaların çok çeşitli olması nedeniyle, bu çalışmanın kapsamı verilen herhangi bir problemi çözmekten çok, üzerinde çalışılan parçaların işlem planlarını üretmekle sınırlı tutulmuştur. Esas itibariyle sistem, düzlem yüzeyler, kanallar, kademeler ve delikler gibi basit şekil özelliklerinin bazılarını bulunduran prizmatik parçaları ele almaktadır. Kuralların bazısı, BSD işleme için uygulanabilir olmasına rağmen, bunların çoğunluğu, konvansiyonel takım tezgahları için yapılandırılmıştır. Veri tabanındaki mevcut kesme takımları, yüzey freze, alın freze, parmak freze çakıları ve helisel matkaplarla sınırlı tutulmuştur. Bununla beraber ihtiyaç duyulduğunda V-kanal, T-kanal freze çakıları gibi, değişik kesme takımları, veri tabanına kolayca ilave edilebilir. İşleme sırasında bir zamanda her özellik için sadece bir operasyon uygulanabilir. Yani grup frezeleme gibi operasyonların kombinasyonu dikkate alınmamaktadır. Programda yer alan kurallar tüm işleme operasyonlarını içermez. Bağlama modülü, parçayı yerleştirmek ve tespit etmek için destekleme, yerleştirme ve bağlama yüzeylerini tayin eder. Üretilen işlem planlama kartları, endüstri tarafından ihtiyaç duyulan tüm detayları kapsamaz. Planlama kartları, her operasyon için seçilen takım tezgahlarını ve takımları, operasyonların sırasını ve hesaplanan takım ömrü, hız ve ilerleme gibi işleme parametrelerini göstermek için kullanılır. Sistem, her parça için bir taslak model tasarlayarak, her bir yüzeyi veya şekil özelliğini buna bağlı olarak işler. Çalışmada sadece örnek olarak alınan parçaları oluşturmada gerekli olan stok boyutları dikkate alınmıştır. Parça üretiminde kullanılan döküm, dövme ve diğer hammadde şekilleri dikkate alınmamıştır. Sistem tarafından kullanılan STEP standardı, hala geliştirilmektedir. Bu nedenle STEP standardının herhangi bir bölümü değişirse, işlem planlama programları yeni tanımlamaları kapsayacak şekilde düzenlenmelidir.

## 2.TARİHÇE VE KAYNAK ARAŞTIRMASI

İşlem planlama, tasarım özelliklerine uygun olarak bir parçanın nasıl imal edilmesi gerektiğini belirleyen görevler olarak tanımlanabilir. Başka bir ifadeyle işlem planlama, bir tasarımın bir ürüne nasıl dönüştürüleceğini ifade eden detaylı talimatların sağlanması ve karar verilmesinden ibarettir. İşlem planlama, parçayı hammadde halinden, bitmiş parça şekline dönüştürmek için gerekli olan imalat talimatlarını oluşturur [25]. İlk olarak, parçanın geometrik ve teknolojik bilgileri değerlendirilir. Sonra işleme operasyonları, operasyon sırası, takım tezgahları, kesme takımlarının seçimi, işleme parametrelerinin hesaplanması ve bağlama ve referans yüzeylerinin belirlenmesi gibi operasyon detayları tayin edilir. Programın çıktısı, seçilen operasyonların sırasını ve işleme parametrelerini, ihtiyaç duyulan miktarda kesici takımı ve işleme kolaylıklarını içeren işlem planlama kartından meydana gelir. [8].

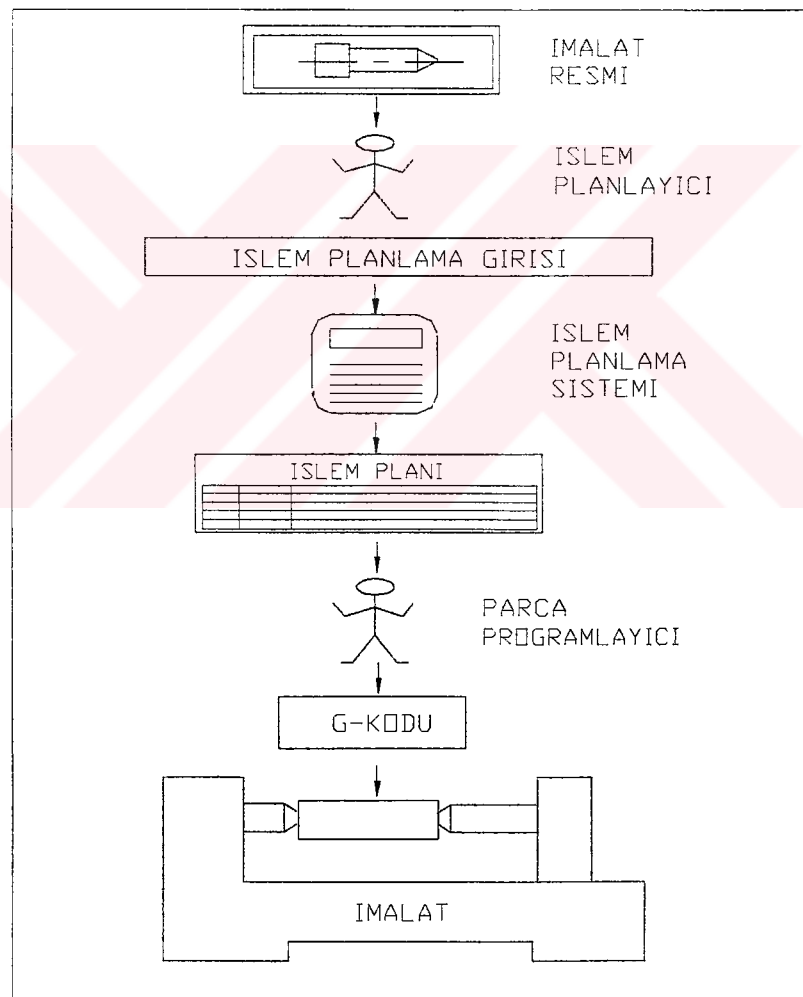
### 2.1. Manuel (Elle) İşlem Planlama

İşlem planlama, genellikle mühendislik çizimlerini (imalat resmi) inceleyerek imalat teknikleri, işlem kapasiteleri, takım tezgahları, takımlama ve malzemeleri de gözönüne alan, imalat bilgisi ve tecrübesine dayanarak bir parçanın nasıl imal edileceğine karar veren, tecrübeli bir işlem planlayıcı tarafından manuel olarak yapılır [8]. Geliştirilen planın kalitesi, planlayıcının tecrübesine, bilgisine, becerisine ve tercihlerine bağlıdır. Manuel işlem planlama, yoğun bir işçilik gerektirdiğinden hata yapma ihtimali yüksektir. Ayrıca üretilen işlem planları birbirlerinden farklı olup uzun zaman alır. Bundan dolayı bilgisayar kullanımı ile işlem planlama faaliyetleri hızlandırılabilir ve daha uygun planlar üretilebilir.

### 2.2. Bilgisayar Destekli İşlem Planlama Sistemleri

Manuel yaklaşımın dezavantajlarını ortadan kaldırarak BDT ve BDÜ arasındaki boşluğu doldurmak için, işlem planlama faaliyetlerini otomatize etmede, BDİP sistemleri geliştirilmiştir. Bilgisayar destekli işlem planlama

sisteminin mantığı iki temele dayanır. Bunlardan ilki, parça özelliklerinin elde edilmesi, ikincisi ise, verilen özellikleri karşılayacak parça üretimi için işlem planlarının oluşturulmasıdır. Bu mantığa göre hazırlanan BDİP sistemlerinin son amacı, insan müdahalesi olmaksızın değişik parçalar için işlem planları geliştirmektir. Hazırlanan BDİP sistemler, manuel planlamaya göre planlama zamanını azaltma, işlem planlama ve imalat maliyetlerini düşürme, uyumlu ve doğru planlar üretme ve prodüktiviteyi artırma gibi önemli avantajlar sağlar [26]. Geleneksel olarak BDİP'in iki esas yaklaşımı üretken (**generatif**) ve değişken (**varyant**) yaklaşım olarak bilinir. Üretken ve değişken olarak adlandırılan geleneksel işlem planlama sisteminin temel yapısı Şekil 2.1'de gösterilmiştir.



Şekil 2.1. Geleneksel işlem planlama prosedürü

İşlem planlamayı otomatize etmek için gerekli olan mantık, muhakeme ve tecrübe sağlanarak bilgisayar programı ile birleştirilmelidir.

### **2.2.1. Değişken işlem planlama sistemi**

Değişken işlem planlama, esas itibariyle manuel işlem planlamanın bir bilgisayarla desteklenerek genişletilmiş şeklidir. Burada bilgisayar planları yeniden elde etme, düzeltme, onları mevcut ihtiyaçlara uygun hale getirme ve benzer parçalar için planları seçmede yardımcı bir araç olarak kullanılır [8]. Planlayıcının bilgisi ve tecrübesi, günümüzde planların geliştirilmesinde hala anahtar faktörlerdir [27].

Değişken yaklaşımda Grup Teknolojisi (GT) kavramı esastır. GT, tasarım ve imalatta verimliliği ve etkinliği arttırmak için parça benzerlikleri hakkında bilgi kullanımı ve bir organizasyon metodudur. GT'de bilgiler, öncelikle organize edilir ve bir bilgisayarda depolanır. Depolanan bu bilgiler, daha sonra gerekli kararları vermek üzere etkin bir şekilde kullanılabilir [28]. GT'de parça sınıflandırması, benzer geometrik ve imalat karakteristiklerine göre parçaları gruplandırma ve belirleme için sınıflandırma şemaları olarak tanımlanabilir. Sonra her grup için kendi sınıfına esas olmak üzere bir kod numarası verilir [29]. Günümüzde DCLASS, OPITZ ve MICLAS gibi bazı sınıflandırma sistemleri mevcuttur [30]. GT, genellikle değişken sistemler tarafından kullanılır. Bununla birlikte GT, PFDM ve SAPT gibi bazı üretken sistemler tarafından da kullanılmaktadır. Benzer parçalar ailelerde gruplandırılır ve aileleri ayırmak için bir kodlama sistemi kullanılır [31].

Parça aileleri için mümkün olabilecek tüm operasyonları kapsayan standart işlem planları, her aile için oluşturulur ve aile kod numarası ile ilişkilendirilmiş olan sistemin veri tabanında depolanır. Bu kodlar, standart plan ile ilişkilendirilmiş olan parça ailesini tanımak için kullanılır. Yeni bir parça için işlem planı, yeni parçanın farklı özellikleri dikkate alınarak, benzer parça için standart planın yeniden elde edilmesi ile oluşturulur [32].

Değişken işlem planlamada bilgisayarlar, veri yönetimi, yeniden elde etme, düzeltme ve işlem planlarının yüksek hızda çıktısını almak için etkinlik sağlamada yardımcı olur. Mevcut işlem planlama sistemlerinin çoğu değişken tipindedir [8].

Değişken yaklaşım manuel işlem planlama ile karşılaştırıldığında, bilgi yönetim kapasitesinin artması, bir avantaj olarak görülür. Bu yaklaşımda işlem

planları daha az zaman ve işçilik gerektirir. Bunun yanında geometri, ölçü ve malzemenin değişik kombinasyonlarını yeterli derecede bir araya getirmede ve sistemin veri tabanını oluşturmada zorluklar vardır. Belirli parçalar için standart planları konstrükte etmek, sürdürmek, bir miktar değişiklik yapmak ve sürekli düzeltmek için kodlama planına aşına olması gereken tecrübeli bir işlem planlayıcıya ihtiyaç duyulması, değişken sistemlerin en büyük dezavantajıdır [33]. İmalatta planlayıcı tarafından yerine getirilen işlem planlarının birçoğu üzerinde yapılan çalışmalar, bir parçanın şekli ile planı arasında küçük bir ilişkinin olduğunu ortaya koymuştur. Örneğin iki parça benzer şekil ve özelliklere sahip olduğu halde, bunlardan birisi, ısıl işlem gerektirirken diğeri, herhangi bir kaplama işlemi gerektirebilir. Bu durumda bir sınıflandırma sistemi, işlem planlayıcıya parça şeklinden başka, bazı özellikleri de dikkate alması gerektiği konusunda yardımcı olmalıdır [28]. Değişken esaslı olarak üretilen işlem planı üzerindeki değişiklikler, manuel giriş gerektirdiğinden, değişken işlem planlama yaklaşımı, tamamen otomatize edilmiş bir sistemde kullanılamaz.

### **2.2.2. Üretken işlem planlama sistemi**

Üretken işlem planlama yaklaşımı, otomatik kompüterize edilmiş bir sistem kullanır. Bu sistem, işlem planlarını üretmek için işlem planlayıcı tarafından kullanılan karar mantığı, bilgi ve formülleri içerir [34]. Bu yaklaşım, değişken yaklaşımın aksine, standart işlem planları gerektirmez.

Bir üretken sistem parça bilgisi elde edildikten sonra takım tezgahları, kesici takımlar ve mevcut donanımları dikkate almak ve bilgiyi kullanmak suretiyle, işlem planını otomatik olarak yapabilir [20]. CAPSY, CIMS/PRO, APPAS, CMPP, CPPP, XPS, EXCAP, XPLAN, AUTOPLAN, AUTAP, GENPLAN, CADAM, AUTAP ve TIPPS gibi bir çok üretken işlem planlama sistemi geliştirilmiştir [30].

Üretken sistem tarafından meydana getirilen işlem planı, tutarlı, hızlı ve otomatiktir. Bununla birlikte komple ürün tanımlama, tanıma karmaşıklığı ile ilgili problemler, karar mantığı elde etme ve bir insan olarak işlem planlayıcının bilgisi ve tecrübesinden dolayı tam üretken işlem planlama sistemi mevcut değildir [35]. Bu şekildeki üretken işlem planlama sistemlerinin çoğu, yapısı gereği yarı üretkendir. Bunun için bu tür sistemler, tam üretken olarak sınıflandırılmaz. CPPP, APPAS , XPS, AUTOPLAN, GENPLAN ve AUTAP bunların bazı örnekleridir [36].



### 2.2.3. Bilgi tabanlı üretken işlem planlama sistemi

YZ tekniklerini kullanan bir US veya Bilgi Tabanlı Sistem, belirli bir sahada bir uzmanın muhakeme etme kapasitesini taklit etmeye çalışır [37]. Üretken BDİP sistemlerin uygulanmasındaki önemli zorluklar, bir bilgisayar ortamında planlama mantığı ve imalat bilgisinin elde edilmesiyle ilgilidir. BDİP'te algoritmik metotları uygulamada güçlükler olduğundan işlem planlama, öncelikle önemli derecede insan tecrübesine ve sezgisel bilgiye ihtiyaç duyar. BDİP sistemlerin geliştirilmesinde karar mantığı ve tecrübenin elde edilerek en yüksek seviyede değerlendirilmesi amacıyla US yaklaşımı kullanılır. US yaklaşımı, kendi kendine öğrenme kapasitesinin yerleşmesini, karar verme işlemlerini ve bilginin en iyi bir şekilde temsilini sağlar [32].

İşlem planlama, karmaşık sorgulama mekanizmalarını kapsar. Algoritmik programlama teknikleri, işlem planlama için geliştirilen US'ler kadar başarılı taklit yapamazlar. İşlem planlama problemlerini çözmek ve bir insan olarak tecrübeli bir planlayıcının düşünme işleminin karmaşıklığını modellemek için mükemmel algoritmik metotlar yoktur.

İşlem planlama esnek olmalı, çünkü sistemde bulunan imalat kurallarını ve olguları, imalat teknolojisindeki hızlı değişmeden dolayı sürekli güncelleştirmek gerekir. Geleneksel BDİP sistemde, kurallar ve olgularda herhangi bir değişiklik yapıldığında orijinal programın yeniden yazılması gerekir. Algoritmik programlama tekniklerinin rijitliği, ideal BDİP sistemleri oluşturmada bir engeldir. US, özel bir tarzda bilgi depolar. Onun için US'de yazılımın yapısını etkilemeksizin bilgi tabanındaki olguları ve kuralları değiştirmek, silmek ve ilave etmek mümkündür [7].

Bu yüzden YZ'nin, işlem planlama faaliyetlerini otomatize etmeyi daha yüksek seviyede sağlayacağı dikkate alınır [38]. GARI, TOM, SIPP, SAPT, PROPLAN, EXCAP, FREXPP, XPS-1, CUTTECH ve AGFPO gibi çeşitli YZ tabanlı işlem planlama sistemleri geliştirilmiştir [39].

### 2.3. Parça Tasvir Etme Metotları

İşlem planlama sistemi tarafından ihtiyaç duyulan bilgi girişinin önemli bir bölümünü parça tasviri oluşturur. Manuel işlem planlama sistemleri, işlem planlayıcı tarafından yorumlanan mühendislik çizimlerini kullanır. Elde edilen bilgiler, daha sonra işlem planını geliştirmek için kullanılır. BDİP sistemleri için bir parçanın geometrik tasviri; GT esaslı parça kodları, özel parça tasvir etme dilleri (etkileşimli giriş) ve bir BDT modelden sağlanır [40].

Mevcut BDİP sistemlerin çoğunda parça tasviri, GT şemalarına dayanır [41]. Bu sistemlerde parçalar sınıflandırılarak kodlanır. Böylece kodlar, sisteme giriş olarak kullanılır. Özellikler, ölçüler ve toleranslar gibi parça tasarım nitelikleri OPITZ, COFORM, DCLASS ve MICLAS [42] gibi şemaları kullanarak bir kodda saklanır. Genellikle GT kodları, parça tasviri için değişken BDİP sistemleri tarafından kullanılır [43].

Üretken BDİP sistemlerde parça tasviri, metinsel girişle (burada kullanıcı, etkileşimli giriş olarak soruların bir kaçını cevaplar) veya bir BDT sistemden sağlanan grafik girişle yerine getirilir. Grafik giriş, kendi karmaşası yüzünden hala geliştirilememiştir. Bu yüzden mevcut BDİP sistemlerinde genelde etkileşimli giriş uygulanır. Etkileşimli giriş kullanan bazı üretken BDİP sistemleri, GCAPPS , AUTAP, EXCAP ve SIPPS' dir [44].

BDİP ve BDT sistemleri arasındaki arayüz, BDÜ ve BDT'in bütünleşmesi için zorunludur. BDT'yi BDİP'e birleştirmek için, şekil özellikleri gibi düşük seviyeli geometrik bilgiyi yüksek seviyeli bilgiye dönüştürmek üzere BDİP sistemleri tarafından kullanılabilen çeşitli özellik tanıma sistemleri geliştirilmiştir. BDT modeller, parça tasvirinin bilgisayarda okunabilir bir şeklini sağlar. Bununla beraber bu modeller, düşük seviyeli varlıklarla tanımlanan bir BDT modelden şekil özelliklerini seçmek için özellik tanıma sistemlerine ihtiyaç duyarlar.

CAD/CAM ve TIPPS gibi ilk geliştirilen sistemler, işlem planını otomatik olarak yapmak için kullanıcı etkileşimli özellikleri tanımada, BDT modelleri kullanmıştır. Etkileşimli olan bu yaklaşımlar, tamamen otomatik değildir. Belirli girinti ve çıkıntıları tanımada syntactic model tanıma tekniği kullanarak sınıflandırma parçaları için bir GT kodu oluşturmak üzere Kyprianou tarafından Üç Boyutlu (3-B) parça modellerinden yararlanılmıştır [45]. Choi, deliklerin şeklini tanımak için 3-B veri tabanından alınan deliğin İki Boyutlu (2-B) enine kesitini kullanmıştır [46]. Liu ve Srinivasan, şekil özelliklerini tasvir etmek için syntactic model tanımayı kullanmışlardır [47]. Kung, arakesiti olmayan bir takım özellikleri tanımak ve işlem planları üretmek için US kurallarını kullanmıştır. Henderson ise, özellikleri tanımak için mantık tabanlı bir yaklaşım kullanmıştır [48].

## **2.4. İşlem Planlama Üzerine Yapılan Çalışmalar**

### **2.4.1. Geleneksel yaklaşımla hazırlanmış işlem planlama sistemleri**

MIPLAN, işlem planlama ile ilgili zaman standartlarını üreten değişken bir planlama sistemidir. Sistem, parça tasviri için MICLASS kodlama şemasını



kullanır. MIPLAN parça kodu, aile matrisi ve kod aralığını esas alarak işlem planlarını elde eder [49].

CAPP sistemi, parçaları sınıflandırmak ve kodlamak için mantık tabanlı GT metotları kullanan değişken bir işlem planlama sistemidir [50]. Sistem, veri tabanı için yapı, yeniden elde etme mantığı ve etkileşimli düzeltme kapasitesi sağlar. Çıkış formatı ve parça sınıflandırma için kodlama şeması, kullanıcı tarafından ilave edilir. Sistem, dönel, prizmatik ve saç metal parçalara uygulanır.

APPAS, Fortran dilinde yazılmış üretken bir sistemdir [51]. İşlenen her yüzeyin detaylı teknolojik bilgi tasviri, özel bir kod yardımıyla yapılır. Bir kapalı karar ağacı yaklaşımı, belirli özelliklere sahip değişik çapta delikler ve tasarlanan yüzey için çok sayıda pasoları seçme ve işleme kapasitesiyle işlem kapasitelerini kodlamak için kullanılmıştır. Bu sistem, delme ve frezeleme operasyonlarına uygulanmıştır.

AUTAP, dönel ve saç metal parçalar için geliştirilmiş üretken bir sistemdir. AUTAP, imalat talimatları oluşturmada kullanılan bütünleştirilmiş sistemin bir parçasıdır. Sistem, taslak malzeme ve operasyon sırasını tayin eder. Takım tezgahları seçildikten sonra işleme parametreleri hesaplanır [52].

ICAPP, prizmatik parçalar için Fortran dilinde yazılmış değişken bir işlem planlama sistemidir [53]. ICAPP özellik uyumlu olup, yüzey frezeleme, çevresel frezeleme, delme, delik büyütme, raybalama, klavuz çekme, silindirik ve konik havşa açma gibi sekiz temel işleme kapasitesine sahiptir. ICAPP, son zamanlarda IGES dosyası kullanılarak bir BDT sistemiyle arayüz edilmiştir.

XPS-1, Fortran 77'de yazılan geliştirilmiş üretken işlem planlama sisteminin prototipidir. Burada kullanıcı, kendi şartları için imalat mantığını ve veri ihtiyaçlarını belirler. Sistem, menü seçimi için kullanıcı etkileşimi, veri kayıt ekranı ve veri dosyasını korumayı sağlar. XPS-1, gelişmiş üretken işlem planlama sistemi için bir temel çerçeve çalışması sağlamak üzere tasarlanmıştır [54].

MICROPLAN sistemi, silindirler, koniler ve şeritler gibi temel geometrik özelliklere dayanan parça modelleme kullanır. Bu sistemde delikler negatif özellikler olarak ele alınır. Yüzey bitirme ve toleransla birlikte düzlükler ve vidalar gibi ilave özellikler uygun temel geometrik varlıklarla ilişkilendirilebilir. Her temel geometrik varlığın yerleştirilmesinde ona komşu temel geometrik varlıkla olan ilişkisinin belirlenmesinde işletmeciler kullanılmıştır [55].

KAPPS, CIMS/MODE olarak adlandırılan BDT sistemden sağlanan parça bilgisini kullanan bilgi tabanlı bir işlem planlama sistemidir. Sistem, esas itibarıyla BDT arayüzü ve kullanıcı giriş modülü, karar verme modülü, bilgi tabanı ve bilgi edinme alt modülünden oluşur. Bilgiyi temsil etmede çerçeveler

ve üretim kuralları kullanılır. Sistem, parçanın işlenen yüzeyini ve kaba şeklini tanıyabilir, referans yüzeyini, takım tezgahlarını, bağlama yüzeylerini ve kesme takımlarını seçebilir ve öncelikli ilişkileri ve kesme şartlarını belirleyebilir [21].

CAPES, program menüleriyle istenen operasyonlara erişmek için yapılandırmayı gerçekleştirmede kullanıcıya izin veren, değişken yaklaşım esaslı bir veri tabanıdır [56]. CAPES, bir işlem planlama ve tahmin etme sistemidir. Sistem, işlem planları, zaman standartları, tahminler ve işlemlerin bütün tipleri için rota kartları üretmek üzere genel bir program kümesine sahiptir.

Zhao ve Kops, çizelgeleme sistemi ICAPPS ve bütünleştirilmiş BDİP'i şekillendirmek için, grup çizelgeleme algoritması ile ilgili GT esaslı otomatik BDİP sistemini geliştirmişlerdir. İş düzeni ve takım tezgahı paylaşmada değişme olduğunda, muhtemel uyuşmazlıkları ortadan kaldırmak için, sistem otomatik olarak işlem planını düzenler [57].

Li, Han ve Ham, CORE adlı kural tabanlı yarı üretken bir BDİP sistemi geliştirmişlerdir. Sistem, parça ailelerinde farklı parçalar için işlem planları oluşturmada GT kavramını kullanır. Sistem, GT kodlama, parça ailesi araştırma, parça özellik yorumlayıcısı, parça bilgi girişi, plan üretme mantığı ve rapor üreticisinden meydana gelmiştir [58].

Tulkoff, deneyimli üretken sistem olan GENPLAN'ı geliştirmiştir. Sistem, kapsamlı operasyonlar üretmek için işlem değişkenleri ve parça geometrisini içeren GT esaslı kodlama şemasını kullanır. Sistem, yönetim karar destek sistemi, otomatik atelye yükleme ve çizelgeleme sistemi, takım düzeni hazırlama gibi diğer sistemlerle bütünleştirilmiştir [59].

TURBO-CAPP Prolog'da yazılmış bilgi tabanlı BDİP sistemidir. TURBO-CAPP, BDT veri yorumlayıcısı, işlem planlama sistemi, SD kodu üreticisi, bilgi edinme mekanizması ve bilgi esaslı yönetici olmak üzere beş modülden meydana gelir. BDT verileri yorumlayıcısının en önemli fonksiyonu, bir tel kafes (2 ve 1/2 boyutlu) BDT sistemden geometrik varlıkları çıkarmak ve tüm yüzey özelliklerini tanımlamaktır. Sistemin algoritması, miller, şaft ve pimler gibi simetrik dönel parçaları yorumlama yeteneğine sahiptir. Çıkarılan her yüzey özelliği daha sonra yüzey koduna atanır ve veri tabanında depolanır [60].

Isik ve Mileham, konvensiyonel takım tezgahlarında imal edilen dönel parçalar için üretken bir işlem planlama sistemi geliştirmişlerdir [40]. BEPPS olarak adlandırılan modüler sistem, prizmatik parçaların işlenmesini ve bağlama işlemini içerecek şekilde BSD operasyonlarına özellik eklemeye müsaade eder. Ayrıca BDT ve üretim yönetimi sistemlerini birleştirerek maliyet tahminlerini hesaplar. Sistem, parça verisi sağlamak için ilk olarak etkileşimli bir sistemi kullanır ve işlem planını otomatik olarak oluşturur. Planlama kademesi, işleme

sırası, uygun makinalar ve seçilecek takımlama, belirlenen operasyonlar ve hazırlanan zaman tahminlerini düzenleme özelliklerine müsaade eden uzman prosedürel kurallara dayanır. Kurallar, üretim kuralları formunda Fortran 77'de yazılmıştır.

AMOPPS, üretken bir işlem planlama sistemi olup BDT, BDİP ve BDÜ sistemlerinin fonksiyonlarını bütünleştirir. AMOPPS, IGES dosyasından alınan parça verilerini kullanarak, uygun SD parça programı talimatlarını (G-kodu) çıktı olarak üretir. Sistem, silindirik, alın ve kanal tornalama gibi üç termel operasyonu ele alır. AMOPPS, bir parçanın işlem planını hızlı bir şekilde yerine getirmek üzere tasarım ve imalat bilgisini birbirine bağlar, belirlenen ihtiyaçlar ile uyum sağlar. Sistem, işleme parametrelerini hesaplayarak bir parçanın işlenmesi için gerekli olan kararları otomatik olarak verir. Burada kullanıcı, sadece iş parçası malzemesini seçer ve taslak parça boyutlarını belirler. Sistem, Turbo Pascal'da yazılmış olup Macintosh mikro bilgisayarlarda uygulanır. AMOPPS, IGES yorumlayıcısı, çizim, BDİP, hesaplama, G-kodu üreticisi, kesme sırasının simülasyonu ve diğer destek modüllerinden meydana gelir. AMOPPS, AUTOCAD gibi bir BDT sistemi tarafından üretilen bir IGES dosyasından alınan geometrik ve boyut bilgilerini kullanır. Bu doğrudan bilgi transferi işlem planlama sistemi için bir çizimi yorumlama ve girişleri hazırlamak üzere insana olan ihtiyacı elimine eder. Böylece sistem, işleme operasyonlarının sıralamasına karar vermek için bir insan işlem planlayıcısının mantığını takip eder. Sistemden elde edilen sonuçlar bir BSD takım tezgahına okunabilir parça programı şeklinde dönüştürülür ve bu parça programı parça üretmede tezgahı kontrol etmek için kullanılabilir [61].

Ngoi, basit geometrik ve yüzey özelliklerine sahip simetrik-dönel parçalar için özel bir profil tanımlama metodu kullanan otomatik bir işlem planlama programı geliştirmiştir. Bu çalışmadaki profil tanımlama metodu, iş parçası profilini temsil etmek için benzer bir alan matrisi kullanmıştır. Bu metod, işlem planlamada daha az hafıza ve güç gerektiren bilgisayarların kullanımına imkan sağlar. Burada kullanılan metod, parça profilini tanımlamak için kodlama sistemi ile birleşik iki boyutlu bir matris kullanır [62].

#### **2.4.2. İşlem planlamada uzman sistemler**

GARI, bilgi tabanı 50 kuraldan oluşan yapay zeka esaslı bir BDİP sistemidir [63]. Sistem, prizmatik parçaların alt grubu ile ilgilenir ve tüm yüzeyleri temel eksene paralel olan parçaların işlem sıralamasını oluşturur. İşlenecek parçalar, yüzeyler, delik büyütme, oyuklar ve delikler gibi ilgili özelliklerin bir kümesi olarak modellenir. Parçalar, genel parça bilgisi, boyutlar ve her özelliğin

tasviri olmak üzere üç kategoriye ayrılır. Sistemde bilgi temsili IF-THEN kurallarına göre hazırlanmıştır. Programlama dili MACLISP'dir.

TOM, Paskal'da yazılmış ve VAX-11 bilgisayarda çalışan prosedürel kural esaslı bir BDİP sistemidir. Sistem, COMPAC olarak adlandırılan BDT sistemden tasarım verilerini kullanıcı yardımı veya dönüştürme yoluyla giriş kabul etmek için tasarlanmıştır. Sonuçlar, EXAPT parça programı ile birleştirilir. TOM, dönüştürme karmaşası nedeniyle sadece deliklerin planlanmasını ele almıştır [64].

TIPPS, YZ ve karar ağacı yaklaşımını kullanarak BDT ve üretken BDİP'i bütünleşik sisteme birleştirir. Sistem, parça verileri girişi olarak bir BDT sınırı temsili ve prosedürel bilgiyi tasvir etmek için İşlem Bilim Bilgisi olarak adlandırılan özel bir dili kullanır. Sistem, delme ve frezeleme operasyonlarını planlamak üzere geliştirilmiştir. Burada kullanıcı, terminal kursorünü kullanarak imalat işlemlerini, operasyon sıralamasını, işleme parametrelerini ve zaman tahminini belirlemek için işlenecek yüzeyleri işaretler [65].

Barkocy ve Zdeblick'in geliştirerek CUTTECH adını verdikleri sistemde, dönel parçalar için kesici seçimi, operasyon sıralaması ve işleme parametrelerinin hesabı yapılır. Sistem, bilgi edinme, bilgi tabanı, veri tabanı ve sonuç çıkarma mekanizması modüllerinden meydana gelir [66].

EXCAP sistemi, dönel parçaların işleme operasyonlarını sıralamak üzere UMIST'de geliştirilen bilgi tabanlı bir işlem planlama sistemidir [67]. EXCAP prototiplerinin EXCAP-A, EXCAP-Y ve EXCAP-P gibi üç versiyonu geliştirilmiştir. EXCAP-A, sınırlı miktardaki tornalama operasyonlarının işlem sıralamasını belirlemek üzere AL/X olarak adlandırılan ticari bir uzman sistem kabuğu kullanmıştır. EXCAP-Y, Pascal/York Portable Prolog sistemi bileşimidir. Sistemin bu versiyonu, daha fazla tornalama operasyonları için otomatik işlem planları üretecek kapasitededir. Sistemin son versiyonu olan EXCAP-P ise, öteki versiyonların fonksiyonlarına ilaveten kama kanalı ve kör delik gibi özellikleri kapsayan operasyonları da yerine getirir. EXCAP-P de POPLOG ve GKS (Graphics Kernel System) gibi iki araç kullanılmıştır. POPLOG, POP11, Prolog ve Lisp gibi üç yapay zeka dilinin bileşimi ile yazılmıştır. GKS, bilgisayar grafik uygulamaları için kullanılan milletlerarası bir standarttır. Sistem, bitmiş parçadan geriye, ilk parça şekline doğru planlama yapar. Parça, yüzey, radyüs ve silindirler gibi temel geometrik elemanların düzenlenmiş sıralamasını kullanarak tasvir edilir.

SIPP, talaş kaldırarak üretilecek parçalar için geliştirilen bilgi tabanlı üretken bir işlem planlama sistemidir. Çeşitli tezgah işlemleri ve işlenecek farklı yüzey karakteristikleri hakkında bilgi içeren bilgi tabanı, en az maliyetli ilk

araştırma stratejisini kullanan kontrol sisteminden meydana gelmiştir. Sistemde bilgi, hiyerarşik çerçevelerin kullanımı ile sunulmuştur. Kontrol stratejisi, en az maliyet ilk araştırma yaklaşımına dayanmıştır. Sistem, işlenecek yüzeylerin her biri için işlem planları oluşturarak parçanın tümü için gereken planı hazırlar. SIPP, operasyon tipine bağlı olarak, işlemdeki tolerans ve geometri ile ilgili sınırların uygunluğuna karar verdiğinde, parça için gereken plan hazırlanmış olur. Programlama dili Prolog'dur [68].

Iwata ve Sugimura, CIMS adlı uzman sistemi geliştirmişlerdir. CIMS, parçaların teknik ve teknolojik bilgilerini içeren ve üç boyutlu katı modelini oluşturan bir alt sistemden ve taslak malzeme ve model bilgilerine bağlı olarak işleme planını oluşturan bir alt planlama sisteminden oluşur. Sistemin bilgi tabanı üretim kurallarına dayanır [69].

PROPLAN sistemi, diğer BDT/BDÜ paketleriyle bağlantı kurmak üzere tasarlanmış üretken bir işlem planlama sistemidir [70]. Sistem, bir BDT pakete, SD arayüze ve bir tahmin etme programına bağlanmıştır. Sistem, öncelik sırası belirlenmiş bir düzeni kullanarak, daha kritik özelliği bulmak için bitmiş parça modelini tecrübe ederek çalışır. Sonra seçilen kritik özellik, takım ve tezgah listesini içeren imalat işlemini organize etmek üzere analiz edilir. Elde edilen işlem, ihtiyaç duyulan işlemler listesine yerleştirilir ve parça modeli değiştirilir. Tüm işlemler model için tekrarlanır ve seçilen takımla birlikte imalat görevlerinin son listesi ve tezgahlar sağlanır. Sistem, sac metal parçalar ve dönel parçalara uygulanır.

SIPS (Semi Intelligent Process Selector,1986) sistemi, enaz maliyet ilk stratejisi kullanır. SIPS ile SIPP'in arasındaki fark şudur: SIPP bilginin sunulmasında çerçeveleri kullanırken SIPP, bunun yerine hiyerarşik bilgi kümesi sunma tekniğini kullanmıştır. LISP'te yazılmış olan bu sistemde bilgi statik bilgiler ve problemi çözen bilgiler olmak üzere iki kategoriye ayrılmaktadır. Statik bilgiler, nesnelere 3-B temsilini, problemi çözen bilgiler ise, hiyerarşik bilgi kümesini tasvir eder [71].

Daha önce açıklanan GARI ile parçanın tanımlanmasında benzerlik taşıyan bir başka sistem ise Hi-MAPP (Hierarchical and Intelligent Manufacturing Automated Procces Planner)'dir. Sistem, GARI'ya göre iki noktada farklılık gösterir. Bunlardan ilki Hi-MAPP öncelikle planın özetini hazırlarken, GARI, öncelikle detaylı bir plan hazırlar. İkincisi ise Hi-MAPP sistemi GARI'nın tersine parça özelliklerinin işlenmesindeki önceliğin tanımlanması ile hiyerarşik bir planlama oluşturur. Sistem, operasyon cinsini belirleyen tezgah ve talaş tiplerini, destekleme yüzeylerini, kesici ve işleme parametrelerini içeren 45 üretim kuralından oluşmuştur [72].



Alting, Zhang ve Lenan, silindirik parçaların işlem planlarını oluşturmak üzere XPLAN adlı kullanıcı etkileşimli bir uzman işlem planlama sistemi geliştirmişlerdir. Sistem, taslak parça tasarımı, operasyon seçimi, maliyet ve zamana bağlı olarak alternatif takım ve tezgah seçimi, taslak ve araparça modellere göre uygun bağlama metodu, karmaşık silindirik yapıları için işlem sıralaması optimizasyonu ve uygulanacak operasyonlar için tahmini zaman hesaplamaları gibi işlemleri yerine getirir. Sistemin çıktısı, işlenecek çap ve boy, kesici ve bağlama elemanları ve uygun takım tezgahlarını içeren bir işlem plan kartıdır [73].

XPLANE (eXpert Process PLANning Environment), imalatın bir kısmında işlemlerin planlanması için geliştirilmiştir. Sistemde uygun talaş kaldırma işlemleri, bunların sıralaması ve gereken takımlar, otomatik olarak seçilmektedir. Yapılan işlemin özellikleri, katı modellerin sınır tasviri ile oluşturulan üretimlerden tanınır ve seçilir. XPLANE, BDT/BDÜ sisteminin bir kısmı olarak kullanılır. Sistemin bilgi tabanı takım tezgahları, kesici takımlar ve üretim kuralları hakkındaki bilgileri içerir. Sonuç çıkarma mekanizması, işleme operasyonlarını belirlemek için geriye zincirleme prensibi kullanır [71].

GIPPS, bir BDT sistemden parça tasvirini alan üretken uzman işlem planlama sistemidir. Sistem bir parçanın imalatı için işlem planını otomatik olarak yapar. Sistem, orta ölçekli işletmelerde hidrolik valf bloklarının işlem planlarını ele almıştır [74].

SAPT, DESIGNER US'nin bir parçası olarak otomatik işlem plan kartları üreten bir uzman sistemdir. Sistem, ürünlerin, imalat işlemlerinin ve üretim planlama ve kontrol sistemlerinin kavramsal tasarımı ile ilgilidir. Sistemin felsefesi, GT ve model teknolojisinin melez kavramıdır. Burada kavramsal tasarım, model tanıma, fonksiyonel mantık ve ekonomik optimizasyon gibi üç bölümde verilmektedir. Dönel ve prizmatik parçalar, tip şekilleri ve tip teknolojisi sıralama kavramları ile tasvir edilir [75].

1989'da GSCAPP sistemi geliştirilmiştir [22]. GSCAPP sistemi, toplu imalat ortamında konvansiyonel takım tezgahları için planlanan BEPPS sisteminin bir modülüdür. Sistem, parça bilgisi çıkarmak için etkileşimli giriş kademesine sahiptir. İşlem planları uzman imalat mantığı kullanılarak otomatik olarak oluşturulur.

SD tornalama operasyonları için işlem planları oluşturmak üzere BEPPS-NC sistemi tasarlanmıştır [67]. Sistem, Prolog dilinde yazılmıştır. Burada ana amaç, parça bilgisinden otomatik olarak herhangi bir dönel parça için SD verileri içeren komple planlama bilgisi üretmektir.

Lee ve arkadaşları tarafından geliştirilen FEXCAPP, 147 kuraldan oluşan ve prizmatik parçaların işlem planlarını yerine getiren üretken bir BDİP sistemidir. Sistem, önceki yıllarda geliştirilen BDİP sistemlerin bazı problemlerini büyük ölçüde çözmüştür. Çözülen problemler, talaş kaldırma işlemiyle ilgili bilginin temsil edilmesi ve modellenmesi, değişen imalat şartlarının sisteme eklenmesindeki güçlükler, BDT arayüzü ve parça tolerans değerlerinin BDİP sisteme girilmesidir. Sistemde sonuç çıkarma ve bilgiyi temsil üzere OPS5 uzman sistem kabuğu kullanılmıştır. OPS5, global veri tabanı olarak isimlendirilen çalışan hafızada bir sonuca ulaşmada kullanılan üretim kurallarını ve bu kuralları sağlayan olguları içerir. FEXCAPP'te olgular Nesne-Nitelik-Değer (NND) formuyla temsil edilmiştir. Burada her nesne kendi karakteristiklerini taşıyan bir kaç nitelik-değer çiftini içerir. FEXCAPP 3-B geometrik modelleri temsil etmede ROMULUS adlı bir geometrik modelleyici kullanmıştır. ROMULUS' ta Sınır Temsili (ST)' ye dayalı Kanatlı Kenar Veri Yapısı (KKVY) kullanılmıştır. Bu yapı, bir çok işaretleyiciyle birbirine bağlantılı topolojik ve geometrik varlık kayıtlarından oluşan birleşik yapılardan meydana gelir. Bu tür yapıların BDT ve BDİP arayüzü için uygun olmaması nedeniyle sistem için katı modelin kenarlarını ve yüzeylerin topolojisini temsil etmede her yüzeyin kenarlarının yönünü gösteren Yönlendirilmiş Kenar Veri Yapısı (YKVY) adlı özel bir yapı hazırlanmıştır. Sistem, geometrik modelleyici, ilk işlemci, sonuç çıkarma mekanizması, rota kartı, son işlemci ve SD kodu oluşturma modüllerinden meydana gelir. Geometrik modelleyici VAX 11-750, diğer program modülleri ise, IBM-PC/AT bilgisayarlarında uygulanmıştır. Programlama dili Fortran 77' dir [76] .

IRPPSS, dönel parçaların işlem sıralamasının belirlenmesi, işleme parametrelerinin tayini ve kesici takımların seçiminde kural tabanlı bir sistem olarak geliştirilmiştir. Sistemde US yaklaşımının uygulanabilmesi için bilgi temsili kurallara dayalı olarak hazırlanmıştır. Geliştirilen sistem, Prolog dilinde yazılmış olup, bilgi tabanı bilgilerin değiştirilmesinde sistemi etkilememesi için ayrı bir modül olarak sonuç çıkarma mekanizmasından ayrılmıştır [77].

Akıllı Bilgi Tabanlı İşlem Planlama ve Bağlama Sistemi adlı çalışmada esnek ve tam otomatik uzman sistem yaklaşımı kullanan üretken bir BDİP sistem geliştirilmiştir. Sistemde BDT sistemleriyle iletişim kurmak üzere STEP standardı kullanılmıştır. STEP formatın kullanılması işlem planlamayı gerçekleştirecek olan sistemi, herhangi bir BDT sisteme bağımlı olmaktan kurtarmaktadır. Sistem, ayrıca tüm ürünün tanımlanması için veri tabanına geometrik olmayan verilerin girişini mümkün kılar. Program içinde olguları ve kuralları değiştirmede referans yüzeyi, takımlar ve takım tezgahlarını seçmede, operasyon sıralama ve

operasyon seçiminde programlama tecrübesine sahip olmayan kullanıcılara müsaade eden, İngilizce benzeri bir bilgi tabanı tasarlanmıştır. Bilgi tabanı kurallar ve olgular'dan meydana gelir. Bilgi tabanından kuralları ve olguları okuyabilme yeteneğine sahip olan sonuç çıkarma mekanizması, sonuçlara ulaşmak için geriye veya ileriye zincirleme ve derinlik ilk araştırması yapabilir. Sonuç çıkarma süreci sırasında sonuç çıkarma mekanizması, NİÇİN sorularına cevap verebilir ve bir sonuca NASIL ulaştığını gösterebilir. STEP formatına 3-B sınır temsili BDT verileri alındıktan sonra sistem, bu bilgiyi Prolog formatına dönüştürür. Ayrıca referans yüzeyi ve her özellik için operasyon seçimi, işleme parametrelerinin hesaplanması ve operasyonun her kademesi için takım ve takım tezgahlarının seçimi ve bağlama işlemi otomatik olarak gerçekleştirilir [78].





### 3. YAPAY ZEKA VE UZMAN SİSTEMLER

Modern endüstrinin karşı karşıya kaldığı en önemli problemlerden birisi, ustalığın ve işlenecek parçaları üretmek için uzman işgücünün eksikliğidir. Bu problem, SD takım tezgahları ve bilgisayar yardımlı SD parça programlamanın geliştirilmesiyle bazı alanlarda çözülmüştür. Bununla beraber önemli tecrübeye dayalı bilgi ve tecrübe gerektiren endüstride, hala otomatik olarak yapılacak birçok görev vardır. Normalde tecrübeli tasarımcılar, işçiler ve planlayıcılar tarafından yerine getirilen bu tür görevlerin otomatize edilmesi için gelişmiş **deneyimli** bilgisayar programlarının kullanılması gerekir. Örneğin, bir atelyede mekanik parçaların imalatının önemli bir bölümünü işlem planlama oluşturur. İşlem planlama, genellikle planlayıcı olarak deneyimli bir insan ve önemli derecede tecrübeye dayalı bilgiye ihtiyaç duyan çok karmaşık bir görevdir. Son yıllarda bilimde ve endüstride YZ'nin bazı pratik sonuçları, mühendislikteki bazı sorgulama faaliyetlerini otomatize etmek ve belirli bir alanda uzman bilgiyi elde etmek için YZ tekniklerinin kullanılabileceğini ortaya koymuştur [68]. YZ esaslı uzman sistemler, normal olarak uzman insanı gerektirdiğini düşündüğümüz bir sahada, iyi bir performansa ulaşmak için özel problem bilgisi kullanır. Alan bilgisi, çözümü düşünülen probleme ait belirli bir formatta temsil edilen bilgidir. Bu nedenle sistemden bazı önemli beklentiler vardır. Uzman sistem yaklaşımı, imalat işleminde içerilen faaliyetlerin bazılarını otomatize etmek için başarılı bir şekilde kullanılabilir. Bu faaliyetlerden birisi de işlem planlamadır.

#### 3.1. Yapay Zeka

YZ, geçen on yıl içerisinde dikkatleri üzerine çeken önemli bir teknoloji olarak ortaya çıkmıştır. YZ'de araştırma, insan zekasının işleyişi ile hesaplanabilen yaklaşımların geliştirilmesinde odaklanmıştır. YZ, akıllı insan davranışının hesaplanabilen taklitleriyle ilgilenir [79].

Herhangi bir problemin çözümünde ilk adım, problemin tanımlanmasıdır. Bu kabullenme, hem geleneksel algoritmik programlama, hem de YZ programlama teknikleri için geçerlidir. 1970'li yıllarda, "YZ'nin ilk araştırma aşamasında eğer bir problemin çözümü mümkün olmamış ise bu problem ancak YZ ile çözülebilir" şeklinde bir kanaat oluşmuştur. Bu bir

anlamda doğru olmakla beraber, bütün alanlardaki problemlere çözüm sağlayacak tek bir YZ uygulaması henüz mevcut değildir.

YZ, akıllı bir insanın davranış karakteristiklerini taklit etmeye çalışan ve akıllı bilgisayar sistemleri tasarlamakla meşgul olan bir bilgisayar bilimi dalıdır. Başka bir ifadeyle YZ, günümüzde insanlar tarafından yapılan görevlerin bir kısmını yerine getirmek için çeşitli tekniklerin geliştirilmesiyle ilgilidir [80]. Tipik olarak bu görevlerden bazıları aşağıda verilmiştir.

- Bilgisayarda görüntüleme (Computer vision)
- Konuşma tanıma,
- Robotlar,
- Ses tanıma,
- Tabii dil işleme,
- Model tanıma,
- Makine öğrenme,
- Uzman sistemler.

### 3.2. Uzman Sistemler

US, uzman bir insanın davranışlarını taklit etmeye çalışan bir programdır. Bu sistem, akıllı kararlar verme ve alan özel bilgisi kullanma yeteneğine sahiptir [81].

US, YZ problemlerinin çözümünde çok başarılı sonuçların alındığı bir alan olmuştur. US teknolojisinin üzerinde ilk çalışanlardan Prof. Edward Feigenbaum US'yi "Çözümleri için çok iyi yetişmiş uzman kişilere ihtiyaç duyulan oldukça zor problemlerde sonuç çıkarma işlemlerini ve bilgiyi kullanan zeki bir bilgisayar programıdır" şeklinde tanımlamıştır [82].

US'de bilgi, kitap, katalog veya uzman şahıslardan elde edildiğinden bu sistem, bazen **Bilgi Tabanlı Sistem** (Knowledge Based System) veya **Bilgi Tabanlı Uzman Sistem** (Knowledge Based Expert System) diye anılmaktadır [83].

Bir uzman sistem, genellikle aşağıdaki genel karakteristikleri sağlamak üzere tasarlanır:

1-Yüksek performans;

Sistem, ilgili olduğu alandaki problemlere çözüm getirmede yetişmiş uzmanla eşdeğer veya daha iyi özellikte ve sistemin tavsiyeleri de güvenilir olmalıdır.

## 2- Yeterli cevaplama zamanı;

Sistemin her hangi bir problemi cevaplama süresi, uzman kişinin kararını oluşturma süresinden daha kısa olmalıdır.

## 3- Güvenilirlik;

Bir uzman sistem güvenilir olmalıdır.

## 4- Anlaşılabilirlik;

Bir problemin çözümü sunulurken her adımdaki sorgulama açıklanabilmelidir. "Siyah siyahtır, beyaz beyazdır" gibi kesin tanımlamaların yanında bir uzmanın her aşamada yapabileceği açıklamalar gibi uzman sistem de açıklama imkanlarına sahip olmalıdır.

## 5- Esneklik;

Bir US'nin uzmanlığı, içerdiği bilginin zenginliği ile ilgilidir. Bu yüzden sistemde yeni bilginin eklenmesi, değiştirilmesi ve silinmesi için bir mekanizmanın bulunması önemli bir özelliktir. US'lerin önemli özelliklerinden birisi de kuralları etkili ve modüler bir tarzda depolama kapasitesine sahip olmasıdır. Açıklama imkanları sisteme bağlı olmakla birlikte mümkün mertebe basit ve bu niteliğe sahip sistemler aşağıdaki işlemleri yapabilecek özellikte olmalıdır:

- Belirli bir hipotez ve buna karşı olan bütün sebeplerin listelenmesi;

Hipotez, doğrulanması gereken bir **amaç**'tır. Örnek olarak gerçek bir problemde, bir hastada birden fazla hastalık olabileceği gibi birçok hipotez mevcut olabilir.

- Ele alınan problemle ilgili bütün hipotezlerin listelenmesi,
- Bir hipotezin dayandığı sebeplerin sıralamalarının açıklanması,
- Hipotezin doğru olma durumunda tahmin verilmesi,
- Bilgi edinmede kullanıcıya sorulan soruların değerlendirilmesi;

Bu sorular, sorgulamanın doğruluğunu yönlendirmek için kullanılabilir.

Birçok gerçek problemde bütün ihtimallerin hazırlanabilmesi hem çok uzun zaman almakta hem de pahalı olmaktadır.

- Program bilgisinin değerlendirilmesi;

Eğer program sonuç olarak bir hipotezin doğruluğunu veriyorsa, kullanıcı bunun açıklamasını isteyebilmelidir.

Bir uzman sistem aşağıdaki ünitelerden meydana gelir [84]:

- Bilgi tabanı,
- Sonuç çıkarma mekanizması,
- Kullanıcı arayüzü,
- Bilgi edinme.

Bilgi tabanlı sistemler yukarıdaki kısımlardan biri veya ikisinden yoksun olduğu halde, gerçek bir uzman sistem bu kısımların tamamını içermelidir [8]. Bununla birlikte birçok sistem, bu karakteristiklerin tümüne sahip değildir. Bu nedenle araştırma çalışmalarının çoğu, uzman sistemlerin kapasitelerini arttırmak için yürütülmektedir.

**Bilgi tabanı:** Kurallar ve olgulardan meydana gelen ve alan özel bilgisi bulduran programın bir parçasıdır. Olgular, objeler arasındaki ilişkiler, sınıflandırmalar ve açıklamalardır. Kurallar, problem alanının kavramları arasındaki mantıksal ilişkileri tanımlar. Bilgi tabanı, kural tabanlı uzman sistemde aynı zamanda **çalışan hafıza** olarak da adlandırılır. Basit bir örnek olarak, bir caddeden karşıdan karşıya geçme problemini ele alalım. Geçme ve geçmemeyi içeren iki kural aşağıda verilmiş olup, oklar sol taraftaki durumların sağlanmasıyla sağdaki sonuçların sağlanacağını ifade eder.

Işık kırmızıdır  $\Rightarrow$  dur

Işık yeşildir  $\Rightarrow$  geç

Uygulama kuralları IF-THEN formatında aşağıdaki gibi hazırlanabilir:

Kural : Kırmızı \_ Işık

Eğer

Işık kırmızı ise

O halde

Dur

Kural : Yeşil \_ Işık

Eğer

Işık yeşil ise

O halde

Geç

**Çalışan hafıza:** Güncel duruma bağlı olguları içerir. Trafik işaretleri örneğindeki **ışık yeşildir** veya **ışık kırmızıdır** gibi. Olguların biri veya hepsi aynı zamanda çalışan hafızada olabilir. Eğer normal olarak trafik ışığı çalışıyorsa sadece bir olgu hafızada aktif olur. Bununla beraber, eğer ışıktaki hatalı bir fonksiyon söz konusu ise her iki olgunun da çalışan hafızada aktif olması mümkündür. Burada **bilgi tabanı** ile **çalışan hafıza** arasındaki farka dikkat

edilmesi gerekir. Olgular birbirleri ile etkileşim halinde değildir. Bununla beraber, trafik ışıkları hakkındaki kişisel bilgiler her iki olgunun da aktif olması halinde ışıkta olumsuz bir fonksiyonun varlığını göstermektedir.

Eğer çalışan hafızada **ışık yeşildir** fonksiyonu varsa, sonuç çıkarma mekanizmasının dikkate alacağı olgunun yeşil ışık kuralının durum kısmını sağlayıp, bu kuralı hafızaya yerleştirmesidir.

Bütün alt amaçları sağlanmış bir kural, **aktifleştirilmiş kural** olarak isimlendirilir. Birden fazla aktifleştirilmiş kural aynı anda hafızada olabilir. Bu durumda sonuç çıkarma mekanizması, bir kuralı işletmek üzere seçer.

**Sonuç çıkarma mekanizması:** Kuralları yorumlar ve sorgulama fonksiyonlarını gerçekleştirir [85].

**Kullanıcı arayüzü:** Sistem ve kullanıcı arasındaki iletişimi sağlar. Kullanıcı arayüzü genellikle NİÇİN ve NASIL sorularına cevap verme kabiliyetine sahip olup bir açıklama kolaylığını içerir. Açıklama kolaylığı, sistemin muhtemel sorgulama akışını göstererek sistemin tavsiyesinden sonra kullanıcının güvenini arttırmak için bir uzman sistemde zorunludur.

**NİÇİN sorularını cevaplama:** Açıklama, kullanıcının istediği bilginin amacını göstermeden ibarettir. Kullanıcı sisteme "Bu bilgi ile niçin ilgileniyorsunuz? " diye sorduğu zaman, burada amaç kullanıcının orjinal sorusunu içeren bu bilgi parçasının bağlantısını kuran kurallar ve olguları göstermektir. Bu yüzden NİÇİN sorularının cevabı, araştırma uzayında mevcut alt hedeflerden öne, ana hedefe doğru hareket etmek suretiyle başılır. Başvurma sırasında sonuca gitmeye muktedir olmak için sistem sorgulama zincirini sürdürür [86].

**NASIL sorularını cevaplama:** Sistem bir sonucu bulduğunda kullanıcı, bu sonuca **nasıl** ulaşıldığını görmek isteyebilir. **Nasıl** sorusunun cevabına ana amaç ve alt amaçlar arasında sağlanan kurallar ve olgularla ilgili çözüme ulaşan ana amaçtan öne, alt amaca doğru hareket etmek suretiyle ulaşılır.

**Bilgi edinme:** Bilgi tabanındaki kuralları ve olguları düzeltmek, değiştirmek veya ilave etmek için kullanıcıya müsaade eden program kısmıdır. Bu kısım, birçok sistemde fazladan bir özellik olarak bulunur.

### 3.2.1. Bilginin anlamı

Bilgi, aşk gibi herkesin anlamını bildiği, fakat tanımlamada zorluk çektiği kelimelerdendir [83]. Bilgi üç şekilde sınıflandırılabilir;

- İşlem sıralamalı bilgi,
- Deklaratif bilgi,
- Sözsüz ifade olunan bilgi.

**İşlem sıralamalı bilgi**, bir şeyin nasıl yapılacağını ifade etmek için kullanılan bilgidir. Örnek olarak, bir çaydanlıktaki suyun nasıl kaynayacağını bilmesi verilebilir.

**Deklaratif bilgi**, bir şeyin **doğru** veya **yanlış** olduğunu bilmesini ifade eder. "Kaynayan su dolu çaydanlığa dokunmayınız" cümlesi deklaratif bir bilgidir.

**Sözsüz ifade olunan bilgi**, konuşma diliyle tanımlanamadığı için bazen **bilinçsiz** bilgi olarak da isimlendirilir. El ve ayakların veya vücudun herhangi bir organının hareketleri yerine getirmesindeki sebebin ne olduğunu bilmemesi örnek olarak verilebilir.

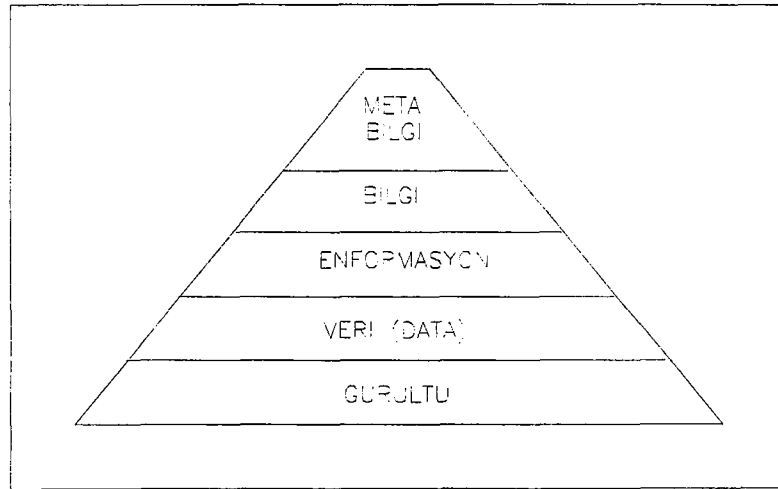
Bilgi, uzman sistemlerde özel bir öneme sahiptir. Klasik sistemlerde Wirth'in aşağıdaki tanımlaması kullanılırken:

Algoritmalar + Veri yapıları = Programlar

Uzman sistemlerde ise aşağıdaki yapı geçerlidir:

Bilgi + Sonuç çıkarma = Uzman sistemler

Giarratano, J., ve Riley, G., bilgiyi hiyerarşinin bir parçası olarak kullanmışlardır (Şekil 3.1). Örneğin piramidin en alt tabakasında ilgili olmayan bilgilerden meydana gelen gürültü mevcuttur. Bir üst düzeyde ilgili elemanlardan meydana gelen veriler (data) vardır. Probleme ilgili işlenmiş veriler, enformasyon olarak adlandırılır. Sonraki düzeyde çok özel enformasyonu tanımlayan bilgi (knowledge) yer alır.



Şekil 3.1. Bilginin hiyerarşisi

Kural tabanlı uzman sistemlerdeki bilgiler, yeni sonuçlar elde etmek için **olgular** tarafından aktif hale geçirilen **kurallar** şeklinde tanımlanır. Olgu terimi veriler veya enformasyon anlamına gelebilir. Uzman sistemler, yazılışlarına bağlı olarak verileri kullanarak gerekli sonuçları elde eder.

Uzmanlık, uzmanların sahip olduğu bilginin özelleştirilmiş çeşididir. Uzmanlık, günlük hayatta kullanılan kitap ve basılı yayınlarda yer almaz. Bunun yanında uzmanlık, uzmanın kendi alanı ile ilgili kesin ve mutlak bilgisidir. Bu bilgi, deneyim yoluyla elde edilen ve uzman sistemde herhangi bir şekilde kodlanan bilgidir.

Üst bilgi, **metabilgi** olarak isimlendirilir ve genel bilgi ve uzmanlık bilgisinden meydana gelir.

### 3.2.2. Bilgi temsil etme teknikleri

US'de bilgi, bir bilgisayarda yerine getirilebilecek şekilde temsil edilmelidir. US'de bilgi temsil etmek için bazı teknikler geliştirilmiştir. Bunlar:

- 1- Predicate (ilişki) mantığı (First Order Predicate Calculus),
- 2- Üretim kuralları,
- 3- Semantik ağlar,
- 4- Çerçeveler (çatılar),
- 5- Obje uyumlu programlama.



### 3.2.2.1. Predicate (ilişki) mantığı

Predicate mantığı, objeler arasındaki ilişkilere göre belirli bir alandaki olguların temsilidir. Bu mantık, bilgi temsil etme vasıtası olarak kullanılır. Bilginin açıklaması mantık cümlelerine göre şekillenir. Prolog dili, predicate mantığı esasına dayanır ve notasyonu aynıdır. Prolog, çok geniş bir ifade topluluğu ile açıklanabilen sembolik bir dildir. Bir ifade, predicate, değişken ve sabit sembollerle açıklanır. Örnek olarak "Yüzey(X) bir deliğe sahiptir" olgusu şu şekilde ifade edilir:

sahip(yüzey(X), bir\_delik).

Bu örnekte **sahip** bir predicate sembolü, **X** bir değişken, **delik** ise bir sabittir. Bu şekilde yapılan temsil, **atomik formül** olarak adlandırılır. Geçerli bir atomik formül iyi tasarlanmış formüldür. İyi tasarlanmış bir formül, bir doğru veya bir yanlış olabilir [87].

### 3.2.2.2. Üretim kuralları

Bilgiyi temsil etmek için kullanılan en popüler format, üretim kurallarıdır. Bilgi, IF-THEN kuralları şeklinde temsil edilir. Şöyle ki;

IF (koşul) THEN (eylem veya sonuç).

Kurallar genellikle iki bölümden meydana gelir. IF bölümü, **koşul** bölümü olarak, THEN bölümü ise, **eylem** veya **sonuç bölümü** olarak adlandırılır. Koşul, doğru ve yanlışları değerlendirmesi gereken tarifler şeklinde yazılır. THEN bölümü, faaliyetleri yerine getirecek ve sonuçlandıracak şekilde bir veya daha fazla faaliyetin bileşimidir. Böyle faaliyetler, bilgi sağlayabilecek, hesaplar yapabilecek, çıktılar üretebilecek ve giriş için kullanıcıya soru sorabilecektir.

Üretim kuralları kümesi, problemi ilk durumundan çözüme doğru hareket etmek suretiyle gerekli dönüşümlere izin verir. Çözümün mevcut durumu bilgi tabanında iddia edilen kurallar ve olgular kümesi ile temsil edilir. Çözüm, bilgi tabanına kuralın bir tarafını benzetmek suretiyle yürütülür. Gerçek bir sistemde herhangi bir zamanda kuralların birkaçının uygulanabileceği muhtemeldir. Bundan dolayı, hangi kuralın çalıştırılacağına dair bir karar vermek için çeşitli kontrol yapılarına ihtiyaç duyulur [88].

Amaç yönlendirmeli problem çözmeye ve kuralların kullanımına örnek olması için aşağıda otomobil problemlerinin çözümüne teşhis koyan küçük bir uzman sistem verilmiştir.

Kural 1:

Eğer

motor yakıt alıyorsa ve  
motor çalışmıyorsa

O halde

problem bujilerdedir

Kural 2:

Eğer

motor çalışmıyorsa ve  
lambalar yanmıyorsa

O halde

problem akü veya kablolardadır

Kural 3:

Eğer

motor çalışmıyorsa ve  
lambalar yanmıyorsa

O halde

problem marş motorundadır

Kural 4:

Eğer

depoda yakıt varsa ve  
karbüratör yakıt alıyorsa

O halde

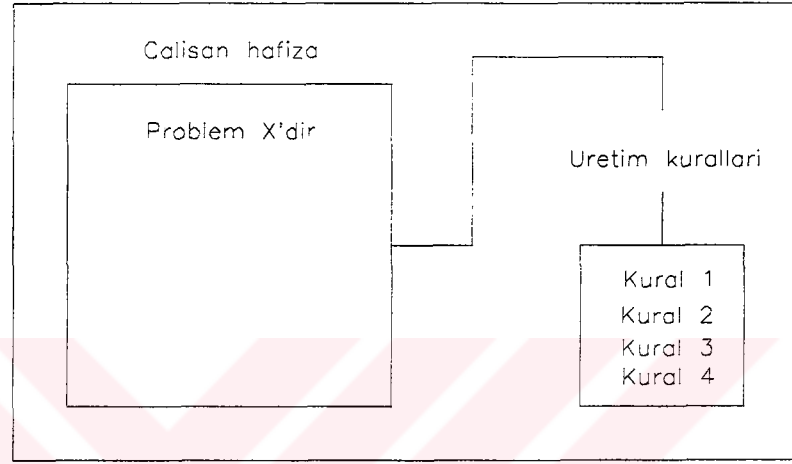
motor yakıt alıyor

Bu bilgi tabanı, amaç yönlendirmeli kontrol altında çalıştırılmak istenirse, en uç düzeydeki amaç Şekil 3.2'deki gibi çalışan hafızaya yerleştirilir. Problem X'dir; X çözüme ait bir değişken olup, herhangi bir durumla uyduğunda problem çözülmüş demektir.

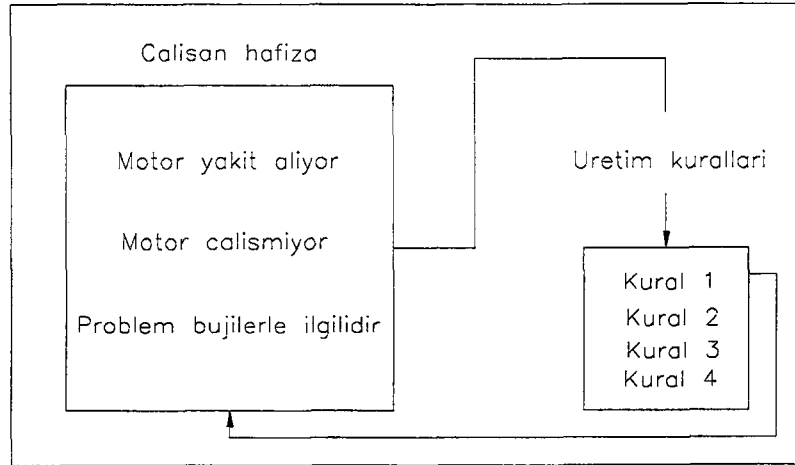
İlk üç kural (Kural 1, Kural 2, Kural 3) çalışan hafızadaki tanımlama ile çakışır. Eğer en düşük numaralı kuraldaki **karmaşa** çözümlenirse, Kural 1

işletilir. Bu X'in buji değerine bağlanmasına sebep olur ve Kural 1'in içeriği çalışan hafızaya Şekil 3.3'te olduğu gibi yerleştirilir.

Sistem, bujilerin kötü olduğu hipotezini işletmek üzere seçer. Kural 1'in iki önermeli olduğuna dikkat edilmelidir. Her ikisi de sonuca ulaşabilmek için doğrulanmalıdır. Bunlar iki alt problemdeki (motor yakıt alıyor ve motor çalışmıyor) problem (bujiler kötü) bütünlüğünü tanımlayan araştırma grafiğinin AND koludur. Kural 4 işletilir ve çalışan hafızaya Şekil 3.4'teki gibi yerleştirilir.

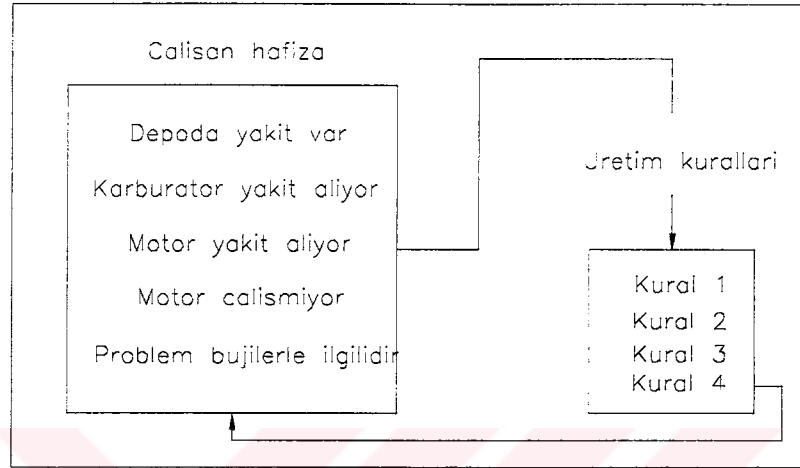


Şekil 3.2. Araba arıza teşhisinde sorgulama başlangıç durumu

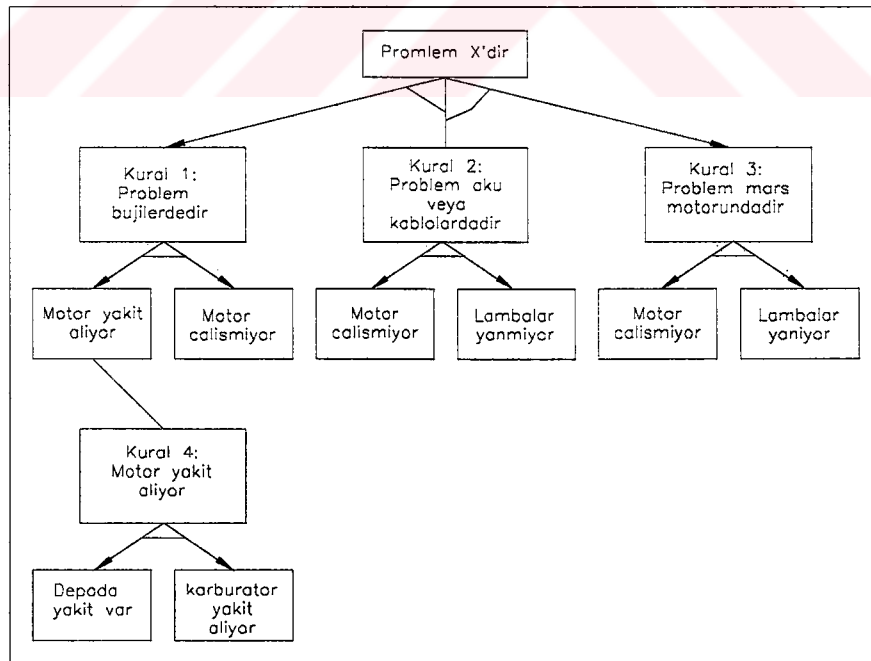


Şekil 3.3. Kural 1 çalıştırdıktan sonraki durum

Bu noktada çalışan hafızada, herhangi bir kuralın sonucuyla çakışmayan üç girdi vardır. Hazırlanan uzman sistem, kullanıcıya bu alt amaçlarla ilgili sorular sorar. Kullanıcı bu üçünü doğrularsa, sistem bujilerin kötülüğünden dolayı motorun çalışmayacağı sonucunu çıkarır. Bu sonuca ulaşmak için Şekil 3.5'te görülen AND/OR grafiği oluşturulur.



Şekil 3.4. Kural 4 çalıştırdıktan sonraki durum



Şekil 3.5. Araba arıza teşhisi AND/OR grafiği [83]

### 3.2.2.3. Semantik ağlar

Semantik bilgi ağı veya kısaca ağ, durum bilgisi için kullanılan klasik YZ temsil tekniğidir [89]. Semantik ağ, bazen **durumsal ağ** olarak da adlandırılır. Durumlar, bir ifadenin doğru veya yanlış olduğunu ortaya koymakla beraber, deklaratif bilginin bir çeşidi olup **olguları** belirler.

Semantik ağlar, ilk olarak dilin anlaşılması ve insan hafızasının tanımlanmasının bir yolu şeklinde, YZ için geliştirilmiştir [90]. Quillian, semantik ağları cümle içindeki kelimelerin anlamlarını analiz etmek için kullanmıştır. O günden bu yana birçok problemin çözümünde bilginin tanımlanması amacıyla kullanılmaktadır.

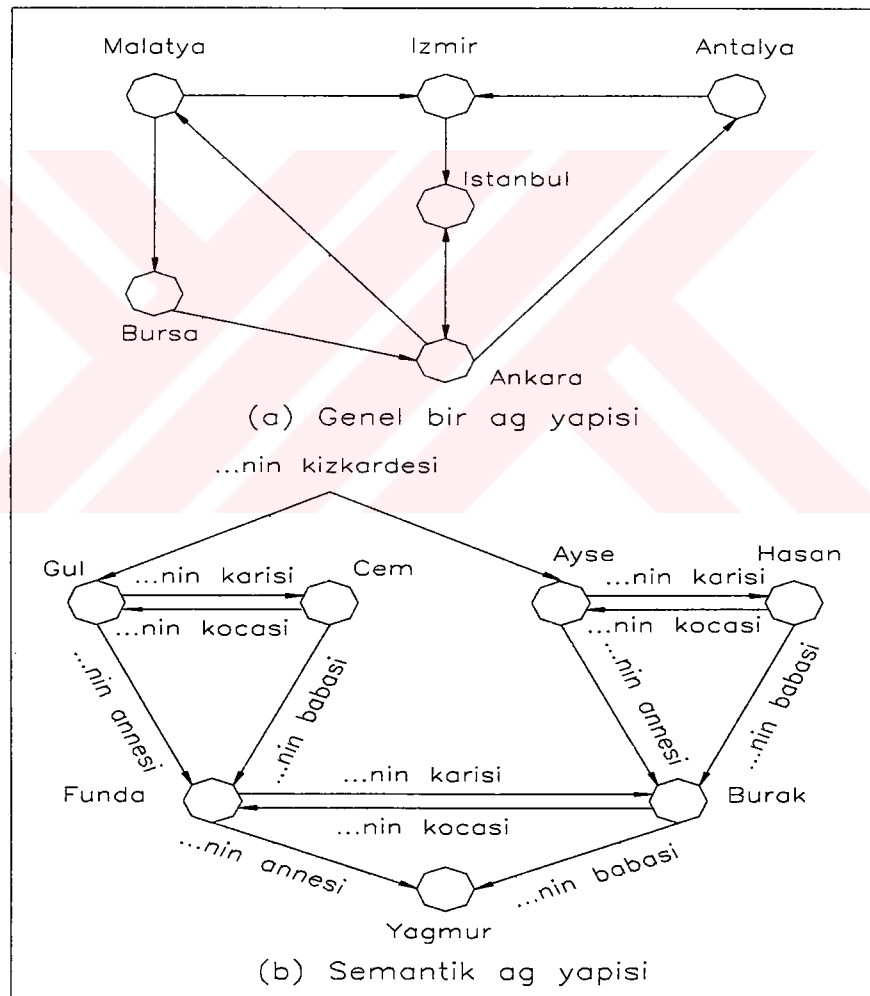
Bir semantik ağ, grafik olarak **düğüm**ler'den (node) ve bu düğümleri birleştiren **bağlantı kolları** veya **kenarlar**'dan meydana gelmiştir. Düğümler bir **nesneyi** tanımlarken, bağlantı kolları düğümler arasındaki ilişkileri tanımlamak için kullanılır.

Şekil 3.6.a' da şehirlerarası havayolu bağlantılarını gösteren genel bir ağ ve Şekil 3.6.b' de bir ailenin üyeleri arasındaki ilişkileri ifade eden semantik ağ verilmiştir. Şekil 3.6.a' da şehirlerarasındaki havayolu bağlantılarında daireler, düğümleri; bunlar arasındaki çizgiler, bağlantıları ve oklar da uçakların yönlerini tanımlamaktadır. Bundan dolayı bu tip ağlara **yönlendirilmiş grafik** ismi de verilir.

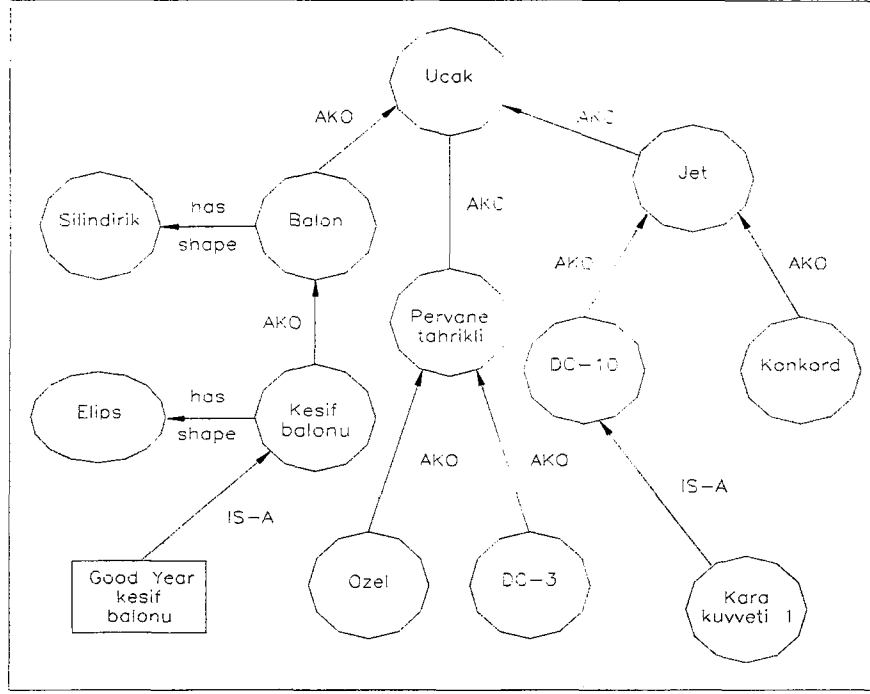
Bilginin organize edilmesi için temel yapı sağlamalarından dolayı, semantik ağlarda **ilişkiler** büyük öneme sahiptir. İlişkisiz bilgi tanımlama, bağlantısız **olgular** bütünüdür. Bilginin ilişkilendirilmesine ait bir semantik ağ, Şekil 3.6.b' de de açıkça görüleceği gibi, aile üyeleri arasındaki ilişkilerin tümünü tanımlamıştır.

Bilginin tanımlanmasında kullanılan belirli ilişki tipleri mevcuttur. Her bir farklı problem için yeni ilişki tiplerinin tanımlanması yerine, alışılmış ilişki tiplerinin kullanımı daha yaygındır. Yaygın olarak kullanılan bağlantı tipleri IS-A (... dır) ve A-KIN-OF (... nın cinsidir)'tır. Bu ilişkiler bazen IS-A ve AKO şeklinde kısaltılarak kullanılır [91]. Şekil 3.7'de IS-A ve AKO'nun kullanımına ait bir örnek verilmiştir. Şekildeki IS-A'nın anlamı, bir örneği tanımlama ve örneğin hangi **sınıfa** ait olduğunu belirlemedir. Sınıfların oluşturulması, bir grup nesnenin matematiksel kavramlar yardımıyla tanımlanmasıyla sağlanır. Her bir gruptaki elemanlar, farklı biçimlerde olmakla beraber, aralarında belirli bir ilişki mevcuttur. Örneğin uçak, tren, otomobil, kamyon gibi taşıtlar, kullanım amacı yönünden farklı olmakla beraber, aralarındaki ilişki **taşımacılık aracı** olmasından ibarettir.

IS-A, münferit küme elemanı ilişkilerini tanımlar; AKO ise, elemanların içinde buldukları sınıflar arasındaki bağlantıları belirleyen bir uygulamadır. Bir başka ifadeyle, AKO bütünü teşkil eden düğümleri birbirlerine, IS-A da münferit bir elemanı genel bir sınıfa bağlamaktadır. Şekil 3.7'de, görüldüğü gibi genel sınıflar daha üst düzeyde iken, özel sınıflar yapının tabanında yer almaktadır. AKO'nun ifade ettiği genel yapı okunun işaret ettiği düğüm, **süper sınıf** olarak adlandırılır. AKO'nun tanımladığı sınıf, bir çoğulu ifade ediyorsa, ARE bağlantısı kullanılır. Bir sınıftaki nesnelere, genel olarak bir veya daha fazla niteliğe sahip olabilir. Her nitelik de bir **değer**'e sahiptir. Nitelik ve değer beraberliği **özellik** olarak adlandırılır.



Şekil 3.6. İki tip ağ yapısı



Şekil 3.7. IS-A ve A-Kind-Of ilişkilerini gösteren semantik ağ

Örneğin bir balon boyut, ağırlık, biçim, renk ve benzeri niteliklere sahipse, biçim niteliğinin değeri eliptir. Diğer bir ifadeyle, balon elips şeklinde bir özelliğe sahiptir. Semantik ağlarda kullanılan diğer bağlantılar ise IS ve CAUSE'tır. IS (...dır) bağlantısı değeri tanımlarken, CAUSE (sebepl) durum bilgisini ifade eder. Semantik ağlar yaygın olarak kullanılmalarına rağmen, uygulamalarında bazı güçlüklerle karşılaşmaktadır. Bu güçlüklerle ilgili bazı durumlar aşağıda açıklanmıştır.

- Aşağıda verilen herhangi bir düğümün tam olarak neyi ifade ettiği veya niçin oluşturulduğu açık değildir. Eğer bir düğüm **sandalye** olarak tanımlanmış ise, acaba bu aşağıdakilerden hangisini ifade etmektedir?

- Özel bir sandalye,
- Sandalyelerin sınıflandırılması,
- Sandalye kavramı,
- Bir toplantıda herhangi bir şahsa ait bir sandalye.

- Diğer bir problem de, özellikle soruya verilen cevabın olumsuz olması karşısında araştırılması gereken düğüm sayısının fazlalığıdır.



- Semantik ağların insan beyninin çalışma fonksiyonlarına göre düzenlenmesi durumunda, bir insan beyni bir bilgiyi tanımlamak için nasıl faaliyet gösteriyorsa, Semantik ağdan da tarama sırasında aynı yolu izlemesi beklenir. Bir insanın beyninde  $10^{10}$  sinir düğümü ve bunlar arasında da  $10^{15}$  adet bağlantının olduğu düşünülürse, böyle bir ağın kurulmasındaki güçlük ortaya çıkar.

- Semantik ağlar bilgiyi mantıksal olarak tanımlamaz. Bu yüzden semantik ağla yapılan herhangi bir mantık tanımlaması, belirli bir sandalyeyi, bazı sandalyeleri, bütün sandalyeleri belirlerse de; hiçbir sandalyeyi tanımlamayabilir de.

- Semantik ağlar, sezgisel olarak yetersizdir; çünkü ağın etkili şekilde nasıl araştırılacağı hususundaki sezgisel bilgiyi yönlendirici bir yol yoktur. Sezgisel bilgi, probleme çözüm bulmadaki kural olarak tanımlanır. Fakat algoritmaların çözümde sağladığı garantiyi vermez. Bununla beraber sezgisel bilgiler YZ'de önemli bir yer tutmaktadır.

- Bağlantı isimlerinin standart tanımlamalarının olmaması da başka bir sınırlamadır. Örneğin bazı kaynaklarda hem genel, hem de münferit ilişkiler için IS-A kullanılır [92]. ART uzman sisteminde, sınıflararası bağlantılarda IS-A kullanılırken, münferit bir nesnenin sınıfla bağlantısı için INSTANCE-OF (...nın örneği) tercih edilmektedir. Yaygın olarak kullanılan bir bağlantı şekli de HAS-A'dır. Bu bağlantı, tipi bir sınıfın bir alt sınıfı ilişkisini tesbit etmek için AKO yerine kullanılmaktadır. HAS-A (sahiptir) ile ilgili bir örnek aşağıda verilmiştir.

Araç HAS-A motor (Araç motora sahiptir).

Araç HAS-A lastikler (Araç lastiklere sahiptir).

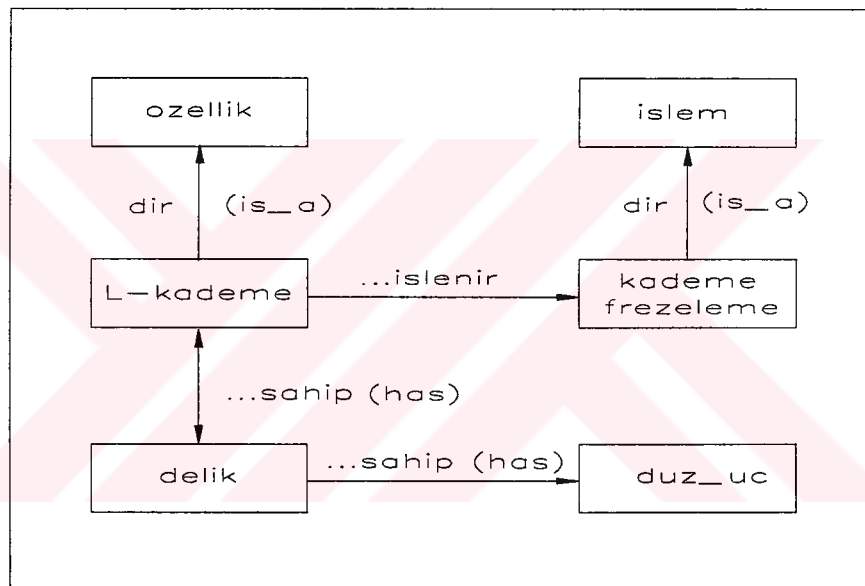
Araç IS-A Ford (Araç Ford'dur).

Nesne-Nitelik-Değer (NND) üçlüsünün kullanılmasıyla semantik ağ kullanımının basitleştirilmesi mümkündür. NND (Object-Attribute-Value, OAV) üçlüsü, semantik ağ içinde tüm bilgilerin karakterize edilmesi için kullanılabilir. MYCIN, uzman sistemde enfeksiyon hastalıklarının teşhisinde kullanılmıştır. Ayrıca bu üçlü yardımıyla bilginin tablo şeklinde listelenmesi de mümkündür. Böyle bir örnek Şekil 3.8'de verilmiştir. Semantik ağları anlamak çok kolay olmasına rağmen, bunları uygulamak oldukça zordur [8].

Şekil 3.9'da L-kademe özelliğini açıklayan tipik bir semantik ağ örneği verilmiştir. Bu şekil bir olguyu açıklar. Olguyu meydana getiren **L-kademe** düz uçlu deliğe sahip bir özelliktir. Bu özellik, kademe frezeleme ile imal edilebilir ve burada yüzey frezeleme, bir işlemdir.

Nesne	Nitelik	Değer
Elma	Renk	Kırmızı
Elma	Tip	Amasya
Elma	Miktar	100
Üzüm	Renk	Sarı
Üzüm	Tip	Sultan
Üzüm	Miktar	500

Şekil 3.8. NND tablosu



Şekil 3.9. Kademe özelliğinin semantik ağla temsili

#### 3.2.2.4. Çerçeveseler (Çatılar)

Birçok YZ uygulamasında kullanılan diğer bir şema tipi de çerçeveselerdir. Oluşturulmalarına göre çerçeveselerin sıralamasını yapan diğer bir şema şekli de skriptlerdir [93]. Çerçeveseler, nesnelerin tanımlanmasında verilen yazılımları doğrudan kullanan uygun bir yapıyı hazırlar. Bir çerçevenin temel karakteristiği, çok geniş bilgi özelliği olan bir nesnenin belirli sınırlar içindeki özellikleriyle

tanımlanmasıdır. Çerçeveseler, makina ve otomotiv endüstrisi gibi birçok parçadan meydana gelmiş bütünlerin tanımlanmasında en uygun seçimlerden biridir.

Semantik ağlarda olduğu gibi, çerçeve tabanlı sistemlerde de herhangi bir standart yoktur. Bununla beraber çerçeve için tasarlanmış FRL, SRL, KRL, KEE, HP-RL gibi özel amaçlı dillerin yanında, Lisp ve Prolog gibi programlama dillerinin özellikleri yardımıyla çerçeve oluşturmak da mümkündür.

Bir kaydın alanları, çerçevenin **kanalı** değerleri ise **doldurma elemanları** olarak adlandırılır. Şekil 3.10'da bir otomobile ilgili çerçeve verilmiştir.

Bazı çerçeveseler Şekil 3.10'da görüldüğü gibi basit değildir. Çerçeveselerden sağlanacak fayda, hiyerarşik çerçeve sistemleri ve bunların özelliklerine bağlıdır. Doldurucular, kanal ve mevcut çerçeveselerin kullanımı ile, etkili bir bilgi tanımlama sistemi kurulabilir. Çerçeve tabanlı uzman sistemler, bilginin sebep ve etki ikilisi ile organize edilmelerinden dolayı, sebebe dayalı bilgi tanımlamada çok kullanışlıdır.

KEE gibi bazı çerçeve tabanlı programlar, elemanların kanallarda kaydedilmesine imkan tanır. Çerçeve kanallarında, kurallar, grafikler, yorumlar, deneme enformasyonu, kullanıcılar için sorularla diğer çerçeveseleri ve durumları içeren hipotezler bulunur. Çerçeveseler, çoğunlukla genel veya belirli bir bilginin tanımlanması için tasarlanırlar. Şekil 3.11'de genel bir çerçeve verilmiştir. Doldurucular, biçim kanallarındaki gibi, değer alanları veya isim kanallarındaki özellik gibi değerler olabilir.

<b>Kanallar</b>	<b>Doldurucular</b>
Üretici	Tofaş
Model	Tempra SX-AX
Yıl	1995
Hareket iletimi	Otomatik
Motor	Benzinli
Lastik adedi	4
Renk	Beyaz

Şekil 3.10. Otomobil çerçevesi

<b>Kanallar</b>	<b>Doldurucular</b>
İsim	Özellik
...nin özelliği	Nesnenin çeşidi
Cinsi	(Araba, Gemi, Ev) if-added: ÖZELLİK EKLEME işlemi
Sahibi	Varolan: Hükümet if-needed: SAHİBİNİ BUL işlemi
Yerleşim	(Ev, İş, Değişken)
Durum	(Kayıp, Zayıf, İyi)
Garanti altında	(Evet/Hayır)

Şekil 3.11. Özellik için genel çerçeve

Genellikle üç tip olan bağlantılar, **işlem sıralamalı bağlantılar** olarak isimlendirilir. Var olan değer uygun olmadığı veya herhangi bir değer atanmadığı değere ihtiyaç duyulduğunda IF-NEEDED (eğer ihtiyaç varsa) tipi kullanılır. İnsan beyninin bazı hallerini modellemelerinden dolayı, mevcut değerler çerçevelerde birincil öneme sahiptir.

Kanala yeni bir değer ilave edilmesi gerektiğinde IF-ADDED (eğer ilave edilecekse) işlemi çalıştırılır. Çeşit (tip) kanalında, gerekli olduğu zaman yeni bir özelliği eklemek amacıyla ÖZELLİK EKLEME işlemi aktif hale getirilir.

Kullanılmayan herhangi bir değer kanalından çıkarılması söz konusu olduğunda IF-REMOVAL (eğer çıkarılacak ise) işlemi çalıştırılır. Şekil 3.11, 3.12, 3.13 ve 3.14'te **...nın çeşidi ...dır** ilişkilerinin çerçeveler arasındaki hiyerarşik bağlantısının nasıl kurulduğu gösterilmiştir.

Şekil 3.11 ve 3.12 genel çerçeveleri ifade ederken, Şekil 3.13 araba ile ilgili olan özel bir çerçeveyi ve Şekil 3.14 ise bir takım tezgahı ile ilgili çerçeveyi tanımlamaktadır. Bilginin tanımlanmasında sağladığı fayda ve kolaylıkların yanında çerçevelerin kullanımında ve uygulanmasında bazı güçlüklerle de karşılaşılır. Bunları şu şekilde sıralamak mümkündür:

- Kanalların iptal edilmesinde ve değiştirilmesinde bazı kısıtlar mevcuttur [94]. Bu probleme örnek teşkil edecek bir fil çerçevesi Şekil 3.15'te verilmiştir. İlk bakışta çerçevenin makul olduğu ve fillerin genel tanımlamasını yaptığı algılanabilir. Buna karşılık herhangi bir sebeple ayaklarından birisini kaybeden

(Clyde) üç ayaklı filin tanımlanması, bu çerçeve ile mümkün olmamaktadır. Çünkü, çerçevenin tanımladığı fil 4 ayağı olan memelileri konu almaktadır. Ayak problemi, çerçevenin 3,2,1; hatta hiç ayağı olmayan filler için düzenlenmesiyle çözülebilir. Bunun yanında renk problemi ortaya çıkabilir. Kahverengi veya siyah renkli filler, çerçevenin tanımlamasına göre "fil sayılmayacak mı? " sorusunu akla getirebilir.

<b>Kanallar</b>	<b>Doldurucular</b>
İsim	Otomobil
...nin özelliği	Özelliğin çeşidi
Cinsi	(Sedan, spor,değişebilir)
Üretici	(GM, Ford, Tofaş, Rnault)
Yerleşim	Değişken
Tekerler	4
Hareket aktarımı	(Manual, otomatik)
Motor	(Motorin,Benzin)

Şekil 3.12. Otomobil özelliğinin genel alt çerçevesi

<b>Kanallar</b>	<b>Doldurucular</b>
İsim	Gökçenin arabası
...nin özelliği	...otomobildir
Üretici	Renault Mais
Sahibi	Gökçe
Yerleşim	Manual
Motor	Benzin
Hareket aktarımı	İyi
Garanti altında	Evet

Şekil 3.13. Bir araba çerçevesi örneği

Çerçevenin Adı: takım_tezgahı	
takım_tezgahı_adı	freze_tezgahı_fu1
x_eksende_hareket_kapasitesi	500
y_eksende_hareket_kapasitesi	400
z_eksende_hareket_kapasitesi	350
ilerleme_miktarı	1-750
işlemsiz_hızlı_hareket	1200
devir_sayısı	28-2400
maksimum_takım_çapı	180
maksimum_takım_boyu	250
kontrol	elle

Şekil 3.14. Takım tezgahı çerçevesi

İsim	Fil
Özellik	Bir çeşit memeli
Renk	Gri
Ayak	4
Hortum	Silindir

Şekil 3.15. Fil çerçevesi

- Çerçeveler, genel tanımlamanın yanında, özel ve basit olarak nesnelerin tanımlanmasını da ifade eder. Örnekteki filin özelliği memeli oluşudur. Bu özelliğe sahip birçok varlık vardır. İnsanlar, atlar, koyunlar ve tavşanlar bu özelliğin kullanımına en güzel örnek olacak varlıklardır.

- Bir kanalın değiştirilmesi veya iptal edilmesi sonucunda, kanalın bütün özelliklerinin değiştirilecek olması önemli problemlerden bir diğeridir.

### 3.3. Uzman Sistemlerde Sonuç Çıkarma

Bilgi tabanlı problem çözmede, çözümü başarmanın genellikle açık ve tek bir metodu yoktur. Araştırma, her kademedeki mümkün olan başarıyı değerlendirmek için, değişik yolları keşfetmek zorundadır. Problem çözme

metodunun kontrol yapısı, yol seçmek için genel stratejiyi tanımlar [88]. US'de problemlerin çözümünde ortaya koymaları gereken sorgulama için sonuç çıkarma metodları oldukça önemlidir.

### 3.3.1. Ağaçlar, kafesler ve grafikler

Ağaç, enformasyon veya bilginin kayıt edildiği düğümlerle bu düğümleri birbirine irtibatlandıran bağlantılardan (dal) meydana gelen hiyerarşik bir veri yapısıdır. Bazı kaynaklarda dallar, **kenar**; düğümler ise, **köşeler** olarak isimlendirilir. Şekil 3.16'da, her bir düğüm için 0, 1 veya 2 bağlantısı olan ikili ağaç yapısı verilmiştir.

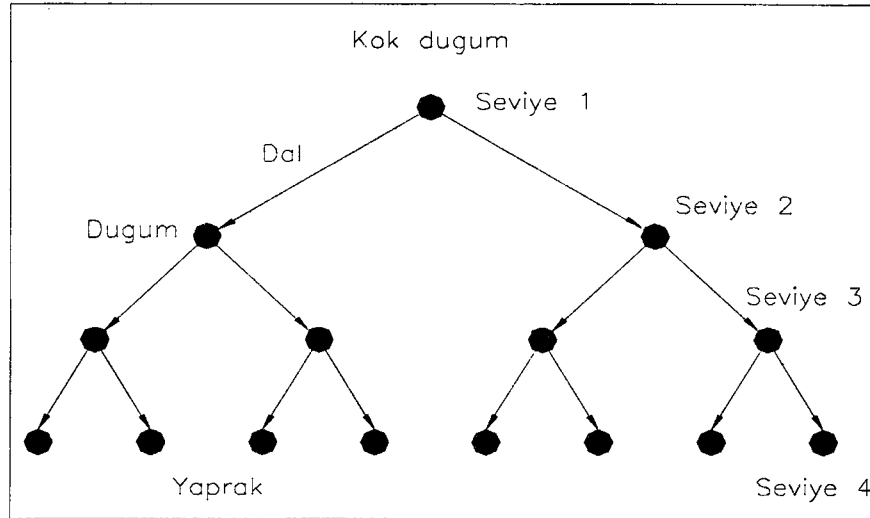
Ağaç, bir anlamda semantik ağların özel bir çeşididir. İkili ağaçların genel kullanımında, her bir düğümün en fazla iki dalı vardır ve her ikisi de birbirinden ayrılmıştır. Eğer bir düğüme bir üst seviyeden birden fazla bağlantı varsa, bu bir ağ uygulaması meydana getirmektedir. Ağaçlar, genel bir matematiksel yapının özel bir durumu olan **grafikler** şeklinde tanımlanabilir. Bir grafik, elemanları arasında herhangi bir bağlantı olmaksızın tanımlanabildiği gibi, düğümler arasında birden fazla bağlantı kurularak da oluşturulabilir.

Grafiğe en güzel örnek haritadır. Şehir ve yerleşim alanları düğümleri, yollar da dalları veya bağlantıları tanımlar. Bağlantılarda yönleri tanımlayan **oklar** ve bazı karakteristikleri tanımlayan **ağırlıklar** vardır. Bir grafikteki ağırlıklar, herhangi bir enformasyon şeklinde tanımlanabilir. Örneğin bir grafik demir yollarındaki rotayı tanımlıyorsa, ağırlıklar şehirler arasındaki mesafeler, bilet fiyatları, harcanacak yakıt ve benzeri şekilde ifade edilebilir.

Ağaç ve kafesler, hiyerarşik yapılarından dolayı nesnelere sınıflamada kullanışlıdır. Ataları ile kuşaklar arasındaki bağlantıları ifade eden aile ağaçları bunlara en güzel örneklerdir. Bu yapıların bir başka uygulaması ise karar oluşturmada kullanılan **karar ağaçları** veya **karar kafesleridir**.

Uygulamada genellikle ağaç ve kafes yerine **yapı** kullanıldığı için, yukarıdaki terimlere karşılık **karar yapıları** deyimini kullanılmaktadır. Karar yapısı hem bilgi tanımlama işlemi, hem de bilgi hakkında bir sorgulama metodudur. Şekil 3.17'de tezgah seçimleriyle ilgili bir karar ağacı verilmiştir. Şekil incelendiğinde düğümlerin soruları, dalların sorulara verilen **evet** veya **hayır** cevaplarını; yaprakların ise, tahmini tezgah seçiminin ne olduğunu içerdiği görülmektedir. Eğer probleme verilmesi gereken cevaplar ikili ise, etkili bir karar ağacını oluşturmak çok kolaydır.





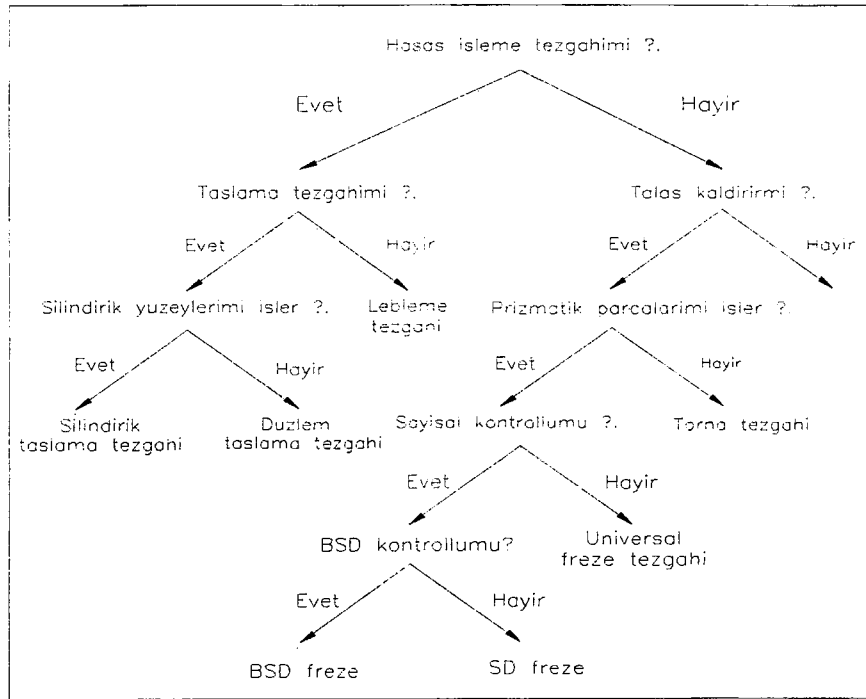
Şekil 3.16. İkili ağaç yapısı

Her bir soru, ağaç içinde bir seviye aşağıya sorulur. Bu yapıda bir sorunun muhtemel cevap sayısı tektir. Sorulan soru iki ise, muhtemel 4 cevaptan biri doğrudur. Üç soru sekiz muhtemel cevaptan birine karar verir. Bu açıklamaya bir başka tanımlama getirilirse, görülen o ki,  $n$  soruya verilen cevap sayısı  $2^n$  dir. Örneğin hayvanların gruplandırılmasında 15 soru ile 32768, 25 soru ile de 33.554.432 hayvandan biri belirlenir.

Karar ağaçlarının bir başka kullanışlı özelliği de **kendi kendine öğrenmeyi** yapabilmesidir. Eğer tahmin yanlış ise, işlem yeni bir soru için sıralamaya çağrılarak sınıflama sorusu düzeltilip tekrar **evet** ve **hayır** cevapları alınmak üzere soru tekrarlanır. Böylece yeni düğüm, dal ve yapraklar dinamik olarak oluşturularak ağaca yerleştirilir.

Basic programlama diliyle hazırlanmış olan bir programda veriler, DATA yazılımı ile saklanır. Yeni bir tanımlama yapılmasının zorunlu olduğu durumlarda otomatik öğrenme aktif hale geçerek yeni nesneyle ilgili enformasyonu içeren DATA yazılımı oluşturulur.

Pascal ve diğer dillerdeki işaretleyici (pointer) imkanlarıyla yeni bilgi ağaçlara eklenir. OPS5 gibi uzman sistem kabukları kullanılarak yeni kurallar otomatik olarak oluşturulur. Buradan da bir programın en önemli özelliklerinden birinin **otomatik bilgi edinme** olduğu anlaşılmaktadır.



Şekil 3.17. Tezgah seçimi yapan karar ağacı

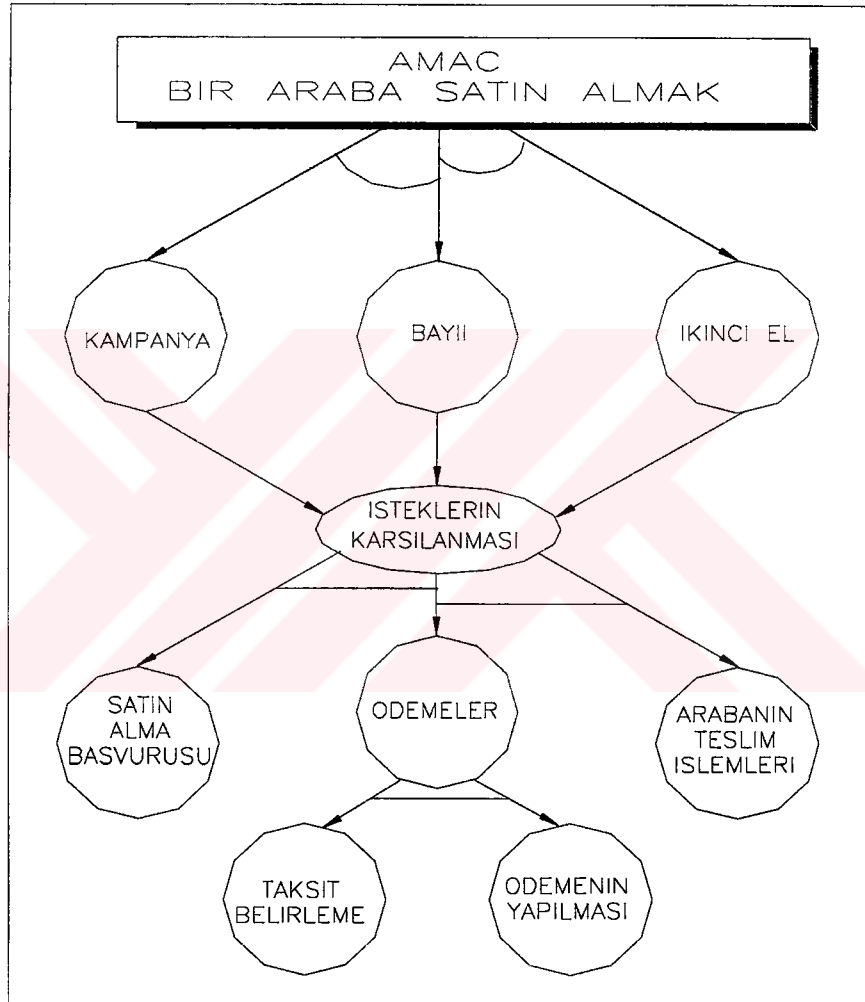
### 3.3.2. VE-VEYA (AND-OR) ağaçları ve amaçlar

Birçok uzman sistemde problem çözümünde, problemin çözümünü alt gruplara bölerek yapmaya imkan veren ve geriye zincirleme özelliğine sahip Prolog veya benzeri diller kullanılır. Ana amacı sağlamak için, bu methodda birçok alt amacın gerçekleşmesi gerekir. Geriye zincirleme problem tanımlamasında kullanılan ağaç veya kafeslerden biri de VE-VEYA ağacıdır. Şekil 3.18'de bir araba alım-satımı amacını sağlayacak basit bir VE-VEYA kafesi gösterilmiştir.

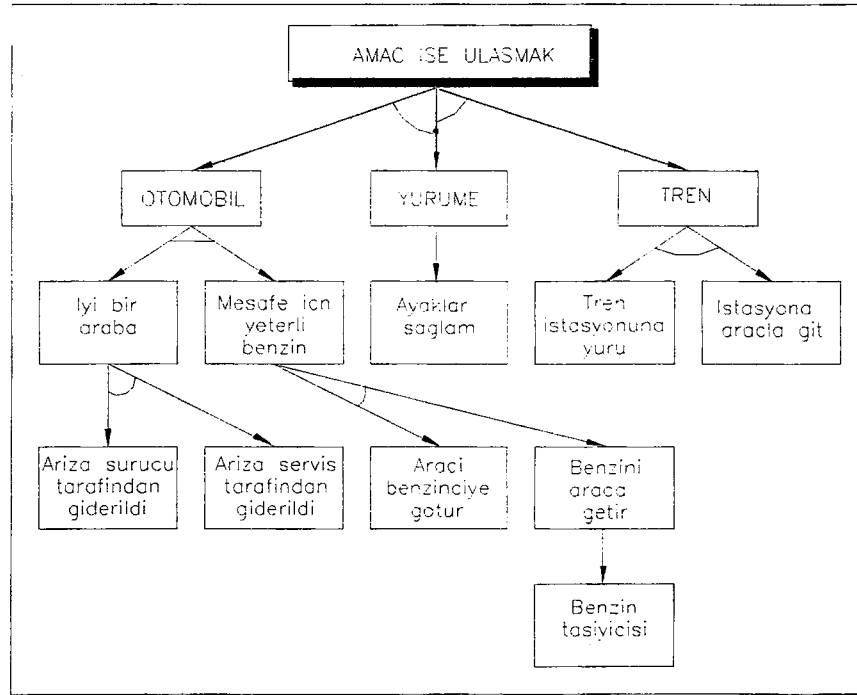
Bu amacı gerçekleştirmek için taksitlerin belirlenip ödemelerin yapılması gerekir. Araba bedelinin ödenebilmesi için bir kampanyaya müracaat edilebilir, bayi ile görüşülebilir veya ikinci el araba için de bir kişi ile bağlantı kurulabilir. Beklentilerin karşılanması amacıyla başvuru yapılır, ödeme üzerine karar verilir ve aracın teslim işlemleri hazırlanır. İsteklerin karşılanması amacı için üç alt amacın sağlanması gerekmektedir. Bu üç alt amaç da bir ark yardımıyla VE düğümüyle sağlanmıştır.

Bu diyagram, isteklerin karşılanması alt amacının bir üst seviyesindeki üç düğümden meydana gelen bir kafestir. Kampanya, bayi ve kişiden meydana gelen bu üç düğüm VEYA bağları ile birbirlerine bağlanmıştır.

Bir başka örnek de Şekil 3.19'da gösterilen şahsın evden işe gitme aracını belirleyen kafes yapısıdır [83].



Şekil 3.18. Bir aracın nasıl alınabileceğini gösteren VE-VEYA yapısı



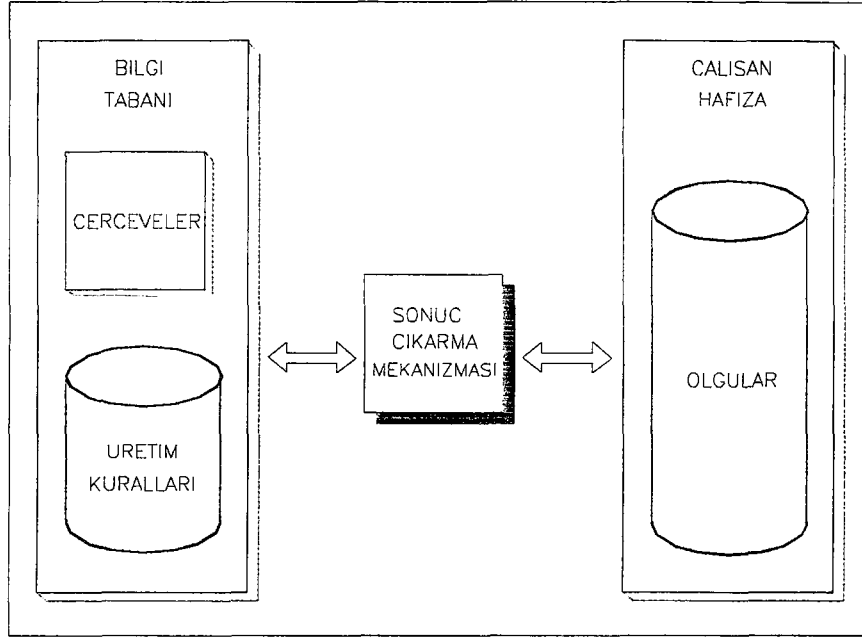
Şekil 3.19. Evden işe gitme alternatiflerini gösteren VE-VEYA yapısı

### 3.4. Sonuç Çıkarma Teknikleri

Bir uzman sistemde sonuç çıkarma teknikleri kontrol stratejisi ile sağlanır. Sonuç çıkarma teknikleri, bilgi tabanındaki bilgi ile çalışan hafızadaki problem olgularını birleştirerek (etkileştirerek) sistemi yönlendirir. Kontrol stratejisi sistem için amaçları hazırlar ve sorgulamanın kılavuzluğunu yapar.

Kullanıcı ile uzman sistemin iletişimi kullanıcı arayüzü yardımıyla sağlanır. Bir uzman sistemin en önemli elemanı olan **sonuç çıkarma mekanizması** ile aktif olan kuralların uyumu sağlanır. Sonuç çıkarmada genel olarak kullanılan **geriye** veya **ileriye zincirleme** olmak üzere iki sonuç çıkarma algoritması vardır.

Sonuç çıkarma işlemi, bilinenlerden gerekli yeni bilgilerin elde edilmesi için kullanılır. Bir uzman sistem, bu işlemi sonuç çıkarma mekanizması ile gerçekleştirir. Şekil 3.20'de bir uzman sistemde sonuç çıkarma işleminin uygulanışı gösterilmiştir [95].



Şekil 3.20. Uzman sistemlerde sonuç çıkarma [95]

Uzman sistemde sonuç çıkarma işlemlerinin nasıl çalıştığı veya oluşturulduğu basit olmasına rağmen, sıklıkla karşılaşılan sorulardandır. Örneğin aşağıdaki soruları sonuç çıkarma mekanizması nasıl bilmektedir?

- Kullanıcıya hangi soru sorulacak?
- Bilgi tabanı nasıl taranacak?
- Çalıştırılacak kural, bir çok kural içinden nasıl seçilecek?
- Sonuç bilgisi taramayı nasıl etkileyecek?

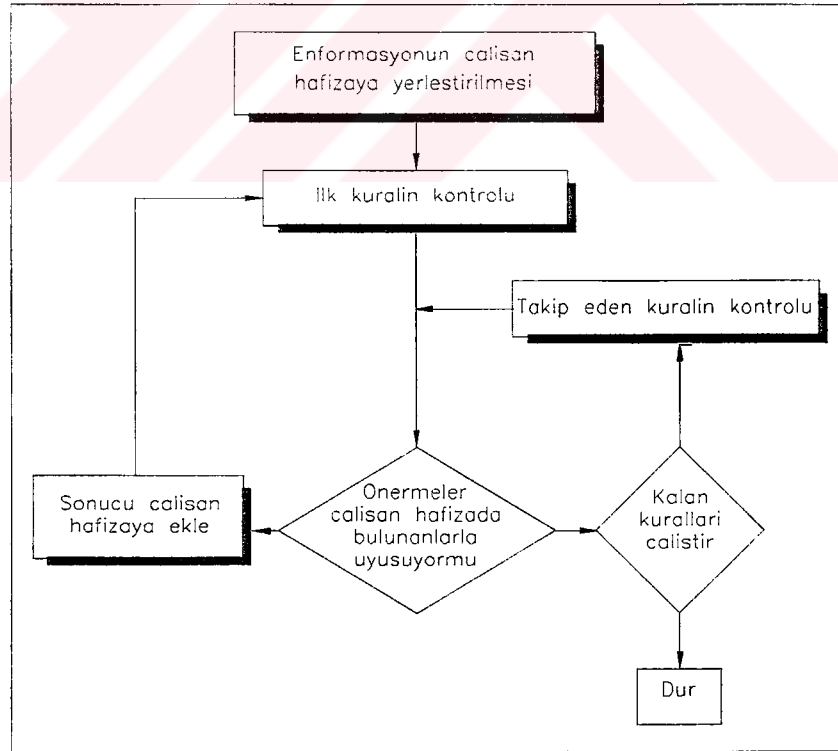
Bu soruların açıklanması ve anlaşılması, ileriye veya geriye zincirleme tekniklerinin iyi kavranmasından sonra açıklık kazanacaktır.

### 3.4.1. İleriye zincirleme

İleriye zincirlemede sonuç çıkarma mekanizması olguları inceler ve bir amaca ulaşmak için çalışır. İleriye zincirleme, kuralın **koşul** kısmıyla başlar ve kuralın **sonuç** kısmını sağlamak için koşulları yerine getirmeye çalışır [7]. Başka bir ifadeyle, ileriye zincirleme sebepten sonuca yapılan bir sorgulamadır. Evden ayrılmadan önce yağmur yağdığı görülürse (olgu) şemsiye alınmalıdır (sonuç)

örneğinde olduğu gibi. Bazı problemler için çözüm işlemi o konudaki enformasyonun toplanması ile başlar. İkinci adım olarak bu bilgi mantıksal sonuçların çıkarılması ile sebeplendirilir. Örneğin doktorun bir hastasını tedavi etmesi, hastalık belirtilerinin doktor tarafından hastadan öğrenilmesi ile başlar. Boğazın ağrması, yüksek ateş veya öksürme verilen ilk cevaplar olabilir. Doktor bu bilgileri teşhise yardımcı olarak sonraki değerlendirmelerinde kullanır. Bu sebepleme çeşidi uzman sistemlerde veri-sürücülü tarama kullanılarak modellenir. Bu işlem ileriye zincirleme olarak isimlendirilir.

Sistem öncelikle problem bilgisini kullanıcıdan alır ve çalışan hafızaya yerleştirir. Sonuç çıkarma mekanizması daha önceden tanımlanmış kuralları tarayıp çalışan hafızadaki bulunanlar ile uyuşanı arar. Eğer bir kural bulursa, bu kuralın sonucunu çalışan hafızaya yerleştirir. Bu işlem **kuralın işletilmesi** olarak isimlendirilir. Çevrim tekrar başlar ve kurallar başka bir uyuşmanın oluşmasına kadar kontrol edilir. Bu yeni çevrimde bir önceki işletilmiş olan kurallar göz ardı edilir. Bu işlem uyuşmanın son bulacağı ana kadar devam eder. Bu adımda çalışan hafıza, kullanıcı tarafından verilmiş ve sistem tarafından oluşturulmuş enformasyondan meydana gelir. İleriye zincirleme tekniği Şekil 3.21'de gösterilmiştir [95].



Şekil 3.21. İleriye zincirlemede sonuç çıkarma

### 3.4.2. Geriye zincirleme

Sonuç çıkarma mekanizması kuralın **sonuç** kısmı ile başlar ve bu bölümü ispatlanacak ana amaç olarak ele alır. Sonuç çıkarma mekanizması daha sonra, bir sonucu olan ana amaca sahip her hangi bir kural veya kural listesini araştırır. Uygun olan kural bulunduktan sonra, sonuç çıkarma mekanizması kuralın koşullarının her bölümünü sağlamak için çaba harcar. Başka bir ifadeyle geriye zincirleme, doğrulanması gereken bir hipotezden, hipotezi destekleyen koşullara geriye dönüş şeklinde yapılan bir sorgulamadır. Örneğin dışarıya çıkılmadan önce, ıslak şemsiye ve ayakkabılarla biri içeriye girerse, hipotez yağmurun yağdığıdır. Bu hipotezi desteklemek için, içeriye giren kişiye yağmur yağıp yağmadığı sorulur; cevap **evet** ise, hipotez doğru olarak onaylatılmış ve bir olgu gerçekleşmiştir.

Tasarım özelliğine bağlı olarak sonuç çıkarma mekanizması ileriye veya geriye zincirleme yapabilir. OPS5 ve CLIPS ileriye zincirleme için tasarlanmışken, MYCIN geriye zincirleme olarak hazırlanmıştır. Bunun yanında ART ve KEE her iki metodu kullanmaktadır. Teşhis problemlerinin çözümü için geriye zincirleme tavsiye edilirken tahmin, ekran tasarımı, görüntüleme ve kontrol için de ileriye zincirleme kuralı daha iyi sonuçlar vermektedir. Netice itibariyle, problemin cinsine göre sonuç çıkarma mekanizması kullanılmalıdır.

Önce var olan bilgiler yardımıyla problemle ilgili çalışmalar yapılır, sonra da mantıksal kararlara varılması düşünülen çalışmalarda iyi bir sonuç çıkarmak için ileriye zincirleme tekniği uygulanır. Bununla birlikte, çözüme bir hipotez ile başlayıp bunu destekleyen bilgilerin bir araya getirilmesi ile doğrulatmak gerekirse, geriye zincirleme daha uygun sonuç çıkarma metodu olmaktadır.

Metod, ispatlanması gereken **amaç** ile başlar. Bir başka bilgi tabanı tarafından doğrulanıp doğrulanmadığının kontrol edilebilmesi açısından, çalışan hafızada bu amacın olup olmadığı araştırılır. Eğer amaç daha önceden doğrulanmamış ise; sistem, kuralların sonuç kısımlarında amacın bulunduğu ve **amaç kuralı** olarak isimlendirilen kuralların taramasını yapar ve kural önermelerinin çalışan hafızada olup olmadığını kontrol eder. Önermelerin listede olmaması durumunda, başka kuralların desteklediği yeni amaçlar (alt amaçlar) ortaya çıkar. Bu işleme sistem her hangi bir kural tarafından desteklenmeyen önermeyi buluncaya kadar devam edilir. Böyle bir önerme bulunduğu kullanıcıya bunun hakkındaki enformasyon sorulur. Sistem bu enformasyonu kullanarak alt amaçların ve başlangıçtaki hedef amacın doğrulanması sağlar.



### 3.5. Temel Araştırma Teknikleri

YZ konusunda çalışan uzmanlar, problem bilgilerini **problem uzayında** grafik olarak tanımlamayı severler. Problem uzayı, problemleri tanımlayan düğümlerin meydana getirdiği ağaç veya grafikler olabilir. Düğümler, problem bütünlüğünü ve dallarını tanımlar. Bu tip problem tanımlamasının kullanımı ile çözüm düğümünün araştırılmasında dallar ve düğümler arasında bağlantı kurularak problem çözümü gerçekleştirilebilir.

Uzman sistemlerin oluşturulması noktasından bakıldığında, düğümlerin önermelerin ve kuralların sonuçlarının tanımlanmasında, dalların ise düğümler arasındaki ilişkilerin belirlenmesinde kullanıldığı görülür. Genel olarak grafik, problem uzayının oluşturulmasında kullanılan **çıkarım ağı** (network) yaklaşımıdır.

Problem uzayı yardımıyla bir problemin çözümünün gösterilebilmesi için Ankara'dan İstanbul'a gitmek amacıyla karayolu veya uçak yolculuğundan hangisinin tercih edileceğinin tesbit edilmesi örneğini ele alalım. Kararın verilmesinde aşağıdaki kuralların kullanıldığını ve problem uzayının da Şekil 3.22'de gösterildiği gibi kurulduğunu var sayalım.

Kural 1:

Eğer

Uçakla gitmeyi tercih edersen — Y1

O halde

THY'den bilet alabilirsin — A1

Kural 2:

Eğer

Karayolunu tercih edersen — Y2

O halde

Varan'dan bilet alabilirsin — A2

Kural 3:

Eğer

Bulunduğun şehirde Varan firması varsa — Y3

VE Varan'dan bilet alma imkanın varsa — Y4

O halde

Otobüs bileti al — A3

Kural 4:

Eğer

Bulduğun şehirde THY acentası varsa — Y5

VE Uçak bileti alma imkanın varsa — Y6

O halde

Uçak biletini al — A4

Kural 5:

Eğer

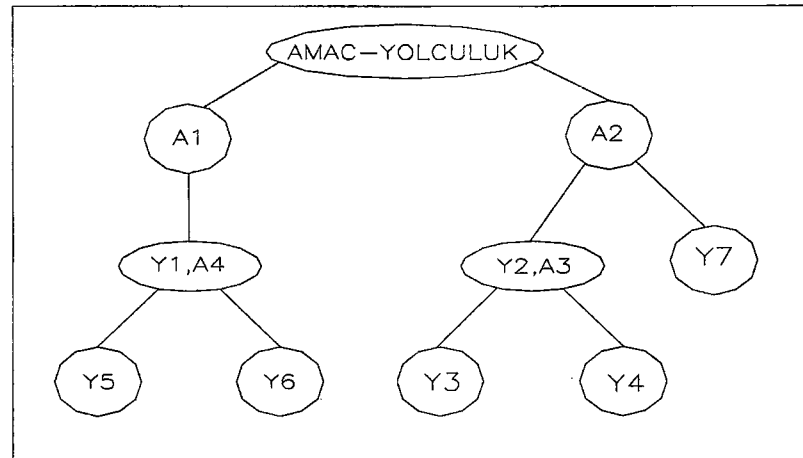
Karayolu ile yolculuktan hoşlanıyorsan — Y7

O halde

Varandan bilet alabilirsin — A2

Herhangi bir uzman sistemin uygulanmasında problem, problem uzayında araştırmanın yapılması ve çözümün elde edilmesidir. Geriye zincirleme tekniğinde sistem, muhtemel çözümlerden, Şekil 3.22'de olduğu gibi, A1 veya A2 düğümlerinin birisinden araştırmaya başlayarak dallar yardımıyla problemi destekleyen düğümlere doğru incelemeye devam eder.

Sonuç çıkarma mekanizması, ileriye zincirleme işleminde, ağ üzerinde bulunan Y3, Y4, Y5, Y6 veya Y7 düğümlerinden herhangi birinden başlayabilir. Başlangıç düğümünden yukarıya doğru bir amaç düğümünün yerleştirilmesine kadar araştırma yapabilir. Araştırma işlemlerinde genel olarak derinlik ilk araştırması (Depth-first search) ve genişlik ilk araştırması (Breadth-first search) olmak üzere iki yöntem kullanılmaktadır.



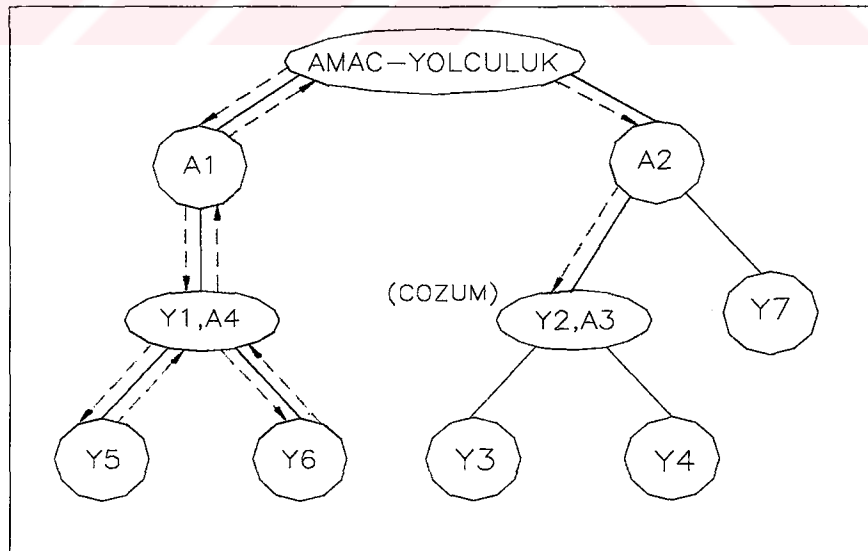
Şekil 3.22. Yolculuk seçiminde problem uzayı

### 3.5.1. Derinlik ilk araştırma tekniği

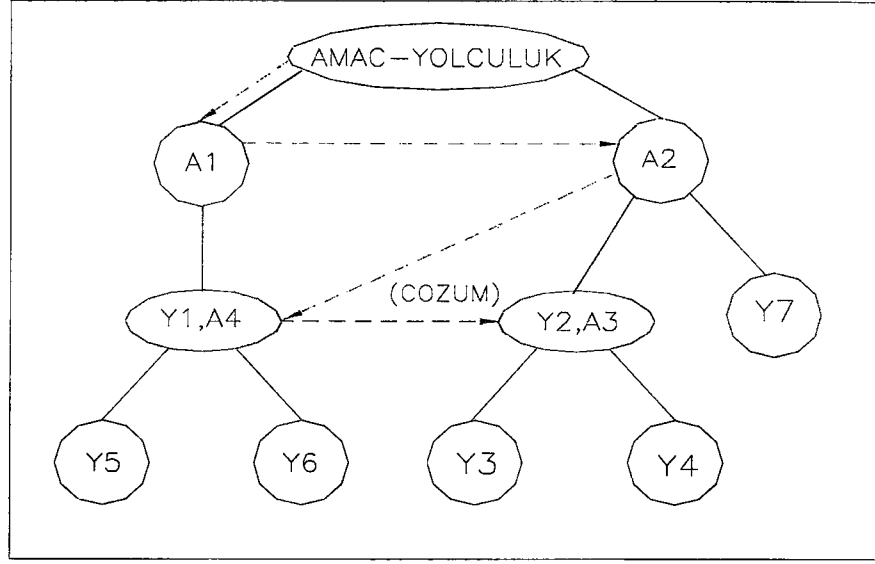
Derinlik ilk araştırmasının anlamı, bir sonuç bulmak için başka bir yol denenmeden önce mümkün olan her yolun son düğümüne kadar araştırılmasıdır. Araştırma, bir sonuca götürmeyen son düğümüne ulaşıldıktan sonra, aynı yoldan geriye döner ve bir çözüme ulaşıncaya kadar diğer düğümleri dener [96]. Bu yöntemde bir düğüm noktasından başlanıp o dalın sonuna veya bir amacın tesbitine kadar araştırmaya devam edilir. Her bir düğümde araştırmanın yönünü tayin etmek için, örneğin "soldaki daldan devam et" şeklinde bir kural belirlenir. Dalın sonuna gelindiğinde, araştırma bir üst düzeye geri döndürülür ve soldaki daldan araştırmaya devam edilir. Bu işlem bazı kaynaklarda **geriye iz sürme** (backtracking) metodu olarak da ifade edilir. Şekil 3.23'te bir önceki örnekteki yolculuk çeşidini tesbit etme problem uzayında araştırma metodunun geriye zincirleme-derinlemesine uygulanışı gösterilmiştir.

### 3.5.2. Genişlik ilk araştırma tekniği

Bu metotta aynı seviyede olan her düğüm, takip eden derinlik seviyesine inilmeden önce kontrol edilmelidir. Burada Şekil 3.24'te verilen yolculuk probleminde bir çözüm düğümü olan A3' ü bulmak için sırasıyla A1, A2, A4, ve A3 alt amaçları izlenmelidir.



Şekil 3.23. Geriye iz sürme ve derinlik ilk araştırma stratejisi



Şekil 3.24. Genişlik ilk araştırma stratejisi

#### 4. GELİŞTİRİLEN SİSTEMİN MODELLENMESİ

Bir BDİP sistemi bilgisayar ortamında geliştirilirken işlem planlama bilgisini, tecrübe ve sorgulama mantığını en iyi şekilde ortaya koyabilmek için, ilk önce bir metod seçilmelidir. Geleneksel algoritmik programlama tekniklerinin bu görevleri yerine getirmede çok fazla uygun olmadığı düşünülmektedir. Çünkü işlem planlama problemleri, genellikle insan kararı, tecrübe ve ampirik kurallara dayanarak sezgisel yöntemle çözülür [34].

İkinci olarak, endüstride imalat kuralları ve imalat işlemleri zamanla değişirken, sistem istendiğinde gerekli değişiklikleri yerine getirmeye müsaade etmek için, yeterli esnekliğe sahip olmalıdır. Bu tür problemleri çözmeye en iyi metod, US yaklaşımı kullanmaktır. Bu yaklaşım, işlem planlama bilgisi ve mantığını efektif bir şekilde kodlama yanında, işlem planlama için tüm programı değiştirmeksizin bilginin değiştirilmesine imkan vermesi gibi birçok avantaj sağlar.

Üçüncü olarak bağlama işleminin, işlem planlama süresince karar verme prosedürüne etkisinden dolayı işlem planlama ve bağlama işlemi birlikte değerlendirilmelidir.

Bu çalışmada esnek, tam otomatik, uzman sistem yaklaşımı kullanan, üretken BDİP sistemi geliştirilmiş ve BDT sistemlerle iletişim kurmak üzere STEP standardı hazırlanmıştır. STEP formatının kullanılması, işlem planlamayı gerçekleştirecek olan sistemi, herhangi bir BDT sisteme bağımlı olmaktan kurtarmaktadır. Ayrıca veri tabanını hazırlamada geometrik verilerin yanında, teknolojik verilerin de her hangi bir şekilde kullanıcı müdahalesi olmadan sisteme otomatik girişi sağlanmıştır. Programın bilgi tabanı Türkçe yapıda tasarlanmıştır. Bilgi tabanı, KURALLAR ve OLGULARDAN meydana gelir. Kurallar, VE/VEYA grafik tekniği kullanılarak yapılandırılmıştır.

Geliştirilen sistemde, her operasyondan sonra bağlama işlemi hakkında güvenilir kararlar vermek için, veri tabanında araparça modeller oluşturulmaktadır. Araparçalar, bu çalışma için tasarlanmış olan Komşu Yüzey Grafiği (KYG) adlı özel bir formatta gösterilmiştir (Bölüm 6). 3-B sınır temsili kullanan BDT verisi STEP formatında hazırlandıktan sonra, bu format, geliştirilen sistemin tanıyabilmesi için Prolog formata dönüştürülmüştür.

Bilgi tabanından kuralları ve olguları okuyabilme yeteneğine sahip olan sonuç çıkarma mekanizması, sonuçlara ulaşmak için geriye zincirleme ve derinlik

ilk araştırması yapabilir. Sistem, işlem plan kartı üretilecek parça için, önce stok ölçüsünü seçer ve parça için bir taslak tasarlar. Yüzeyler arasındaki açılar ve yüzeylerin komşuluk ilişkileri bulunduktan sonra taslak parça modeli KYG'de gösterilir.

Ayrıca referans yüzeyi, her özellik için operasyon, operasyonun her kademesi için takım tezgahları, bağlama yüzeyleri ve kesicilerin seçimi ve işleme parametrelerinin hesaplanması, otomatik olarak gerçekleştirilir.

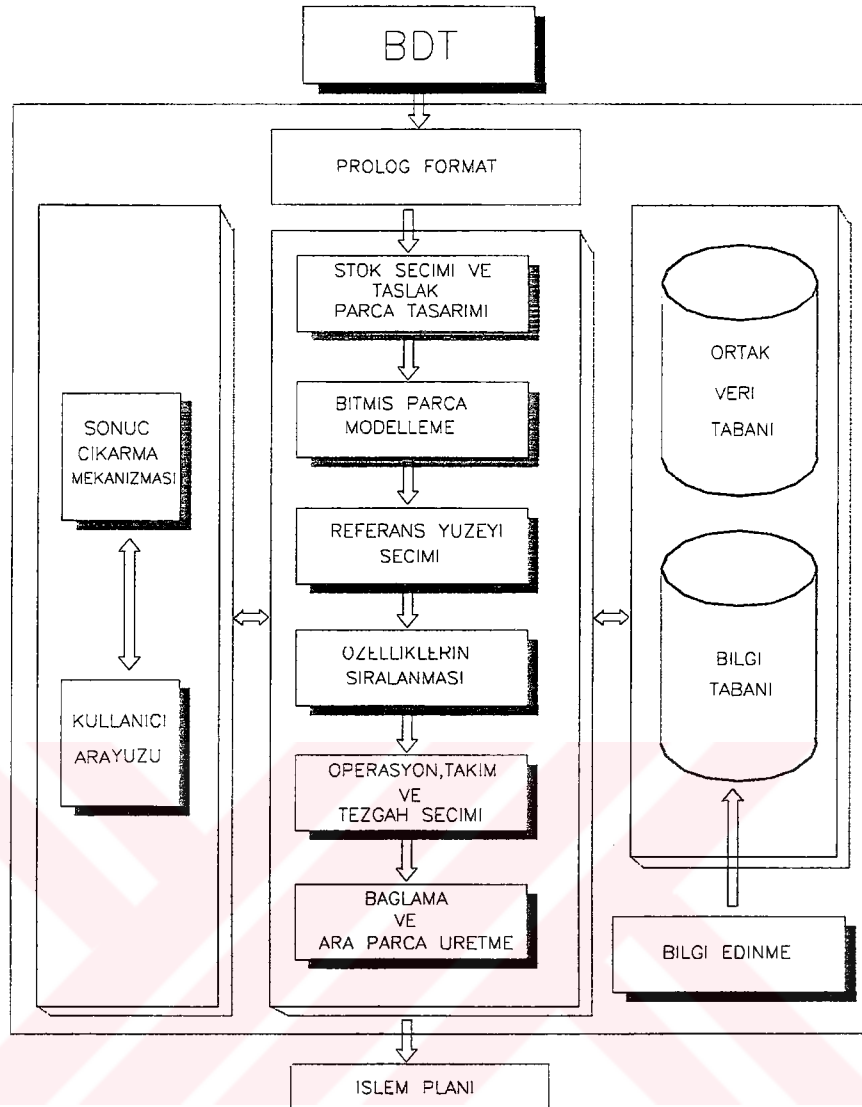
KYG şeması, sadece yüzeyleri ve özellikleri temsil etmez. Ayrıca bağlama işlemi ve operasyon seçimi için, bilgi gerektiren kompakt bir formda yüzeyler ve özellikler arasındaki açılar, içbükeylik-dışbükeylik ilişkisi gibi ilgili bilgileri de içerir. Böylece KYG, işlem planlama ve bağlama faaliyetleri için kullanılabilir bir yapıyı oluşturur.

#### **4.1. STEP Standardı ve US Yaklaşımı Kullanan İşlem Planlama Sistemi**

Üzerinde çalışılan araştırma, temel prizmatik parçaların işlem planlarının üretimini ele almaktadır. Sistem, esas itibarıyla operasyon seçimi, operasyonların sıralanması, kesiciler ve takım tezgahlarının seçimi, işleme parametrelerinin hesaplanması, bağlama yüzeylerinin belirlenmesi ve dinamik araparça modelleri oluşturma faaliyetlerini yerine getirmektedir. Sistem aşağıdaki modüllerden meydana gelir (Şekil 4.1).

- Kullanıcı arayüzü ve sonuç çıkarma mekanizması,
- Stok seçimi,
- Taslak parça tasarımı ve taslak parça oluşturma,
- Bitmiş parça modelleme,
- Referans yüzeyi seçimi,
- Operasyon sıralama ve operasyon seçimi,
- Kesici ve takım tezgahı seçimi,
- Bağlama yüzeylerini belirleme,
- Araparça modelleri oluşturma.

**Kullanıcı arayüzü ve sonuç çıkarma mekanizması:** Kullanıcı arayüzü, sistem ve öteki program modülleri arasındaki iletişimi sağlar. **Sonuç Çıkarma Mekanizması,** bilgi tabanından kuralları ve olguları okuyup uzman bir işlem planlayıcının yaptığı gibi, sezgisel bilgi kullanarak sonuçları çıkarır.



Şekil 4.1. Geliştirilen sistemin yapısı

**STEP dosyası:** Burada parçaların özellik tabanlı BDT sistemde tasarlandığı ve 3-B sınır temsili kullanan BDT dosyasının STEP formatında oluşturulduğu kabul edilir. STEP'te bir parça, yüzey, kenar, köşe kümesi ve özellik verisi, malzeme ve benzeri ilgili imalat bilgisi ile tasvir edilir. STEP dosyası sistemin tanıyabilmesi için Prolog cümlecikleri formuna dönüştürülmüştür.

**Stok seçme ve taslak parça tasarlama modülü:** Taslak parça bilgisi, işlem planlama sistemi için öncelikli bir giriştir. Bu yüzden taslak parça, talaş kaldırma işlemi başlamadan önce tasarlanmalıdır. Bitmiş parça modeli esas alınarak taslak parça tasarlanır ve stok boyutları seçilir. Taslak parça boyutları,



bitmiş parçanın en büyük **X, Y, Z** koordinat değerleri dikkate alınarak belirlenir. Bu nedenle taslak parça boyutları, bitmiş parça modelini içine alacak şekilde tayin edilir. Taslak parça için stok, veri tabanındaki stok ölçü dosyasını araştırarak seçilir. Taslak parça KYG' de açıklandıktan sonra veri tabanına yerleştirilir.

**Referans yüzeyi seçimi:** Bir parçanın referans yüzeyi, herhangi bir özellik işlenirken önemli bir rol oynar. Genel olarak işleme ve bağlama yüzeylerinin belirlenmesi, referans yüzeyine göre yapılır. Bu yüzden talaş kaldırma ve bağlama işleminden önce bir referans yüzeyinin seçilmesi gerekir. Referans yüzeyi seçimi için kurallar, bilgi tabanında Türkçe yapıda üretim kuralları halindedir. Sonuç çıkarma mekanizması her yüzeyi kontrol ederek bunlarla ilgili kuralları uygular ve yüzeylerden birinin referans yüzeyi olmasına karar verir.

**Operasyon sıralama ve operasyon seçimi:** Operasyonların sıralanması, özelliklerin sıralamasına dayanır. Bilgi tabanındaki kurallara göre, her özellik, başka bir özellikten önce işlenmelidir veya bazen bu özellik diğer özelliklerle birlikte işlenebilir. Örneğin genel olarak bir düzlem yüzey, onun üzerinde bulunan bir kanaldan önce işlenir. Kurallara göre bilgi tabanındaki özellik sıralaması, her özellik bir işleme önceliğine sahip olacak şekildedir. Sistem tarafından, en büyük işleme önceliğine sahip özellik başta olacak şekilde, özellikler, sırasıyla bir listeye yerleştirilir. Sonuç çıkarma mekanizması, özellikleri sıralamak için ilgili kuralları uygulayarak hangi özellik/yüzey'in ilk olarak işleneceğine ve bunu hangisinin takip edeceğine karar verir.

Hangi özelliğin ilk olarak işleneceğine karar verildikten sonra, bu özelliğe uygun bir operasyonun belirlenmesi gerekir. İmalat operasyonu seçimi için, örnek parçalara uygulanabilen gerekli kurallar hazırlanıp programa yerleştirilmiştir. Sonuç çıkarma mekanizması ilgili kuralları eşleştirerek uygular ve gerekli operasyonlara karar verir. Özelliklerle ilgili kuralları uygulamadan önce, boyut ve tolerans gibi ilgili teknolojik bilgiler dikkate alınır. İlgili bilgi varsa, sonuç çıkarma mekanizması, özelliği işlemek üzere gereken çözümü bulabilir.

**Kesici ve takım tezgahı seçme modülü:** İşlenecek özelliğe uygun takım ve takım tezgahı, her operasyon için bilgi tabanındaki ilgili kurallara başvurularak seçilir. İşleme parametreleri (güç, hız, ilerleme, takım ömrü vs) bu modülde hesaplanır.

**Bağlama yüzeylerini belirleme:** Özellikler için operasyonlar tayin edildikten sonra, her operasyon için bağlama şekli belirlenir. Bir özellik /yüzey işlenirken bağlama işlemi için esas problem; destekleme, yerleştirme ve tesbit etme yüzeylerinin belirlenmesidir. Bağlama işlemi için hazırlanan kurallar çalıştırılıp işlenecek özelliğe uygun destekleme yüzeyi seçilir. Yerleştirme yüzeyleri için uygun kenarların bulunmasıyla ilk ve ikinci yerleştirme yüzeyleri belirlenir. Daha sonra tespit etme yüzeylerinin seçimi hakkında karar verilir.

**Araparça modelleri oluşturma:** Her işlemten sonra, işlenen bölge parçadan kaldırılır ve oluşan yeni aramodel ilerleyen operasyonlarda kullanılmak üzere KYG'de temsil edilir.

Bir parça için sistemin çıktısı, seçilen kesicilerin tablosu, her kademede operasyonları kontrol etmek için kullanılabilen araparça modeller, her operasyon için bağlama bilgisi ve işlem planından meydana gelir. Program Modülleri takip edilen bölümlerde detaylı olarak ele alınacaktır.



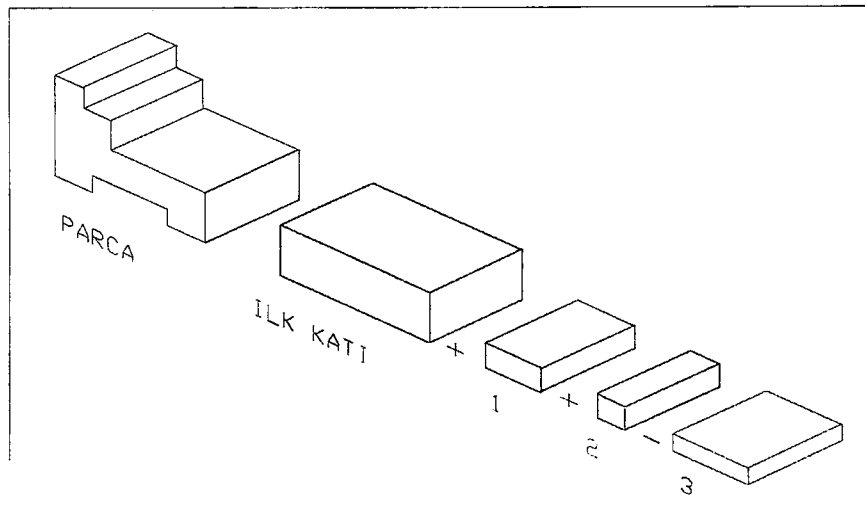
## 5. BİLGİSAYAR ORTAMINDA PARÇA MODELLEME

### 5.1. Üç Boyutlu Cisimleri Temsil Etme Teknikleri

Bilgisayar ortamında, üç boyutlu cisimleri (katı geometrinin) temsil etmede kullanılan çeşitli modelleme teknikleri vardır. Bununla beraber, genelde ticari amaçlı olarak iki çeşit modelleme sistemi kullanılır. Bunlardan birincisi Konstrüktif Katı Geometri (KKG), ikincisi ise Sınır Temsili (ST)'dir [97].

#### 5.1.1. Konstrüktif katı geometri (KKG)

KKG, katı cisimlerin hacimsel olarak temsilidir. Burada küp, küre gibi basit geometrik şekiller karmaşık katı cisimleri konstrükte etmek için kullanılır. Bu elementer geometrik şekiller, ilkel olarak adlandırılır. İlkeller, birleşim, kesişim ve ayrışım gibi Boolean veya mantıksal işlemlerle, bir katı modeli yapılandırmak için bir araya getirilir [6]. Örneğin Şekil 5.1'de bir parçanın konstrüksiyonu görülmektedir. Burada, 1 ve 2 numaralı prizmatik bloklar, bir birleştirme işlemiyle bazı noktalardan ilk bloğa eklenirken, 3 numaralı blok, bir ayrıştırma işlemiyle ilk bloktan çıkarılabilir. Her blok, hacim, yüzeyler, kenarlar veya köşeler gibi yüzey varlıklarını tanımlamadan belirlenebilir. Bloklara ait hacimler ve bunların uzaysal konumları, yüzey varlıkları yerine, parametrelerle belirlenir. Bu durumda KKG, genelde parametrik bir gösterim sistemi olarak isimlendirilir [98].



Şekil 5.1. Bir parçanın KKG ile temsil edilmesi

### 5.1.2. Sınır temsili (ST)

Katı cisimlerin sınırları, bir sınır temsili modelleyicide açık bir şekilde belirlenir. Burada sınırsız bir yüzey, yüzey, kenarlar ve köşeler olmak üzere üç elemanıyla temsil edilir. Bunlara ilave olarak bir cismi tam olarak temsil etmede, **kabuk** "kapalı birleştirilmiş yüzeyler takımı" ve **çevrim** "kapalı birleştirilmiş kenarlar takımı" kullanılır. Sonra yüzey elemanları, ayrı ayrı belirlenen geometri ve topoloji ile birleştirilir [8]. Topoloji, köşeler, kenarlar ve yüzeylerin nasıl ilişkilendirileceğini; geometri ise her varlığın konumunu ve boyutlarını açıklar [6]. Şekil 5.2'de gösterildiği gibi, ST metoduyla temsil edilen bir cisim, yüzey-kenar-köşe grafiği kullanılarak veri tabanına yerleştirilir. Yüzey-kenar-köşe grafiği, aşağıdaki Euler eşitliğini sağlamalıdır.

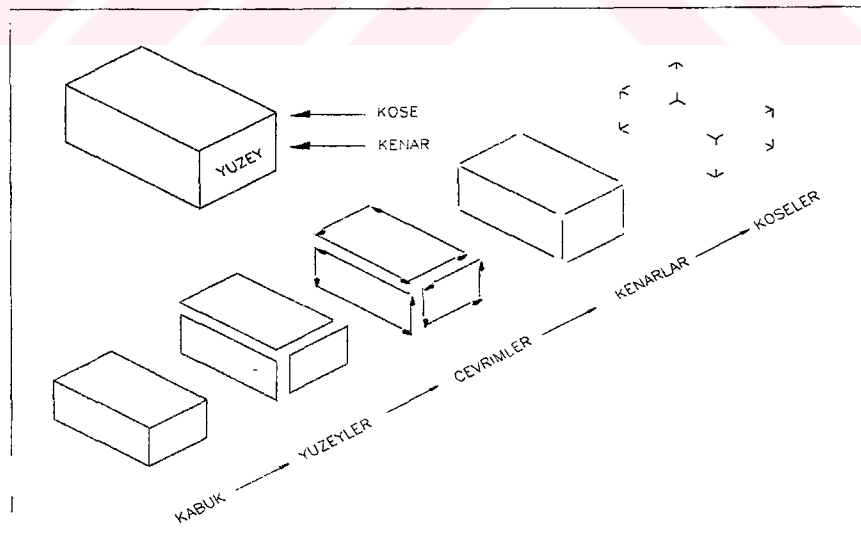
$$V+F-E=2 \quad (5.1)$$

Eğer parçada delikler ve geçişler varsa formül :

$$V+F-E-H-2(M-G)=0 \text{ olur.} \quad (5.2)$$

Burada:

V= Köşe sayısı, F= Yüzey sayısı, E=Kenar sayısı, H= Delik sayısı,  
M= Birleştirilmemiş parça sayısı, G= Geçiş sayısıdır.



Şekil 5.2. Bir parçanın ST ile temsil edilmesi

Bu çalışmada yüzeyleri, kenarları, köşeleri ve sınır elemanlarını açık bir şekilde temsil etmesinden dolayı, STEP'te BDT modelleri yapılandırılmada, ST tekniği kullanılmıştır.

## 5.2. Özellik Tanıma ve Özellik Tabanlı Tasarım

BDT sistemleri, üç boyutlu cisimleri temsil etmek için, bir hayli geliştirilmiştir. Bununla beraber parçaların BDT ile elde edilen veri yapıları, insan katkısı olmaksızın, BDT ile BDİP/BDÜ sistemlerini arayüz etme veya bütünleştirmeyi sağlamada doğrudan kullanılamaz.

Ticari olarak bulunan BDT sistemleri, bir parçanın özelliklerini tümüyle tanımlamak için yeterli değildir. BDİP ve imalat için temel bilgi sağlandıktan sonra, özellikler, üretimdeki önemine göre, bir parçanın bölgeleri olarak tanımlanabilir. Bu özellikler, genellikle mekanik bir parçanın bir grup yüzeyini içerir [99]. Tolerans ve yüzey bilgisi, genellikle BDT veri tabanından, istendiğinde kolayca elde edilebilecek bir yapıda değildir. Üç boyutlu BDT modellerde parça tasviri, basit geometrik ve temel seviyede topolojik bir formdadır [100]. Halbuki BDİP sistemleri, **şekil özellikleri** yanında, onlarla ilgili imalat bilgisine bağlı farklı bilgilere de ihtiyaç duyar.

Genel olarak işlem planlama sisteminin, tasarım ve imalat arasında, bir köprü olduğu kabul edilir. BDİP sistemleri için en önemli giriş bilgisi, BDT'den gelen parça bilgisidir. Henüz BDT ve BDİP sistemleri arasında doğrudan bir iletişim yoktur. Bu yüzden işlem planlamada, aradaki iletişimi sağlamak üzere, uygun bir format gerekir [14]. Bu tür iletişim problemlerini çözmek için **Özellik Tanıma ve Özellik Tabanlı Tasarım** adlı iki yaklaşım tavsiye edilir.

### 5.2.1. Özellik tanıma

Özellik tanıma, veri tabanında depolanan BDT geometrik modelini test ederek, bir geometrik modelde bulunan yüzeyler, kenarlar, köşeler ve ilkel katılar gibi basit varlıklardan (kanallar, delikler, cepler ve çıkıntılar gibi) şekil özelliklerini çıkarmaya çalışır [16]. Bu metod hem ST, hem de KKG modelleyicilere uygulanmaktadır [101]. Şekil özelliklerini tanımak için uygulanan metotlar aşağıdaki gibi sınıflandırılabilir:

- 1- Syntactic (dizimsel) model tanıma,
- 2- Ayrıştırma yaklaşımı,
- 3- Uzman sistem mantığı,

- 4- KKG yaklaşımı,
- 5- Grafik tabanlı yaklaşım.

Bugüne kadar çeşitli özellik tanıma sistemleri geliştirilmiştir. Özellik tanıma sistemleri tarafından kullanılan KKG ve ST katı modelleyiciler, teknolojik bilgi içermez. Bu yüzden bu bilgiler gerekliyse, sisteme manüel olarak ilave edilir veya sistemde bulundurulmaz. Yani özellik tanıma sistemleri, henüz tam olarak geliştirilememiştir. Uygulanan teknikleri, insan katkısı olmaksızın, yerine getirmek oldukça karmaşıktır ve karmaşık parçalar için uygulanması güçtür [102].

### 5.2.2. Özellik tabanlı tasarım

Özellik tanıma ihtiyacını ortadan kaldıran bir yaklaşım, **Özellik Tabanlı Tasarım** olarak kabul edilir [43]. Özellik Tabanlı Tasarım, ürün modeli, geometri, topoloji, şekil özelliği, boyutlar, toleranslar, yüzey pürüzlülüğü ve malzeme hakkındaki gerekli bilgilerin tümünü içerir [103]. Bu yaklaşımda özellikler ve tüm diğer imalat bilgileri, tasarım işlemi sırasında belirlenir ve böylece BDİP için gerekli tüm veriler, üretilecek parçanın veri tabanında bulundurulur. Veri tabanında bulundurulan bu bilgilere ihtiyaç duyulduğunda, buradan kolayca çıkarılabilir.

### 5.3. BDT Verileri Değişirme Standartları

Her BDT sistem, kendine has bir temsil tekniğine ve parça geometrik modeli için veri tabanı yapısına sahiptir. Bunun anlamı, bir BDT sistemi diğer bir BDT veri tabanından verileri okuyamaz. Farklı firmalar veya aynı firma içerisindeki değişik gruplar, kendilerinin farklı ihtiyaçlarına göre, değişik BDT sistemleri seçebilirler. Bu yüzden, bir sistemde tasarlanan sayısal parça modelini, diğer bir sistemde kullanabilmek için sayısal parça modellerini değiştirmek gerekir. BDT sistemleri arasındaki bu uyumsuzluk problemlerini ortadan kaldırmak için, IGES, VDA-FS ve SET gibi grafik verileri değiştirme standartları geliştirilmiştir. Bu standartlar, farklı BDT sistemleri arasındaki iletişimi sağlayan tabii bir formatı kullanır. Hala BDT sistemleri, giriş işlemcisi (preprocessor) ve çıkış işlemcisi (postprocessor) olmak üzere iki programa ihtiyaç duymaktadır. Bir BDT sistemde parça tasarlandıktan sonra, bu sistemin kendine has temsil formatını, başka bir sisteme parça bilgisi olarak göndermek için, giriş işlemcisi kullanılarak standart bir formata; benzer şekilde son işlemci de diğer BDT

sistemlerden gelen verileri, sistem formatındaki standart formata dönüştürür. Sonra bu parça BDT sistemde görüntülenebilir ve gerektiğinde düzenlenebilir.

**Başlangıç grafikleri değiştirme özelliği (IGES):** IGES, öncelikle BDT sistemleri arasında parça tanımlama verilerini transfer etmek üzere geliştirilmiştir. Bu yüzden BDT sistemleri tarafından yaygın olarak kullanılmaktadır. IGES, AEC (Mimari Mühendislik ve Konstrüksiyon) ve tesis tasarımını içeren geniş bir uygulama alanını kapsamaktadır. Burada ürün, geometrik olan ve olmayan bilgilere göre temsil edilir. Geometrik bilgiler, fiziksel şekilleri tanımlayan noktalar, eğriler, yüzeyler, katılar ve bunlar arasındaki ilişkileri içerir. Geometrik olmayan bilgiler ise, dip notlar, tanımlar ve organizasyonel verilerdir. Bu bilgi, metin, notasyon, ana hatlar, birleşme, gruplar ve benzerlerini içerir. IGES, Sonlu Eleman Modellemeye ek olarak Tel Kafes, KKG ve ST modellerini de destekler [104].

**VDA-FS standardı:** VDA-FS, Alman Otomobil İmalatçıları Birliği tarafından geliştirilmiştir. Bu standart, noktalar, parametrik polinom eğrileri ve ekleri kapsar. DIN tarafından bir standart olarak kabul edilmesi ile, basit bir arayüzü elde etmek üzere tasarlanmıştır. Bu standart, sadece yüzey transferinde gereken temel elemanları ele alır [11].

**SET:** SET (Standart d'Exchange et de Transfer), IGES'e göre daha fazla veriyi değiştirmek üzere geliştirilmiştir. SET, IGES'in sahip olduğu geometri, yüzey, yapı ve dipnotlar gibi özellikleri bulundurması yanında, geometri transfer etme kapasitesine de sahiptir.

**Ürün verileri tanımlama arayüzü (PDDI):** PDDI projesi, tasarım ve imalat arasındaki komple bilgisayar tabanlı ürün tanımlama arayüzünü belirlemek ve göstermek için, Amerikan Hava Kuvvetleri tarafından düzenlenmiştir. PDDI, ST katıları, toleranslar, parça özellikleri ve parça kontrol bilgisini içerir [105].

**Ürün Verileri Değiştirme Standardı (PDES):** (Ürün Model Verileri Değiştirme için STEP Standardı) PDES, orijinal olarak IGES organizasyonunca hazırlanmış olup halen ISO 10303'ün bir alt komitesi olan TC 184/SC4/ WG tarafından, Uluslararası Standartlar Organizasyonunda geliştirilmektedir. STEP projesini geliştirmekte olan komite, esas itibarıyla PDES ve STEP'in aynı standartlar olduğunu vurgulamıştır. PDES (STEP) standardını geliştirmek üzere



çalışan komite, bu standart tamamlandığında, IGES, SET, VDA-FS ve benzerlerinin yerini alarak bu alandaki tek dünya standardı haline geleceğini ileri sürmektedir. ESPRIT, CAD-1, IGES, VDA-FS ve SET gibi çeşitli standartlar, STEP standartının geliştirilmesine önemli katkılarda bulunmuştur [106].

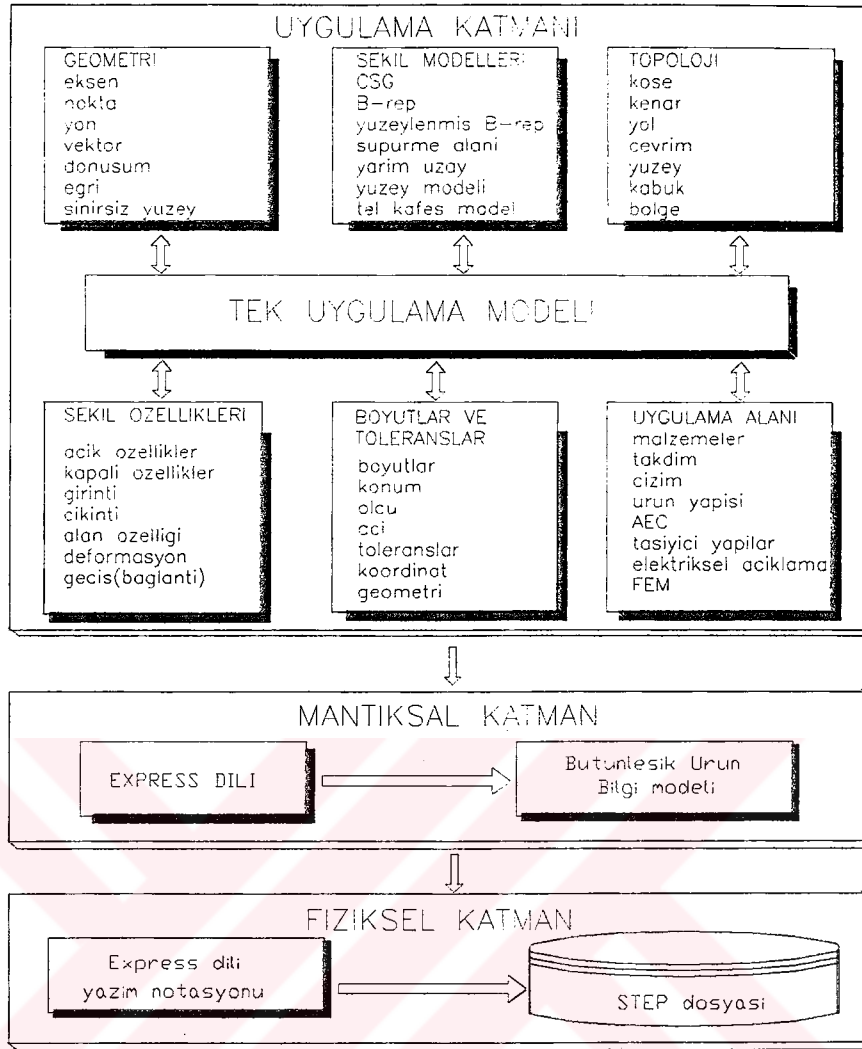
Tüm BDT verileri değiştirme standartları, esas itibariyle geometrik varlıkları kullanan BDT sistemler arasında (PDDI) hariç, iletişimi sağlama amacına yöneliktir. Örneğin IGES, her ne kadar BDT verileri değiştirme standardı olarak çok yaygın bir şekilde kullanılsa da, komple bir ürünü tanımlamak için gerekli olan özellikleri, toleransları, yüzey pürüzlülüğünü ve malzeme gibi bilgileri desteklemez. Esas itibariyle IGES, BDT ve BDİP/BDÜ sistemleri arasında verileri transfer etmek üzere geliştirilmiştir. Burada, ürün model verilerinin anlamını yorumlamak suretiyle bir sonuç elde etmek için, bir kişinin bulunması gerektiği kabul edilir. Örneğin bir parçada kaç tane delik olduğunu belirlemek için, bir kişiye ihtiyaç vardır. Çünkü bir delik, standartta tek başına bir varlık değildir. Bu nedenle, tüm ürün model verilerini desteklemek için, değiştirme formu ve daha ayrıntılı bir tasvir gereklidir. PDES projesi, tabii bir formatta yeterli derecede bilgiyi sağlayarak ürün modellerini değiştirmek için geliştirilmiştir. Böylece veriler, BDT sistemleri ve BDİP/BDÜ sistemleri tarafından doğrudan yorumlanabilir.

STEP, geometriye ek olarak, imalat, tolerans, malzeme ve yüzey bitirme özellikleri gibi geometrik olmayan bilgileri de destekler. STEP modeli, Tel Kafes, ST, KKG ve yüzey modellemeyi içerir. STEP projesinin amacı; imalat, üretken işlem planlama sistemleri, BDT yönetimli muayene, otomatik SD kodu oluşturma ve montaj işlemleri gibi bir çok alanda kullanılan programların gerektirdiği komple ürün modeli için ihtiyaçları desteklemektir.

STEP, mekanik parçalar, mekanik montajlar, gemi modelleri, AEC modelleri, FEM (Sonlu Eleman Modelleri) modelleri imalat ve çizim uygulamalarını içeren, geniş bir uygulama alanını kapsar. Ayrıca toleranslar, katı modelleme (KKG ve ST'de), eğri ve yüzey modelleme ve sunuş verileri de desteklenir [12].

#### 5.4. STEP Standardın Genel Hatları

STEP metodolojisi, resmi diller, referans modelleri kullanımı ve üç seviyeli bir mimariye dayanır. Seviyeler, **uygulama**, **fiziksel** ve **mantıksal katman**'dan meydana gelir (Şekil 5.3). Modeller ilgilenilen her özel alan için uygulama katmanında geliştirilir. Üretilen veri modelleri, **güncel modeller** olarak adlandırılır ve bunlar daha sonra, **Bütünleşik Ürün Bilgi Modeli**



Şekil 5.3. STEP standartının yapısı

(IPIM) veya **Mantıksal katman** olarak isimlendirilen komple bir bilgi modelinde toplanır. Uluslararası STEP standardı, temsil edilecek ürün verilerini tasvir etmek için resmi verileri belirleme dili olan **EXPRESS** dilini (IPIM)'i formüle etmek üzere kullanır (ISO 10303-11). Express dili, ürün verilerinin açıklanmasında bütümleştirme imkanlarını ve uygulama protokollarını sağlar. Bu dil kullanılarak hazırlanan yapı, fiziksel katmanda bulunan **fiziksel dosya** içerisine ayrıntılı bir şekilde yazılır. Fiziksel Dosya, **Wirth Syntax Notation (WSN)**'de yapılır. WSN, bir söz dizini dosyasıdır. Böyle bir dosya ile bir bilgisayarın problemi işlemesi kolaylaşır.

### 5.4.1. Uygulama katmanı

Şekil 5.3'te gösterildiği gibi uygulama katmanı, şekil modelleri, geometri, topoloji, şekil özellikleri, boyutlar ve toleransları içerir.

#### 5.4.1.1. Şekil modelleri

Ürünün uygulama alanı belirlendikten sonra, uygulama alanına özgü şekil modeli seçilir. Uygulama alanları, SEM, ürün yapısı (mekanik ürünler), sunuş, çizim, elektrik-elektronik uygulamaları ve MMK'dır. Bir şeklin modeli, ürün verilerinin herhangi bir geometrik elemanın, nominal şeklinin temsilidir. **Şekil Model Şeması**, Katı Model, Tel Kafes Modeli, Katı Örnek ve Geometrik Setten oluşur (Şekil 5.3).

**Katı model**, ST, Yüzeylenmiş ST, KKG ve Süpürülen Alan Katları gibi değişik tip katı model temsillerini destekler.

**Sınır temsili (ST)**, bağlanmış, yönlendirilmiş ve sonlu bir yüzeyde yerleştirilmiş kenarlar ve köşelerin grafiğini kullanır. Yerleştirilen grafik, sınırsız bir yüzeyi, alanları birbirine bağlanmış sınırlı yüzeylere böler. Burada kenarlar ve köşeler, yüzeylerin sınırlarını teşkil eder. Grafik içerisindeki her varlık tek bir varlıktır. Bir yüzeyin sınırlarını çevreleyen kenarların her biri, başka bir yüzey tarafından da paylaşılır. ST, bir veya daha fazla **kapalı kabuk** tarafından temsil edilir.

**Yüzeylenmiş sınır temsili**, yalnız sınırlı düzlem yüzeye sahip katı modelleri temsil etmek üzere, çeşitli sistemleri desteklemek için hazırlanmıştır. ST modelin aksine, kenarlar ve köşeler, modelde açık bir şekilde temsil edilmez. Fakat çevrimlerden yararlanılarak kapalı bir şekilde, sınır temsil modelleri elde edilebilir. Yüzeylenmiş sınır temsili, ST gibi aynı topolojik kısıtları karşılamak zorundadır.

**Konstrüktif katı geometri (KKG) modeli**, ilkeller olarak adlandırılan geometrik varlıklar, bir katıyı konstrükte etmek için kullanılır. Bu ilkeller, bir blok, kama, silindir, koni, küre ve süpürülen katları kapsar. Bir ilkel

için model bilgisi, ilkelin şeklini belirleyen boyutları, ilkelin lokal koordinat sistemini belirleyen noktayı ve yön koordinatlarını kapsar. KKG katı varlığı, bu elemanlar üzerinde yapılacak birleşim, ayrışım ve ara kesit gibi işlemlerle, varlığa ait elemanlara özgü bilgileri içerir.

**Süpürülen alan katı varlığı**, bir düzlem şekil üzerinde, süpürme hareketiyle belirlenen varlıkları içerir. Süpürülen katı uzayındaki konum, süpürülen yüzeyin konumuna bağlıdır ve bu da süpürülen alan katısının bir yüzeyidir.

**Yüzey modeli**, bazı mekanik parçaların temsilleri bir katının bütün sınır formunu gerektirmeyen yüzeylerden oluşur. Böyle bir model, çeşitli yüzeylerle temsil edilir. Yüzeyler, kenarlar ve köşeler dışında ara kesite sahip olmamalıdır.

**Tel kafes modeli**, mekanik bir parçanın sınırını şekillendiren yüzeylerin ara kesiti hakkındaki bilgiyi içerir; fakat yüzeyler hakkında bilgiyi kapsamaz. Tel kafes modeli, kenarlar ve köşelerin grafiğinden meydana gelir. Grafik içerisindeki kenarlar, kendi köşeleri dışında ara kesit oluşturmazlar.

**Geometrik Set**, topolojik bir yapı mevcut olmadığı zaman, modelleri transfer etmek için gereklidir. Geometrik set, geometrik iki boyutlu, geometrik üç boyutlu ve üç boyutlu yüzey setlerinden meydana gelir.

**Katı örnek**, yeni bir konumda başka bir katının kopyasıdır.

#### 5.4.1.2. Geometri

STEP standartta ortak bir şema, iki ve üç boyutlu geometrinin tanımlanması için kullanılmıştır. En temel geometrik varlık, **nokta** ve **yöndür**. Daha karmaşık geometrik varlıklar, noktalar ve yönlere göre, direk ve endirek olarak belirlenir. KKG ilkelleri, elementer yüzey tanımlarıyla uyumlu bir şekilde belirlenir. Tüm eğriler ve yüzeyler, konikler, küre, toroidal ve silindirik yüzeyleri kapsayan birleşik parametre sistemi ile belirlenir.

Yüzey varlıkları, basit ST katı modelleyicilerin ihtiyaçlarını destekler. Basit yüzeyler, düzlemsel, küresel, silindirik, konik, toroidal yüzeylerdir. Yüzeyler genellikle sınırsız olarak tanımlanır ve bunların açık, ya da kapalı olarak sınırlandırılacağı kabul edilir. Açık sınırlandırma, kesilmiş veya sınırlanmış yüzey

varlıkları ile tanımlanır. Geometrik varlıkların ana yapısı, Şekil 5.4'te gösterilmiştir. Genellikle kullanılan varlıklar aşağıda açıklanmıştır.

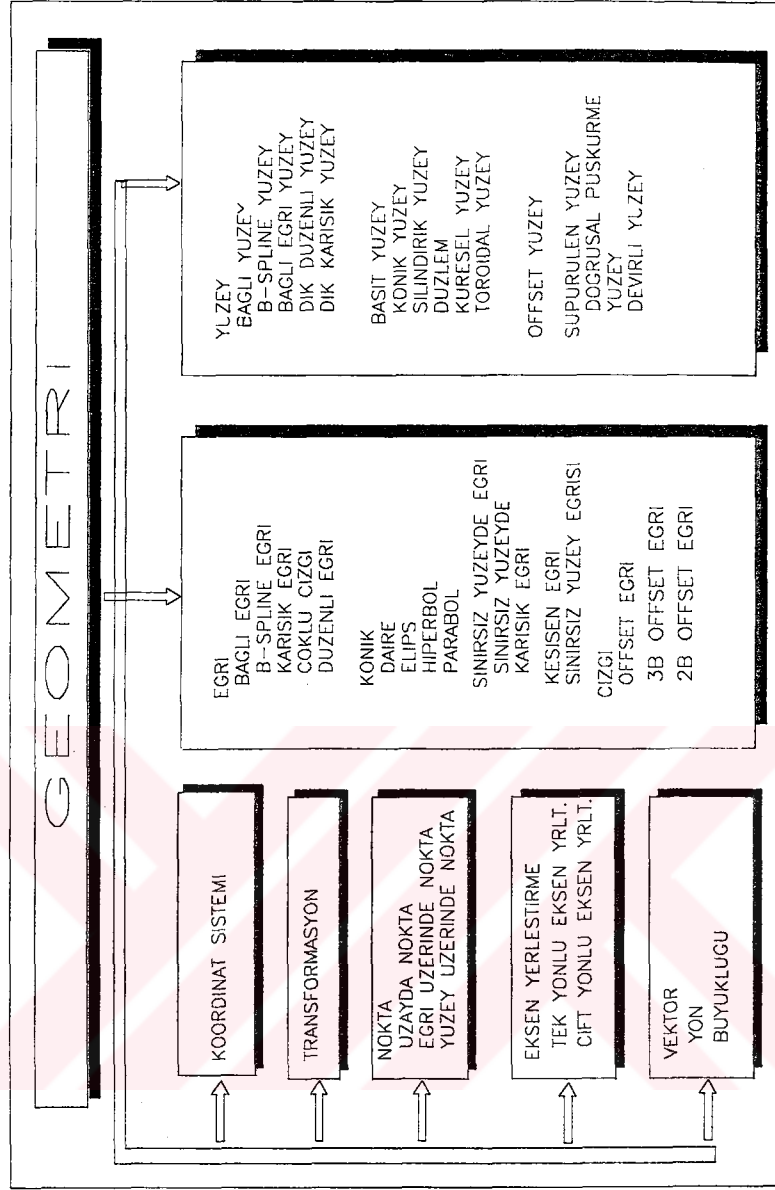
**Nokta**, en basit geometrik varlık olan nokta, herhangi bir koordinat uzayında bir yer olarak tanımlanır. Bir nokta, **kartezyan**, **eğri** veya **yüzey üzerindeki nokta** olmak üzere üç farklı şekilde tanımlanabilir. Kartezyen nokta, iki veya üç boyutlu Dik Kartezyan Koordinat Sisteminde temsil edilir. Eğri üzerindeki bir nokta, özel bir parametreye göre eğrinin değerlendirilmesi ile bulunan, eğriye ait bir noktadır. Yüzey üzerindeki bir nokta, bir yüzeyin parametrik uzayında yerleşir.

**Vektör**, **yön** veya **vektörün büyüklüğü** ile tanımlanır. Yön, iki veya üç boyutlu uzayda genel bir yön vektörünü tanımlar. Burada bileşenlerin gerçek büyüklüğü önemsizdir. Büyüklüğü olan vektör, bir vektörün yönünü ve şiddetini açıklar.

**Eksen Yerleştirme**, geometrik bir varlığın tanımlanması için bölgesel ortamı açıklayıp tanımlanacak varlığı yerleştirir ve bunun uyumunu verir.

**Eğri**, kendi koordinat uzayında hareket eden bir noktanın yolu olarak ifade edilir. Eğri, çizgi, koni, sınırlı eğri, yüzey üzerinde eğri ve offset eğri (boru dirseği eğri)'den oluşur. **Çizgi**, bir yön ve bir noktayla tanımlanan sınırlandırılmamış bir eğridir. Çizginin pozitif yönü, yön vektörü tarafındadır. **Konik eğri**, geometriden ziyade kendi parametrelerine göre tanımlanır. Konik eğri, daire, elips, hiperbol ve parabolü içerir. **Sınırlandırılmış eğri**, yay uzunluğu sınırlı olan eğridir. **B-Spline eğri** de diğer bir eğri çeşididir.

**Sınırsız yüzey**, uzayda hareket halinde olan ve sürekli bir şekilde değişen eğri olarak tanımlanır ve sınırsız, elementer, süpürülen, sınırlandırılmış ve offset yüzeyden oluşur. **Elementer sınırsız yüzey**, düzlem, silindirik, konik, küresel ve toroidal sınırsız yüzeyleri içerir. Sınırsız düzlem yüzey, sınırsız yüzey üzerindeki bir noktayla tanımlanır. Sınırsız silindirik yüzey, radyüsü, uyumu ve konumu ile belirlenir.



Şekil 5.4. STEP'te geometrinin mimarisi

### 5.4.1.3. Topoloji

Bir parçayı meydana getiren nesnelere arasındaki bağlantı ilişkileri, topoloji olarak ifade edilir. Topoloji köşe, kenar, yol, çevrim, yüzey, alt yüzey, kabuk, bağlanmış kenar ve yüzey kümesini içerir. Topoloji sınıf yapısı, Şekil 5.5'te gösterilmiştir. Topoloji esasen ST katı modelleme için tasvir edilir, fakat başka uygulamalarda da kullanılabilir. Burada bağlantıyı temsil etmek için açık bir metod gereklidir. Topolojik varlıkların çoğu, isteğe bağlı geometrik niteliklere sahiptir. Bu nitelikler, topolojik varlıkların uygun geometrik verileri ile birleştirilir.

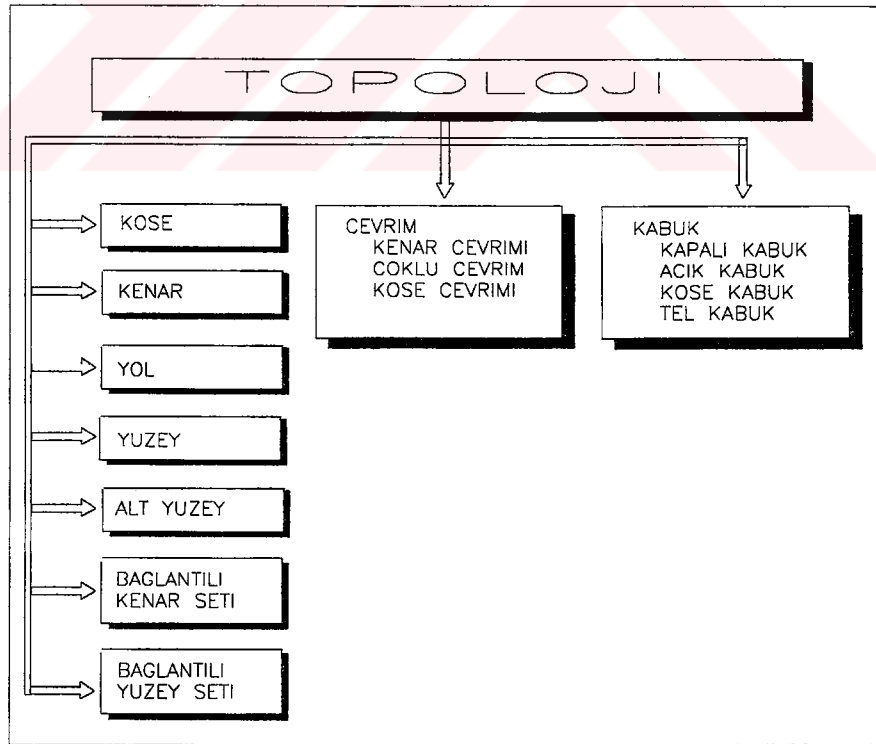
Köşe, kenar ve yüzeyler için geometrik birleşimler, sırasıyla nokta, eğri ve sınırsız yüzeydir. Topolojik varlıklar içerisinde en basit olan varlık köşedir ve tüm diğer topolojik varlıklar, köşelere göre direk veya endirek olarak tanımlanır. Topolojik varlıklar, hiyerarşik bir tarzda tanımlanmaktadır. Bunlardan bazıları aşağıda kısaca verilmiştir.

**Köşe**, bir noktaya karşılık gelen topolojik bir varlıktır.

**Kenar**, iki köşeyi birbirine bağlayan çizgi bir kenarı oluşturur. Kenar, sınırlarını belirleyen bir köşeden, diğerine doğru, hareket yönünün seçimi ile yönlendirilir. Koordinat uzayında bir kenarı yerleştirmek için, belki geometrik bir eğri, herhangi bir kenar ile ilişkilendirilebilir.

**Yol**, kenar ve köşenin birbirini izleyerek oluşturduğu topolojik bir varlıktır. Bir kenar, belirli bir yol tarafından, bir defa referans alınabilir.

**Çevrim**, bir köşeden başlayan ve aynı köşede sona eren köşeler ve kenarların ard arda sıralanması ile oluşturulur.



Şekil 5.5. STEP'te topoloji sınıflandırma yapısı



**Yüzey**, sınırsız bir yüzeyin, çevrimlerle sınırlandırılmış bir parçasına yüzey denir. Bir yüzey, kendi sınırlarını oluşturan çevrimlerle temsil edilir ve her yüzey topolojik bir normale sahiptir. Bir parçayı oluşturan yüzeylerin çevrimleri, dışardan bakılınca, yüzey etrafında saat ibresinin tersine hareket edecek şekilde olmalıdır.

**Kabuk**, bir bölgeyi sınırlandırmak için kullanılan kabuk, kenarlar boyunca yüzeylerin birleştirilmesi ile konstrükte edilir.

#### 5.4.1.4. Şekil özellikleri

Diğer topolojik modellerle birlikte, tek bir modeli oluşturan şekil özellik modelleri, **Şekil Özellik Bilgi Modeli (FFIM)** olarak adlandırılır. Bazı örnek modellere uyan bir şeklin, bir bölümü olarak dikkate alınan şekil özelliği, bazı amaçlar için bir ünite olarak düşünülür. Şekil özellik bilgi modeli, şekli ifade eden bilgiyi sağlar. Bu modellerle, işlemler, fonksiyonellik, yüzey bitirme, toleranslar gibi şekle ait olmayan bilgilere de ulaşılabilir. Özellik, bir şeklin örtüsü olarak kabul edilir. Yüzeleştirilmiş tel kafes, ST, KKG gibi özellik ihtiva eden geometrik modellerin genelinde, şekil özellik verileri uygulanır. Şekil özellik bilgi modeli, "U şeklindeki bir profilin boydan boya süpürülmesi" gibi bir şeklin özelliklerini temsil eder. Kullanıcılar, şekil özelliklerini tanımlamak için, uygulamada kullanılan bazı özel terimleri kullanarak karar vermeye çalışır. Örneğin çeşitli uygulamalarda **kanal**, **oyuk** veya **yarık** gibi bazı şekil özellikleri, U şekil özelliği olarak yorumlanabilir. Şekil özellikleri, açık ve kapalı şekil özellikleri olmak üzere, iki tipte temsil edilir (Şekil 5.6).

**Açık şekil özellikleri:** Bir geometrik modeli oluşturan elemanların gruplandırılmasıdır. Açık şekil özelliklerinde bir özelliği tanımlamak için gerekli olan şekil elemanları ayrıntılı olarak listelenir. Örneğin açık bir cep özelliği, cebi meydana getiren beş adet yüzeyin tanımlanmasıyla açıklık kazanır.

**Kapalı şekil özellikleri:** Kapalı şekil özellikleri, şekil bilgisini, geometrik olmaktan ziyade parametrik olarak modeller. Örneğin bir delik, sınırsız silindirik bir yüzeyi vermek yerine, çap ve eksen çizgisinin verilmesi ile açıklanabilir.

Açık şekil özellikleri, özelliği oluşturan elemanların referans alınmasını sağlar, fakat deliğin derinliği gibi bir özellik parametresine gerek duyulunca bu bilgi sağlanmalıdır. Kapalı şekil özellikleri ise bu tür parametrelere doğrudan

ulaşmayı mümkün kılar. Kapalı şekil özellik temsilileri ilk şekil üzerindeki özelliğin etkisine göre sınıflandırılabilir. Bunlar:

**1- Pasajlar**, ilk şekilden malzeme çıkarılmasıdır. Çıkarılan hacim, ilk şeklin sınırlarında (ilk şekil üzerinde pasajın giriş ve çıkış yaptığı yüzeyler) arakesit oluşturur.



Şekil 5.6. STEP standartta şekil özelliklerinin sınıflandırılması

**2- Girintiler,** ilk şeklin bir ucundan malzeme çıkarılarak elde edilen çukurluktur. Şeklin cinsi değiştirilmez.

**3- Çıkıntılar,** ilk şeklin bir ucuna malzeme eklenerek elde edilen tümsekliktir. Şeklin cinsi değiştirilmez.

**4- Geçişler,** ilk şekil elemanlarının ara kesitlerinin düzgünleştirilmesidir.

**5- Deformasyon,** ilk şeklin bükülmesi ve gerdirilmesidir.

**Birleştirilmiş hacim özellikleri:** Yukarıdaki özelliklerin ilk üçü belirli hacimsel ilişkilere sahiptir. İlk şekle eklenen veya çıkarılan bir hacmin belirlenmesi pasajın, girintinin ve çıkıntının kapalı temsilleriyle sağlanır. Eklenen veya çıkarılan hacimleri belirlemede izlenen yol, süpürme ve bununla ilgili bazı kurallar koymadır. **Özellik Süpürme,** profil ve bir süpürme yoluyla bir hacmi belirler. Şekil özellik bilgi modelinde uç şekillerinin tipi belirlidir. Örneğin simetri eksenli özellik süpürme, küresel, düz veya konik bir uç şekline sahip olabilir. Özellik süpürme esas itibarıyla iki sınıfa ayrılır (Şekil 5.6).

1- Kapalı Özellik Süpürme Profili

A) Diğer Kapalı Özellik Süpürme Profili

B) Standart Kapalı Özellik Süpürme Profili

a) Çokgen Kapalı Özellik Süpürme Profili

b) Diktörtgen Kapalı Özellik Süpürme Profili

2- Açık Özellik Süpürme Profili

A) Dairesel Yay Özellik Süpürme Profili

B) L-Özellik Süpürme Profili

C) Yarım Yuvarlatılmış Özellik Süpürme Profili

D) Çizgi+Radyüs Özellik Süpürme Profili

E) Diğer Açık Özellik Süpürme Profili

F) Yuvarlatılmış U-Özellik Süpürme Profili

G) Kare U-Özellik Süpürme Profili

H) T-Özellik Süpürme Profili

I) V-Özellik Süpürme Profili

#### 5.4.1.5. Boyutlar ve toleranslar

**Boyutlar:** Boyutlar, teorik olarak modellenen cismin tam şeklini belirler. Toleranslar ise, fiziksel cismin müsaade edilen şekil sapmalarını göstermek için,

boyutlara eklenen değerlerdir (Şekil 5.7). Genel olarak açısal, konum ve ölçü boyutları kullanılmaktadır.

**Açısal boyut**, bir özelliğin değerine göre yönlenme ölçüsünü belirleyen değere açısal boyut denir. Açısal boyutun değeri, iki doğru arasındaki açıdır ve bu açılar da açısal parametre boyutu, yönsüz açısal boyut ve yönlü açısal boyuttan ibarettir. Tolerans değeri ise, boyut değerini kısıtlayan sayısal bir aralıktır.

**Yerleştirme boyutu**, bir özelliğin değerine göre yerinin ölçüsüdür. Yerleştirme boyutu, yönlü veya yönsüz olabilir. Yerleştirme boyutunun değeri, şekil temsil elemanlarından türetilen iki paralel şekil temsil etme elemanı arasındaki mesafedir. Yerleştirme boyutundaki her özellik, iki karakteristikten oluşur. Bunlar:

- 1- Boyutun orijini olan varlık,
- 2- Kontrol edilen cisim üzerindeki gerçek şekil temsil elemanı (sınırsız yüzey/alan, eğri veya nokta/köşe)'dir.

Yerleştirme boyutunun değeri, doğrusal bir yol üzerinde ölçülür. Bununla beraber düzlemsel, doğrusal veya silindirik olmayan karmaşık şekil elemanları da boyutlandırılabilir.

**Ölçü boyutu**, bir cisme ait şeklin bir tek özelliğinin ölçüsüdür. Tolerans değeri boyut aralığını kısıtlar. Ölçü boyutunun üç karakteristiği vardır.

- 1- Boyutsal değer,
- 2- Özelliği belirleyen şekil temsil elemanları,
- 3- Özellik elemanlarının simetri merkezi.

Özelliğin yerleştirilmesine bağlı olmayan bir büyüklük olarak ifade edilen ölçü boyutunun değeri, özellik üzerindeki karşılıklı iki nokta arasında ölçülen çizgisel mesafedir. Bu mesafenin belirlenmesi için ölçme işlemi simetri ekseninden geçen bir doğru üzerinde yapılmalıdır. Bu yüzden çoğu durumlarda (bir deliğin çapı veya bir kanalın genişliği gibi) boyutsal değer belirlenmesi gerekmez.



Şekil 5.7. Tolerans ve boyut sınıflandırma yapısı

**Toleranslar:** Anma ölçüsünden müsaade edilebilen sapmaları ifade eder. Tolerans çizelgeleri hem boyut, hem de geometrik toleransları içerir. Tüm tolerans boyutları, şekil/ölçü elemanlarından türetilir veya kapalı özelliklerin bir parçası olarak açıkça ifade edilir.

**Koordinat tolerans aralığı,** bir ürünün nominal boyutsal değerinden çıkarılan veya eklenen sayısal değerlerdir. Koordinat tolerans aralığı, resim üzerindeki bir boyutun geleneksel  $\pm$  tolerans değerlerini temsil eder.

**Geometrik tolerans,** özelliğin nominal tasarım değerinde meydana gelen salgı, yerleştirme, uyum ve şekil sapmasıdır. Geometrik tolerans, açısallık, dairesel salgı, dairesellik, eş eksenlilik, silindiriklik, düzlük, paralellik, diklik, konum, profil çizgisi, doğrusallık ve toplam salgıyı kapsar.

### 5.4.2. Mantıksal katman

Daha önce bahsedildiği gibi özel modeller, uygulama katmanında geliştirilir. **Güncel Modeller** olarak adlandırılan veri modelleri, **Bütünleşik Ürün Veri Modeli (IPIM)** veya **Mantıksal Katman** isimi verilen komple bir bilgi modelinde toplanır. Bu nedenle IPIM, **EXPRESS** dilinde formülize edilen varlıklar, toleranslar, özellikler, geometri ve şekil modelini içerir.

**Varlıkları modelleme:** Bir varlık kendisini oluşturan çeşitli niteliklerle tanımlanır. Üç boyutlu uzayda NOKTA varlığının X,Y,Z nitelikleriyle tanımlanması buna bir örnektir. Her varlık **tip** ve **isim** olmak üzere iki parçadan meydana gelir. Örneğin bir **DAİRE**, Merkez, Eksen ve Radyüs olmak üzere üç nitelikle tanımlanır. Burada Merkez (isim) bir nokta (tip), Eksen bir vektör ve Radyüs (isim) gerçek bir sayı (tip) dir.

**Kuralları Modelleme:** Bir kural, doğru olması gereken bir durumu ifade eder. Kurallar doğru olduğu sürece veri tabanı geçerli olur. Kurallar çeşitli koşulları açıklamak için yazılabilir. Örneğin;

- Bir radyüs sıfırdan az olamaz.
- Bir kenar iki köşeden daha fazlasına sahip olamaz vs.

Bir varlığın bir kısmını açıklamak üzere yazılan kural, lokal kural olarak isimlendirilir. Lokal kurallar, sadece varlıkta mevcut olan niteliklerin değerlerine uygulanır. Şemanın bir kısmı olarak yazılan kural, global kural olarak isimlendirilir. Global kurallar, kuralın uygulandığı her tip varlığın her durumu ile ilgilidir.

**İşlemleri modelleme:** Bir işlem, herhangi bir varlığın nasıl kullanılabileceğini açıklar. Sayılar üzerinde yapılan toplama, çıkarma, çarpma, bölme gibi işlemler, ünersaldır. Sayılara uygulanan işlemler, matematiği kullanan herkese bir çerçeve sağlar. Geometrik elemanlara, malzeme faturalarına ve diğer ürün verilerine aynı sorgulama uygulanır. Bir işlem sembolik bir gösterime sahiptir. Toplama;

"sayı+sayı" şeklindeki bir notasyonla yazılır.

Burada iki sayı işlem görenler ve '+' sembolü ise bir operatör olarak açıklanır. Çeşitli programlama dilleri, aynı işlemi farklı şekillerde yerine getirir. Pascal, bir sayının her hangi bir sayıya bölündüğünde kalanı,  $X \text{ MOD } (X \text{ MOD } Y)$  işlemiyle, Prolog ise aynı işlemi " $X \text{ mod } Y$ " şeklinde yapar. EXPRESS dili, fonksiyonlarla yazılacak işlemleri yerine getirir. Benzer işlemleri yerine getiren tüm diller, izin verilebilen bütün giriş tipleri için, özel bir fonksiyon geliştirme ihtiyacından kaçınmak maksadı ile kullanılabilir.

#### 5.4.2.1. EXPRESS dili

EXPRESS, bilgileri doğru bir şekilde yapılandırmaya imkan tanıyan PASCAL veya ADA'ya benzer bir dildir. Oluşturulan bu yapıyı, insanlar anlayabileceği gibi, bilgisayarlar tarafından da direkt olarak kullanılabilir. STEP standardı için tasarlanmış olan ekspres dili, açıklama ve icra edilebilir deyimler ile algoritmaları içeren güçlü bir veri modelleme dilidir. EXPRESS dilinin icra edilebilir deyimleri ve algoritmaları, ürün veri varlıklarının nitelikleri üzerine kısıtlar koymak ve bu varlıkların tanımını netleştirmek ve açıklamak için kullanılır. İcra edilebilir deyimler, değiştirme formatında ayrıntılı olarak yer almaz. EXPRESS dili dört temel isteği yerine getirmek üzere tasarlanmıştır.

- İlgilenilen varlıkları modelleme,
- Varlıklar üzerine konulacak kısıtları belirleme,
- Varlıklarda yapılacak ortak işlemleri belirleme,
- Bilgisayara uyumlu tarzda modelleme.

EXPRESS'in ana tasvir elemanları şunlardır:

Şema

Tip

Varlık

Algoritma

Fonksiyon

Prosedür

Kural

**Şema**, ilgilenilen bir alan için tip, varlık, fonksiyon ve benzer ifadelerden meydana gelen bir açıklama iken **Tip**, varlıkları temsil etmek için kullanılan verilerin karakteristiklerini; **Varlık**, ilgilenilen nesnelere ve **Kurallar** da varlıklar ve nitelikler üzerindeki kısıtları belirler. **Fonksiyonlar**, **prosedürler**



ve **kurallar**, nesnelerin davranışlarını açıklamak için kullanılır. EXPRESS dilinin daha geniş bir açıklaması Ek-A ve Ek-B'de verilmiştir.

### 5.4.3. Fiziksel katman

EXSPRESS'te tüm ürün veri varlıkları tanımlandıktan sonra, **Wirth Syntax Notation (WSN)**'ye göre yapılandırılmış olan fiziksel dosyaya yerleştirilir. Fiziksel dosyanın WSN'ye göre hazırlanması Ek-A'da açıklanmıştır.

## 5.5. STEP Dosyasının Prolog Cümleciklerine Dönüştürülmesi

Bir parçanın STEP dosyası oluşturulduktan sonra, dosyanın Prolog cümlecikleri formuna dönüştürülmesi gerekir. İşlem planlama sistemi Prolog'da yazıldığından, STEP'te yapılan tanımlamalar uygulama için Prolog formatına dönüştürülmelidir. Bu çalışmada örnek olarak seçilen bir parçanın STEP ve Prolog formatları EK-B'de ayrıntılı olarak açıklanmıştır.

### 5.5.1. Prolog dizini

Prolog programları, cümleciklerden meydana gelir. **Kurallar** ve **olgular**dan meydana gelen cümlecikler bir nokta ile sona erer. Olgular ve kurallar, sabitler olarak isimlendirilen **atomlardan** veya **değişkenlerden** meydana gelir. Tipik bir Prolog cümlesi, bir **ilişki ismi** ve bu ilişki ismine ait **argümanlardan** oluşur. Bir ilişki ismi, kendisiyle karakterize edilir. Argümanlar aynı zamanda ilişki ismine ait parametreler olup ya küçük harflerden oluşan atomlar (sabitler)'dir, ya da büyük harflerden oluşan değişkenlerdir. Örneğin basit bir olgu aşağıdaki şekilde yazılabilir.

dış\_yüzey(yüzey\_350).

Bu ifade şu şekilde açıklanır; "Yüzey 350 dış yüzeydir" burada **dış\_yüzey** ilişki ismi ve **yüzey\_350** argümandır.

**Olgular**, bir veya daha çok argümana sahip olan ve daima bir nokta ile sona eren atomik bir formüldür. Örneğin,

sahip(yüzey\_350, özellik(l\_kademe)).

Bu ifade şöyle açıklanır; "yüzey\_350, bir l-kademe özelliğine sahiptir".

**sahip** ilişki isimli yukardaki örnekte, cümlenin ikinci argümanı (parametresi) bir olgudur ve bu "özellik (1\_kademe). " ifadesidir. Bu durumda, özellik **funktor, 1\_kademe** de bunun argümanı olmaktadır.

**Kurallar**, bir **koşul** (kuralın sağ tarafı) ve bir de **sonuç** (kuralın sol tarafı) kısımlarına sahiptir. Kurallar ve olgular arasında önemli bir fark vardır. Olgular, daima şart koşulmayan doğrudur. Diğer taraftan kurallar, bazı koşullar sağlanırsa, sonuçların gerçekleşebileceğini belirler. Sonuç kısmı cümlenin **başı**, koşul kısmı ise cümlenin **gövdesi** olarak isimlendirilir. Cümlenin baş kısmı gövdeden **"IF"** olarak da okunan ":-" sembolü ile ayrılır.

dikdörtgen(Yüzey) :-  
sahip(Yüzey, (kenar(4))).

Yukardaki ifade "eğer yüzey (değişken) dört kenara sahipse, yüzey bir dikdörtgendir" şeklinde okunabilir.

**Birleşik veri objeleri**, çeşitli bilgileri tek bir parça olarak işleme tabi tutmaya imkan veren yapılardır.

eksen2\_yerlestirme(koordinat(0,0,0) ,vektor(0,0,1) ,vektor(1,0,0)) .

Bu örnekte "**eksen2\_yerlestirme**" birleşik objesine ait koordinat(0,0,0), vektor(0,0,1) ,vektor(1,0,0) olmak üzere üç obje parçası vardır. Birinci objenin funktoru **koordinat**, ikinci ve üçüncü objenin funktoru **vektor**'dur. Prolog cümlelerinin bu yapılarından faydalanarak program yazmada çeşitli kolaylıklar sağlanır. Örneğin "eksen2\_yerlestirme" ilişki ismine ait üç argüman, bir değişken olan (X)'e bağlanabilir. Örnek:

eksen2\_yerlestirme(X).

Bununla beraber bu ilişki ismine ait argümanlar farklı üç değişkene de (X,Y,Z gibi) bağlanabilir.

eksen2\_yerlestirme(X ,Y, Z).

**Listeler**, köşeli parantezler içerisine alınmış ve virgüllerle ayrılmış bir çok parametrenin bir sıralamasıdır. Örnek:

([köşe1, köşe2, köşe3, köşe4, köşe5]).

Bir liste iki kısımdan meydana gelir. Listenin ilk kısmına **Baş** (Head), geride kalan kısmına ise **Kuyruk** (Tail) denir. Bir listenin başı, daima bir, ya da daha çok elemandan; kuyruk kısmı ise, bir listeden meydana gelir. Bir liste her zaman baş ve kuyruk olmak üzere ikiye ayrılabilir. Örnek:

[köşe1, köşe2, köşe3, köşe4, köşe5] bir liste ise, liste baş ve kuyruk olarak ayrıldığında; listenin başı "köşe1" ve listenin kuyruğu da "[ köşe2, köşe3, köşe4, köşe5]" olur. Eğer liste tek elemanlı ise, örneğin [köşe1], bu listenin başı " köşe1", kuyruğu ise boş "[ ]" listedir. Boş bir liste baş ve kuyruk olarak bölünemez. Bir listenin başı, kuyruğundan dik taksim "|" işareti ile ayrılır. Örneğin;

[köşe1, köşe2, köşe3, köşe4, köşe5] Listesi  
[köşe1 | köşe2, köşe3, köşe4, köşe5] şeklinde veya;  
[köşe1 | Kuyruk] şeklinde ayrılır.

Prolog'da liste elemanlarını işleme tabi tutmak üzere, çeşitli kontrol yapıları geliştirilmiştir.

## 6. STOK SEÇİMİ, TASLAK VE BİTMİŞ PARÇA MODELLEME

İşlem planlamada taslak parça tasarımı, öncelikli yapılması gereken bir işlemdir. Bu yüzden taslak parça, talaş kaldırma işlemi başlamadan önce tasarlanmalıdır. Bitmiş parça modeli esas alınarak taslak parça tasarlanır ve buna bağlı olarak stok boyutları seçilir. Taslak parça boyutları, bitmiş parçanın en büyük x, y, z koordinat değerleri dikkate alınarak belirlenir. Bu nedenle belirlenen taslak parça boyutları, bitmiş parça modelini içine alacak şekilde tayin edilir. Taslak parça için stok, veri tabanındaki stok ölçü dosyasını araştırarak seçilir. Taslak parça, Komşu Yüzey Grafiği (KYG)' de açıklandıktan sonra veri tabanına yazılır.

### 6.1. Komşu Yüzey Grafiği (KYG)

İşlem planlama süreci sırasında, her işlemten sonra parçadan talaş kaldırılır. Bu yüzden daha ileride yapılacak operasyonlarda işlem planlama ve bağlama işlemi hakkında güvenilir kararların verilmesi için, yeni araparça modellerin oluşturulması gerekir.

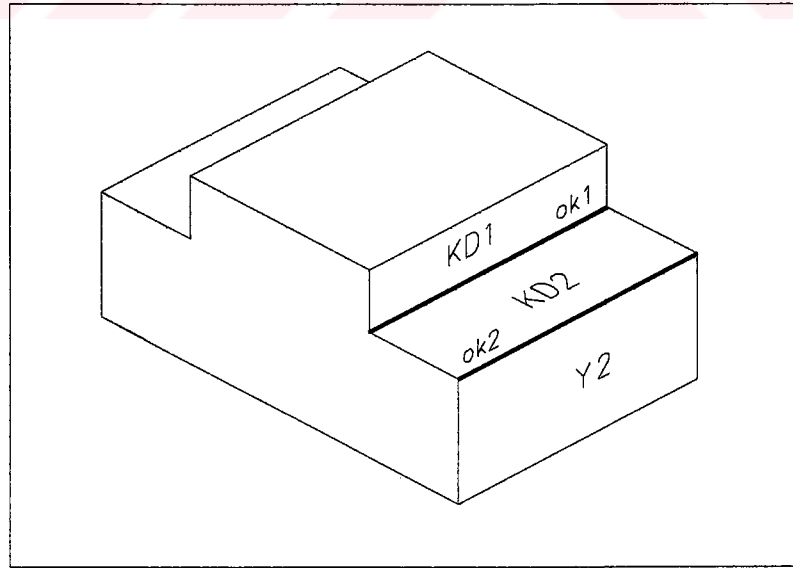
ST modeli (STEP dosyasında), yüzeyler, özellikler, köşeler ve kenarları içerse bile, yüzeyler arasındaki açılar, komşuluk ve içbükeylik-dışbükeylik ilişkileri gibi bazı ilave bilgilerin, imalat işlemi seçimi ve bağlama işlemi için, veri tabanında açık bir şekilde bulunması gerekir. İşlem planlama sisteminin, STEP dosyasını doğrudan kullanarak istenen sonuçları elde edebilmesi için, varlıklarını araştırması (örneğin bir köşenin koordinatlarını bulma gibi) bir hayli zordur. İşlem planlama sistemi tarafından ihtiyaç duyulan bilgilere, en kısa yoldan ve hızlı bir şekilde ulaşılabilmesi gerekir. Ayrıca bu bilgilere, işlem planlama ve bağlama faaliyetleri hakkında doğru kararlar vermek için, gerekli olan araparçaları, veri tabanında modellemek üzere ihtiyaç duyulur. Bu sebeple, taslak, bitmiş ve araparça modeller KYG'de temsil edilir. KYG, yüzeyler arasındaki açılar, komşuluk ve içbükeylik-dışbükeylik ilişkileri gibi daha açık bilgilere sahiptir. Bu yüzden işlem seçimi ve bağlama işlemi için elverişli bir yapıda olup araparça modellerin oluşturulmasında kolaylık sağlar. Böylece KYG, hem işlem planlama, hem de bağlama faaliyetlerinde kullanılabilir.

STEP dosyasını kullanmak suretiyle yüzeyler ile özellikler belirlenir ve yüzeyler arasındaki açılar hesaplanır. Ayrıca yüzeyler arasındaki komşuluk ve içbükeylik-dışbükeylik ilişkileri tayin edilir. Böylece parça modeli, üretim kurallarını sağlayacak bilgi bakımından, elverişli bir formda temsil edilmiş olur. Parça ilk önce yüzeyler seti olarak açıklanır.

Parça = {yüzeyler seti} ve her yüzey şu şekilde temsil edilir :

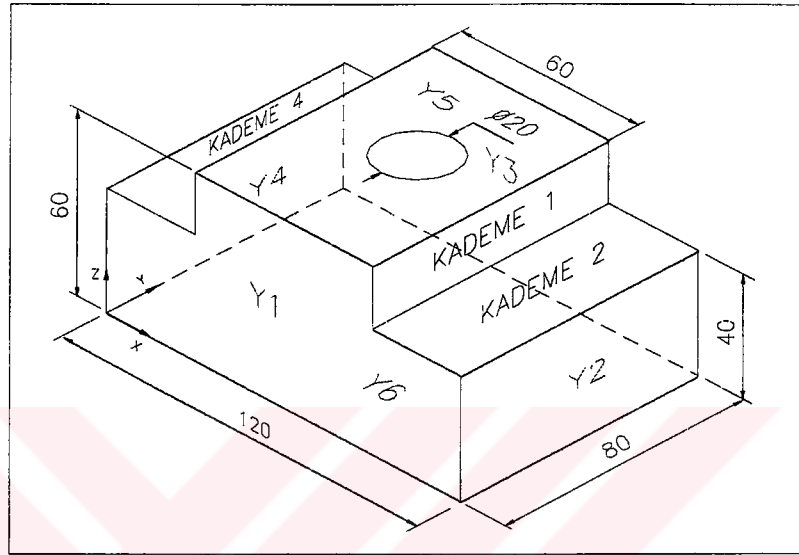
Yüzey = {yüzeyler, özellikler, yön, kenar çevrimi}.

Burada **yüzeyler**, açıklanan yüzeyin komşu yüzeyleri; **özellikler**, bu yüzeydeki alt özellikler (delik, kanal, kademe vb.); **yön**, yüzeyin yönü; **kenar çevrimi**, yüzeyi oluşturan kenarlar listesini ve yüzeyin köşelerini kapsar. Her yüzey, komşu bir yüzey ile ortak bir kenarı paylaşır. Komşu yüzeyler, ortak kenarları boyunca bir açı ve içbükeylik-dışbükeylik ilişkisi gibi özelliğe sahiptir. Şekil 6.1'de görüldüğü gibi **ok1** ortak kenarını paylaşan **KD1** yüzeyi, **KD2** yüzeyine komşudur ve **içbükey** bir açıyı oluşturur. Bunun yanında **ok2** ortak kenarını paylaşan **KD2** ve **Y2** yüzeyleri, **dışbükey** bir açıyı meydana getirir. Her yüzey, STEP dosyasında özel bir varlık numarasıyla tanımlanmıştır. Bu nedenle yüzeylerle ilgili herhangi bir bilgiye ihtiyaç duyulduğunda, ilgili bilgi STEP dosyasından sağlanabilir.

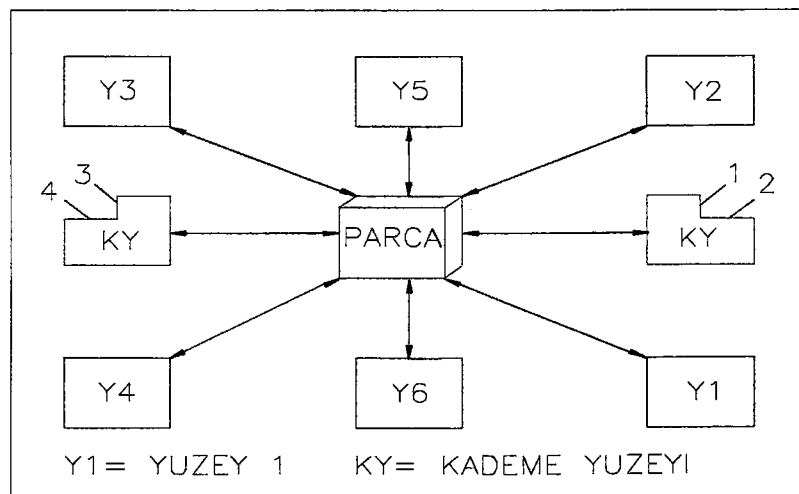


Şekil 6.1. Yüzeyler arasındaki içbükeylik-dışbükeylik ilişkisi

Örneğin Şekil 6.2'de gösterilen parça, ilk önce yüzeyler seti olarak temsil edilir (Şekil 6.3). Her yüzey için komşu yüzeyler, yüzey ve komşu yüzeyler arasındaki açılar ve içbükeylik-dışbükeylik ilişkileri, yön, noktaların koordinatları ile kenarlar listesi ve alt özellikler belirlenerek KYG'de temsil edilir.



Şekil 6.2. KYG için örnek parça



Şekil 6.3. KYG'de parça temsil etme

### 6.1.1. KYG'de yüzey temsil etme

KYG'de temsil edilecek parçanın yüzey listesi, STEP dosyasından alınır. Listeden ilk yüzey alındıktan sonra, program, bu yüzeyin herhangi bir özellik içerip içermediğini kontrol eder. Herhangi bir özellik içeriyorsa, yüzey buna göre işleme tabi tutulur.

STEP standartta **kanal** ve **kademe** gibi bazı özellikler süpürme; **delik** gibi dairesel kesitli olanlar ise, dönme hareketiyle oluşturulur. Bu yüzden **kademe** ve **kanal** özelliklerinin köşe koordinatları, uyumu, yerleşmesi ve delik derinliği gibi bazı bilgilerin bulunması gerekir. Herhangi bir yüzeye bir özellik yerleştiğinde, bu özelliğin diğer yüzeyler üzerindeki etkisinin dikkate alınması gerekir. Örneğin Şekil 6.4'te görüleceği gibi parçaya iki kademe özelliği yerleştiğinde **Y1**, **Y2**, **Y3**, **Y4** ve **Y5** yüzeyleri bunlardan etkilenecektir (Şekil 6.5).

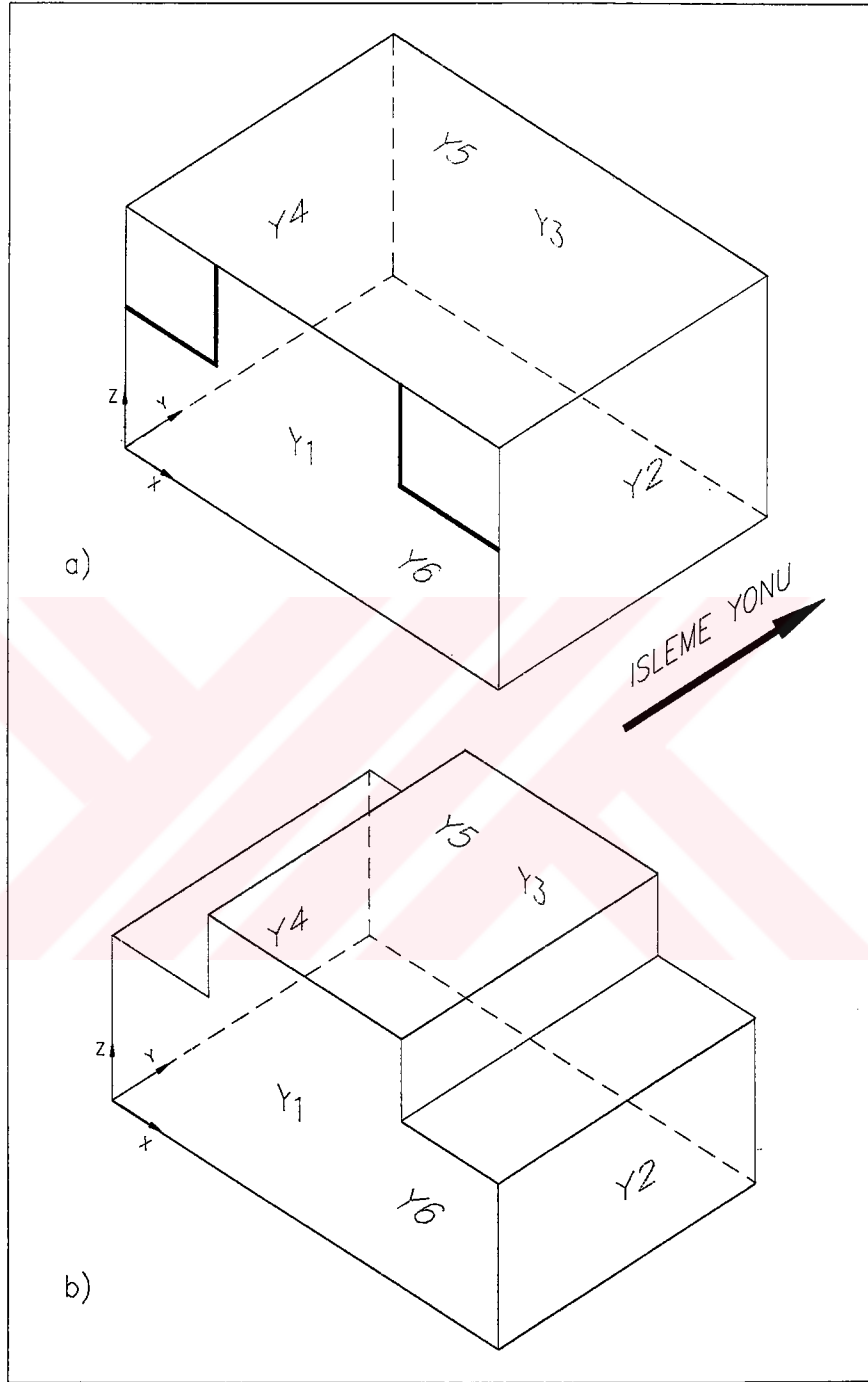
Özelliğin profili işlendikten sonra, profilin yüzeyleri değerlendirilerek yüzeylerin yerleşimi, uyumu ve profilin yüzeylerinin köşelerini meydana getiren noktaların koordinatları bulunur. Bir özelliğe ait yüzeyin köşelerinin koordinatları bulunurken özelliğin yüzeyleri tarafından etkilenen yüzeyler de belirlenmiş olur. Özellik tarafından etkilenen her yüzeyin yeni boyutları elde edilerek komşu yüzeyleri belirlenir. Ayrıca komşu yüzeyler arasındaki açılar ve içbükeylik-dışbükeylik ilişkileri bulunur. Daha sonra özellikler tarafından etkilenen komşu yüzeyler, yeni durumlarına göre KYG'de temsil edilir.

Örneğin Şekil 6.2'deki **Y1** yüzeyi, iki **kademe** özelliğine ait yüzeylerle beraber, sekiz komşu yüzeye sahiptir. **Y1** yüzeyi, KYG'de Şekil 6.6'da gösterildiği gibi temsil edilir. Komşu yüzeyler Prolog'da köşeli parantezler içerisinde alınmış bir listeye yerleştirilir. **Y1** yüzeyinin Prolog formatına göre temsil edilmesi aşağıdaki şekilde olur.

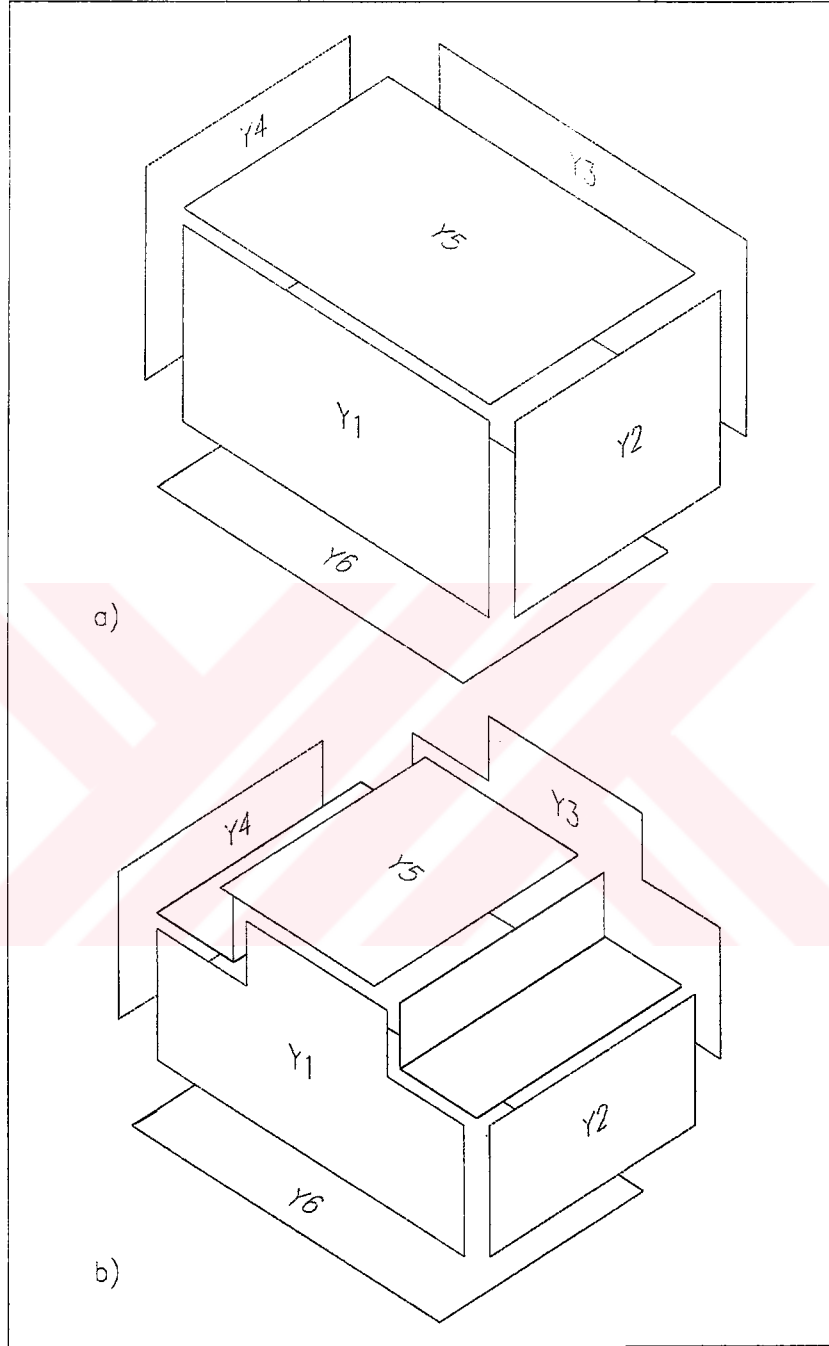
```
yüzey(b, dz, 350, [kyüzey(dz , 355, 90_db), kyüzey(dz , 365,90_db),
kyüzey(445,kademe(1),90_db),kyüzey(445,kademe(2),90_db),
kyüzey(450,kademe(3),90_db),kyüze(450,kademe(4),90_db),
kyüzey(dz,435,90_db) ,kyüzey( 455,90_db)],y_n).
```

```
boyutlar(b,350, [koord(0, 0, 0), koord(120, 0, 0), koord(120, 0, 30),
koord(90, 0, 40), koord(90, 0, 60),koord(30, 0, 60), koord(30, 0, 40),
koord(0,0,40), koord(0,0,0)]).
```





Şekil 6.4. L kademe özelliklerini oluşturmak için parçaya yerleştirme ve işleme



Şekil 6.5. Parçaya özellikleri yerleştirme öncesi ve sonrası yüzeylerin durumu

Her komşu yüzey, yüzeyin tipini, STEP numarasını, açısını ve içbükeylik-dışbükeylik ilişkisini içerir. Örneğin ikinci yüzey (Y2) için, **b** bitmiş parçayı, **dz** düzlem yüzeyi, **355** STEP'ten alınan varlık numarasını, **90\_db** açığı ve içbükeylik-dışbükeylik ilişkisini gösterir. **db** dış bükey olduğunu, **y\_n** ise yüzeyin yönünü gösterir. Burada **y** eksen, **n** veya **p** yüzeyin yönünün negatif veya pozitif olduğunu temsil eder. Bu örnekte **Y1** yüzeyi **-y** yönündedir.

**boyutlar** ilişki ismi, yüzeylerin köşelerine ait noktaların koordinatları ile birlikte, dolaylı olarak kenar çevrimini de gösterir. Kenarların yönü saat ibresinin ters yönündedir. Burada yüzeyin kenar çevrimi (0,0,0) noktasında başlar, ilgili noktalardan geçerek yüzeyin çevresini dolaşır aynı noktada son bulur. Şekil 6.2'de verilen parçaya ait **Y1** yüzeyinin köşelerinin koordinatları ve yüzey çevrimi, Şekil 6.7'de gösterilmiştir.

### 6.1.2 KYG'de delik temsil etme

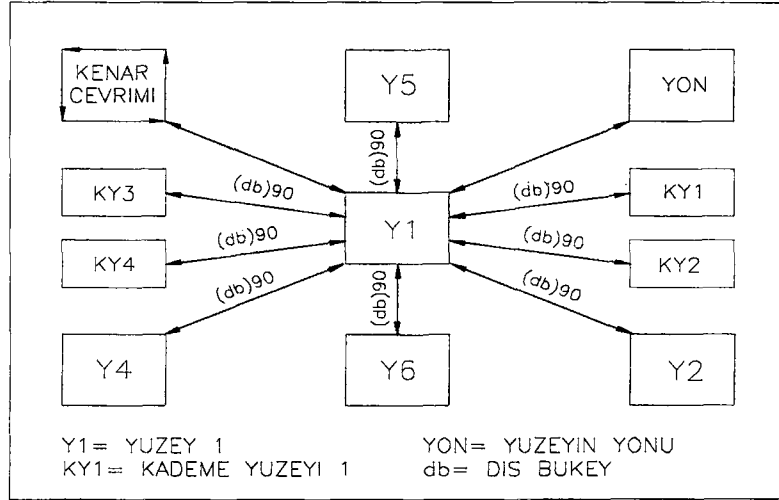
Herhangi bir yüzeye bir delik özelliği yerleştiğinde, deliğin STEP dosyasındaki özel numarası, açısı, çapı ve deliğin uç tipi ilgili yüzeyde gösterilmelidir. Şekil 6.2'de gösterilen parçanın üst yüzeyi Y5, dört komşu yüzeye ve bir deliğe sahip olup, KYG'de Şekil 6.8'de gösterildiği gibi temsil edilir. Y5 yüzeyinin grafiği planlandıktan sonra bu yüzey, Prolog'da aşağıdaki şekilde görülür.

**yüzey**(b, dz, 435, [delik(440, düz, 90\_20), kyüzey(445, kademe(1), 90\_db), kyüzey(450, kademe(3), 90\_db), kyüzey(dz, 350, 90\_db), kyüzey(dz, 360, 90\_db)], z\_p).

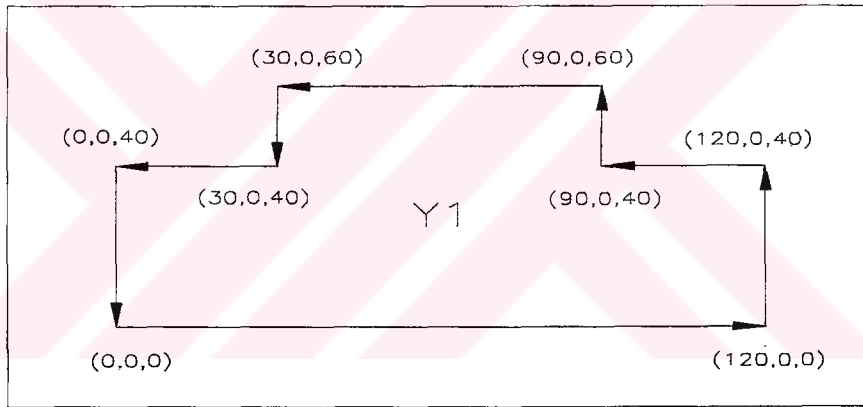
**boyutlar**(b, 435,[koord(30, 0, 60), koord(90, 0, 60), koord(90, 80, 60), koord(30, 80, 60), koord(30, 0, 60)]).

**yüzey** ilişki ismi, **Y5** (435) yüzeyine ait beş özelliğin niteliklerini temsil etmektedir.

**boyutlar** ilişki ismi, yüzeyin köşelerine ait noktaların koordinatlarını ve dolaylı olarak kenar çevrimini göstermektedir.



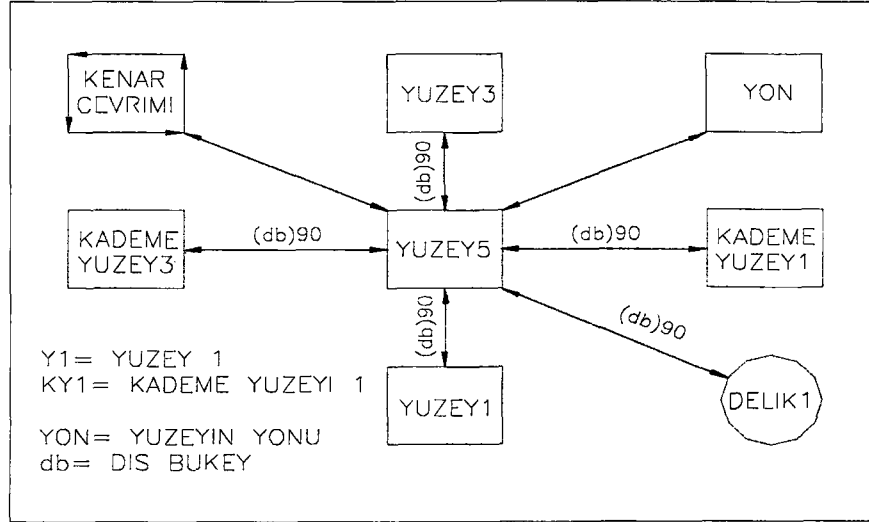
Şekil 6.6. Örnek parçanın Y1 yüzeyinin KYG'de temsil edilmesi



Şekil 6.7. KYG'de Y1 yüzeyinin kenar çevrimi ve koordinatları

## 6.2. Stok Seçme ve Taslak Parça Oluşturma Modülü

İşlem planlama faaliyetlerinde taslak parça tasarımı, kare çubuk, lama, platina veya döküm, dövme gibi taslak parçanın imalat işlemine bağlı olarak gerçekleştirilir. Bu çalışmada taslak parçanın kare çubuk, lama veya platinadan seçileceği kabul edilmiştir. Bununla birlikte taslağın, döküm ve dövme gibi parçalardan tasarlanacağı planlanırsa, bunlarla ilgili bilginin sisteme ilave edilmesi gerekir.



Şekil 6.8. KYG'de Y5'in temsil edilmesi

### 6.2.1. Stok boyutlarının belirlenmesi

İşlem planlama prosedürünün ilk görevi, bitmiş parçayı esas alarak, uygun stok ölçüsünü seçmek ve taslak parçanın şeklini belirlemektir. Seçilen stoğun boyutlarını, bitmiş parçanın ölçüsü tayin eder. Fazla miktarda talaş kaldırmak her bakımdan ekonomik olmayacağı için, ideal olarak seçilen stok ölçüsü, bitmiş parçanın ölçüsüne mümkün mertebe yakın olmalıdır. Bu çalışmada sistem, stok ölçüsünü minimum talaş kaldıracak şekilde seçmektedir.

STEP dosyasından bitmiş parçanın köşelerine ait noktaların en büyük  $x$ ,  $y$ ,  $z$  değerleri bulunur ve her yüzeye 2.5 mm işleme payı ilave edilerek taslak parça boyutları belirlenir. Böylece taslak parça, bitmiş parça modelini kapsayacak ölçülerde olur. Sonra taslak parça, veri tabanındaki stok ölçü dosyasının araştırılmasıyla seçilir. Seçilen boyutlar, stok ölçü dosyasında bulunmadığı zaman, veri tabanındaki stok ölçü boyutları, buna uygun olmak üzere artırılır.

X eksenı yönünde stoğun uzunluğunun istenilen boyutta kesileceği kabul edilirse, sistem Y-Z düzleminde stok için, istenilen enine kesiti bulmaya çalışır. Seçilen taslak, bitmiş parçayı oluşturmak için yeterli büyüklükte olmalıdır. Bu kademede kaldırılacak talaş için gerekli malzeme ilavesi, stok boyut dosyasında mevcut olan stok malzemesi ve hesaplanan boyutlar arasında bir karşılaştırma ile belirlenir.

### 6.2.2. İdeal boyutun seçimi

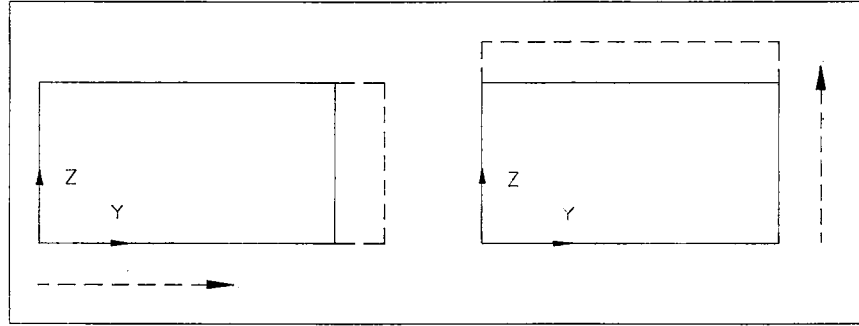
Taslak parçanın hesaplanan boyutları, stok ölçüsü dosyasında, tesadüfen bir stok ölçüsü olarak bulunursa, bu durumda bu ölçü stok ölçüsü olarak seçilir. Stoğun en küçük kesitini elde etmede aşağıdaki prosedür uygulanır. Burada dikkate alınması gereken üç alternatif vardır.

- 1- Z boyutu (yükseklik) stok ölçü dosyasında olabilir.
- 2- Y boyutu (genişlik) stok ölçü dosyasında olabilir.
- 3- Her iki boyut (yükseklik ve genişlik) stok ölçü dosyasında olmayabilir.

**Z boyutu stok ölçü dosyasında varsa:** Parçanın hesaplanan Z boyutunun, stok ölçü dosyasında olup olmadığı kontrol edilir. Eğer bu ölçü dosyada varsa, parçanın Y boyutu stoğun Y boyutu ile karşılaştırılır. Eğer bunlar aynı değerde ise, bu boyut seçilir. Aksi halde Z boyutu değişmeden, Y boyutu artırılarak, bulunan yeni Y-Z değerlerinin veri tabanında olup olmadığı araştırılır. Eğer arttırılan Y boyutu dosyada yoksa, prosedür bundan sonra en yakın Y boyutu bulununcaya kadar tekrarlanır.

**Y boyutu stok ölçü dosyasında varsa:** Parçanın hesaplanan Y boyutunun, stok ölçü dosyasında olup olmadığı kontrol edilir. Bu ölçü dosyada varsa, parçanın Z boyutu, stoğun Z boyutu ile karşılaştırılır. Şayet bunlar aynı değerde ise, bu boyut seçilir. Aksi halde Y boyutu değişmeden, Z boyutu artırılarak bulunan yeni Y-Z değerlerinin veri tabanında olup olmadığı araştırılır. Eğer arttırılan Z boyutu dosyada yoksa, prosedür bundan sonra en yakın Z boyutu bulununcaya kadar tekrarlanır.

**Stok ölçü dosyasında (Y-Z) boyutları yoksa:** Bu durumda hesaplanan Y boyutu, dosyadaki bir ölçü ile eşit oluncaya kadar arttırılır. Benzer şekilde hesaplanan Z boyutu da arttırılır. Arttırılan Y-Z boyutları dosyadaki Y-Z boyutları ile karşılaştırılır. Değerler uyduğu takdirde işleme son verilir. Aksi halde hesaplanan Y-Z boyutları dosyadaki kesit boyutlarından birine eşit oluncaya kadar prosedür tekrarlanır (Şekil 6.9).



Şekil 6.9. Stok seçiminde Y ve Z boyutlarının artırılması

### 6.2.3. Taslak parça oluşturma

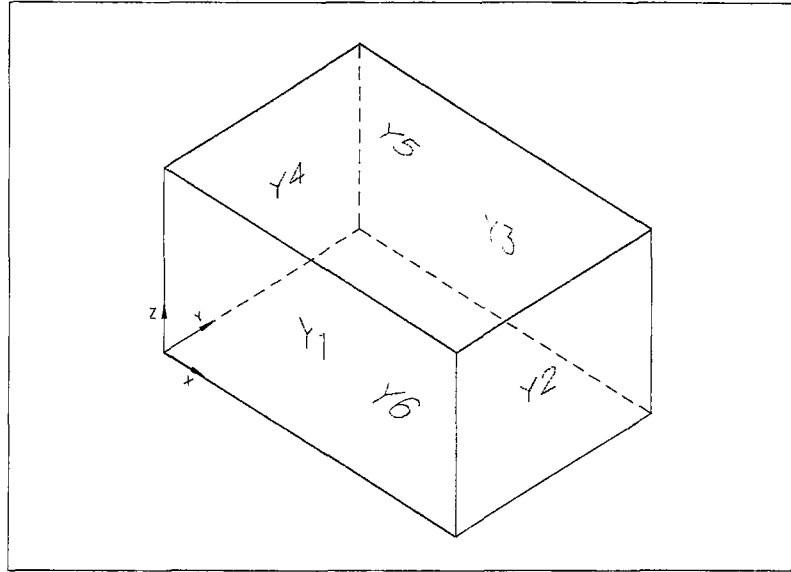
Taslak parça modeli, gerekli bilgiler sağlandıktan sonra, KYG'de temsil edilir. Her yüzeyin komşu yüzeyleri, STEP dosyasındaki ortak kenarların kontrol edilmesiyle bulunur. Ayrıca yüzeylerin yönleri de STEP dosyasından elde edilir. Taslak parça kare çubuk, lama veya platinadan seçildiğinde, yüzeyler birbirine göre dışbükey konumda, bunlar arasındaki açılar da  $90^\circ$  olacaktır. Yön, komşu yüzeyler, açılar ve içbükeylik-dışbükeylik ilişkisi içeren detaylı yüzey bilgisine sahip taslak model, KYG'de temsil edilerek taslak model dosyasına yazılır.

### 6.2.4. Dosyaya taslak modeli yazma

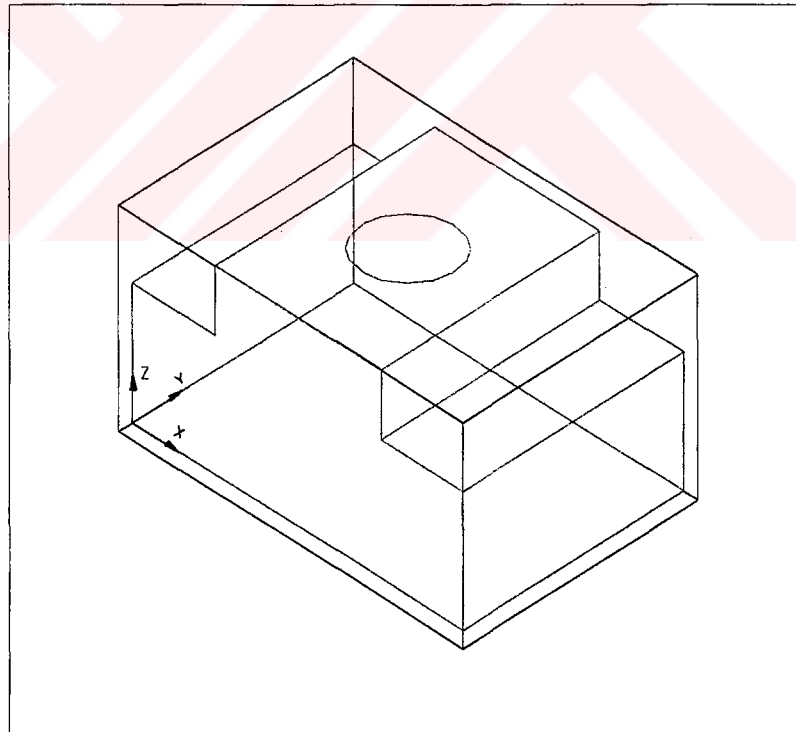
STEP dosyasındaki yüzey listesinden bir yüzey alınarak bu yüzeyin kenar listesine erişilir. Kenar listesindeki ortak kenarları kontrol etmek suretiyle, yüzeyin komşu yüzeyleri belirlenir ve komşu yüzeyler içbükey ve dışbükey olma durumlarına göre, açıları da içerecek şekilde, taslak dosyasına yazılır. Ayrıca yüzeyin yön vektörü de kontrol edilerek yüzeyin yönü dosyaya yazılır. Bu prosedür yüzey listesi boşalincaya kadar tekrarlanır. Daha sonra bitmiş parçanın en büyük  $x$ ,  $y$ ,  $z$  değerleri bulunup bu değerlere 2.5 mm işleme payı eklenir. İşleme paylarının eklenmesiyle daha önceden tayin edilen stok ölçüsü boyutlarına erişilir.

Şekil 6.2'de verilen örnek parça için sistem tarafından tasarlanan taslak parça modeli, Şekil 6.10 ve 6.11'de gösterilmiştir.





Şekil 6.10. Örnek parça için oluşturulan taslak parça



Şekil 6.11. Örnek parçayı kapsayan taslak parça

Tasarlanan taslak parçanın KYG'de, Prolog cümlecikleri şeklinde temsil edilmesi aşağıda gösterildiği gibidir.

```
% ***** Stok seçimi ve taslak parça tasarımı
parca_adi(Parca2).
secilen_stok_olcusu(125,85,65).
yuzey(t,dz,350,[kyuzey(t,dz,455,p_90),kyuzey(t,dz,355,p_90),kyuzey(t,dz,435,p_90),
kyuzey(t,dz,365,p_90)],y_n).
yuzey(t,dz,355,[kyuzey(t,dz,455,p_90),kyuzey(t,dz,360,p_90),kyuzey(t,dz,435,p_90),
kyuzey(t,dz,350,p_90)],x_p).
yuzey(t,dz,360,[kyuzey(t,dz,455,p_90),kyuzey(t,dz,365,p_90),kyuzey(t,dz,435,p_90),
kyuzey(t,dz,355,p_90)],y_p).
yuzey(t,dz,365,[kyuzey(t,dz,455,p_90),kyuzey(t,dz,350,p_90),kyuzey(t,dz,435,p_90),
kyuzey(t,dz,360,p_90)],x_n).
yuzey(t,dz,435,[kyuzey(t,dz,350,p_90),kyuzey(t,dz,355,p_90),kyuzey(t,dz,360,p_90),
kyuzey(t,dz,365,p_90)],z_p).
yuzey(t,dz,455,[kyuzey(t,dz,360,p_90),kyuzey(t,dz,355,p_90),kyuzey(t,dz,350,p_90),
kyuzey(t,dz,365,p_90)],z_n).

boyut(t,350,[koord(-2.5,-2.5,-2.5),koord(122.5,-2.5,-2.5),koord(122.5,-2.5,62.5),
koord(122.5,82.5,62.5),koord(-2.5,-2.5,-2.5)]).
boyut(t,355,[koord(122.5,-2.5,-2.5),koord(122.5,82.5,-2.5),koord(122.5,82.5,62.5),
koord(122.5,-2.5,62.5),koord(122.5,-2.5,-2.5)]).
boyut(t,360,[koord(122.5,82.5,-2.5),koord(-2.5,82.5,-2.5),koord(-2.5,82.5,62.5),
koord(122.5,82.5,62.5),koord(122.5,82.5,-2.5)]).
boyut(t,365,[koord(-2.5,82.5,-2.5),koord(-2.5,-2.5,-2.5),koord(122.5,82.5,62.5),
koord(-2.5,82.5,62.5),koord(-2.5,82.5,-2.5)]).
boyut(t,435,[koord(122.5,82.5,62.5),koord(122.5,-2.5,62.5),koord(122.5,82.5,62.5),
koord(-2.5,82.5,62.5),koord(122.5,82.5,62.5)]).
boyut(t,455,[koord(-2.5,82.5,-2.5),koord(122.5,82.5,-2.5),koord(122.5,-2.5,-2.5),
koord(-2.5,-2.5,-2.5),koord(-2.5,82.5,-2.5)]).
```

### 6.3. KYG'de Bitmiş Parçayı Modelleme

Bitmiş parçayı temsil etmeden önce, parça ile ilgili bazı bilgiler sağlanmış olmalıdır. Yüzeyin yönü bulunduktan sonra, her yüzeyin komşu yüzeyleri bulunur ve veri tabanına yerleştirilip ortak kenarların kontrol edilmesi ile yüzeyler arasındaki açılar hesaplanır. Yüzeyler arasındaki içbükeylik-dışbükeylik ilişkisinin tayin edilmesiyle işlem sürdürülür. Parça son olarak KYG'de temsil edilerek yüzey ve alt özellik temsillerini içeren son model dosyasına yazılır.

### 6.3.1. Yüzeyler arasındaki açılar hesabı

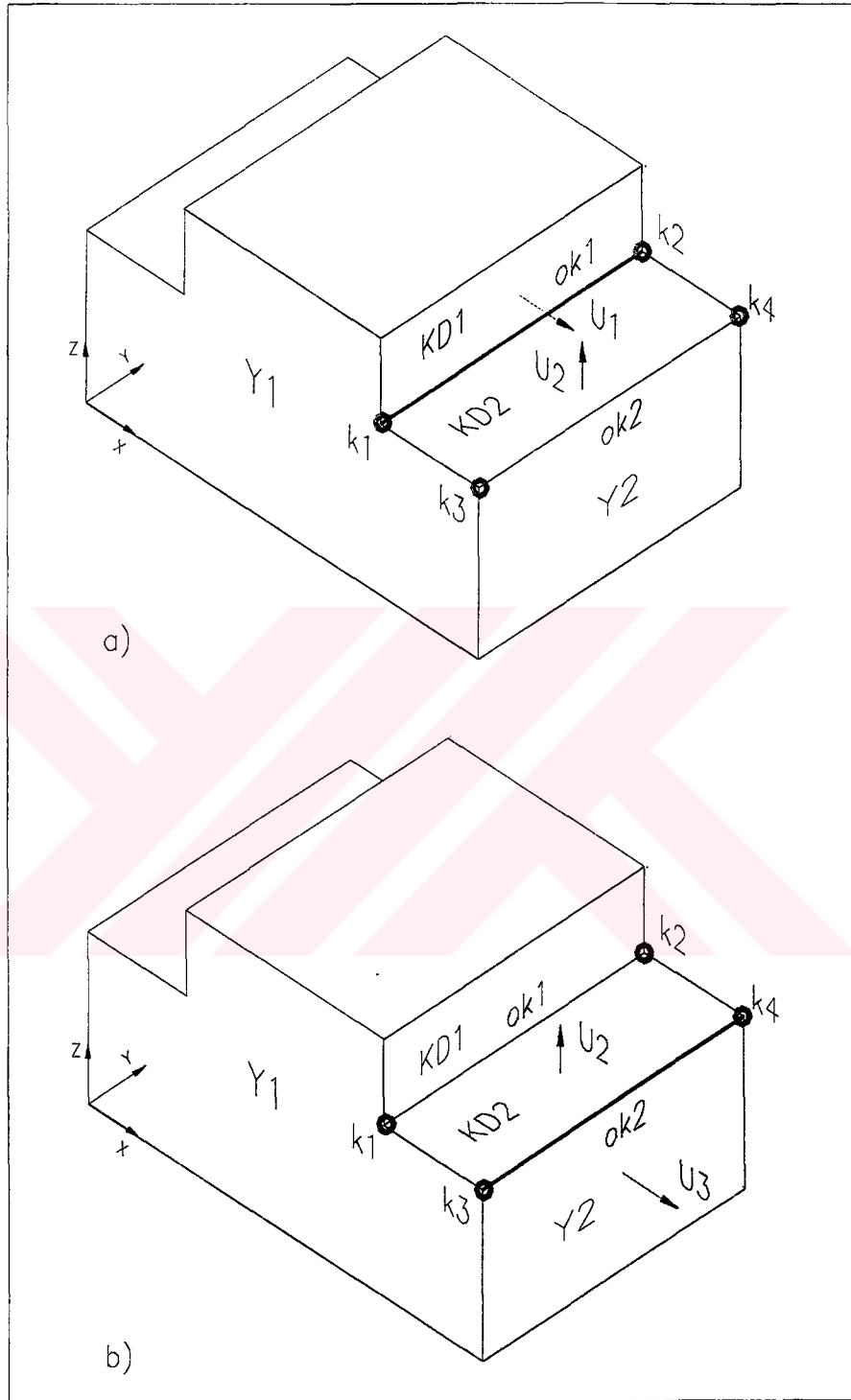
STEP dosyasında her yüzey, birim vektör olan bir yön vektörüyle yönlendirilir. İki yüzey arasındaki açının hesabı, bu yüzeylerin yön vektörleri arasındaki açıya bağlıdır. Yüzeyler arasındaki açılar hesaplanması Ek-C'de açıklanmıştır.

### 6.3.2. İçbükeylik-dışbükeylik ilişkisinin belirlenmesi

İki yüzey arasındaki içbükeylik-dışbükeylik ilişkisinin belirlenmesi, yüzeylerin konumuna ve ortak kenarına bağlıdır. Yüzey listesi STEP dosyasından alınarak her yüzey ayrı ayrı işleme tabi tutulur. Listedeki ilk yüzey alındıktan sonra, bu yüzeyin komşu yüzey listesinden bir komşu yüzeyi alınır ve bu iki yüzeyin ortak kenarı bulunur. Ortak kenar karşılaştırılırken, komşu yüzey, ileriye doğru uzanıyorsa, bu iki yüzey içbükeydir (Şekil 6.12). Diğer taraftan komşu yüzey geriye doğru uzanıyorsa, bu yüzeyler de dışbükey açıyı teşkil eder. Bu prosedür yüzeyin tüm komşu yüzeyleri için, kenar çevrimi listesi boşalincaya kadar tekrarlanır. Daha sonra yüzey listesinden ikinci bir yüzey alınır ve yukarıda izlenen yol, yüzey listesi boşalincaya kadar, her yüzeye uygulanır. Örneğin  $+X$  yönünde  $u1$  vektörüne sahip  $KD1$  yüzeyi,  $KD2$  yüzeyine komşudur ve bu yüzeyler içbükey bir açı oluşturur (Şekil 6.12). Bu ilişkiyi belirlemek için aşağıdaki prosedür uygulanır.  $ok1$  ortak kenarı bulunduktan sonra,  $ok1$ 'in köşelerine ait  $x$  koordinatları bulunur.

Daha sonra  $KD2$  yüzeyinin  $k1$  ve  $k2$  köşeleri dışındaki  $k3$  ve  $k4$  köşelerinin en büyük  $x$  koordinatları belirlenir. Eğer  $KD2$  yüzeyinin en büyük  $x$  koordinatı,  $ok1$ 'in en küçük  $x$  koordinatından büyükse bu iki yüzey içbükeydir.

Benzer şekilde  $Y2$  yüzeyi komşu yüzeyi  $KD2$  ile karşılaştırılabilir (Şekil 6.12). Burada  $Y2$  yüzeyi  $+X$  yönünde  $u3$  yüzey vektörüne sahiptir. Ortak kenar  $ok2$  bulunduktan sonra,  $ok2$ 'ye ait köşelerin  $x$  koordinatları bulunur.  $KD2$  yüzeyinin  $k3$  ve  $k4$  köşeleri dışında kalan köşelerinin en büyük  $x$  koordinatı tespit edilir.  $KD2$  yüzeyinin en büyük  $x$  koordinatı,  $ok2$ 'nin en küçük  $x$  koordinatından büyük olmadığı için,  $KD2$  ve  $Y2$  yüzeyleri dışbükey bir açı oluştururlar.



Şekil 6.12. Yüzeylerin içbükeylik ve dışbükeylik ilişkilerinin belirlenmesi

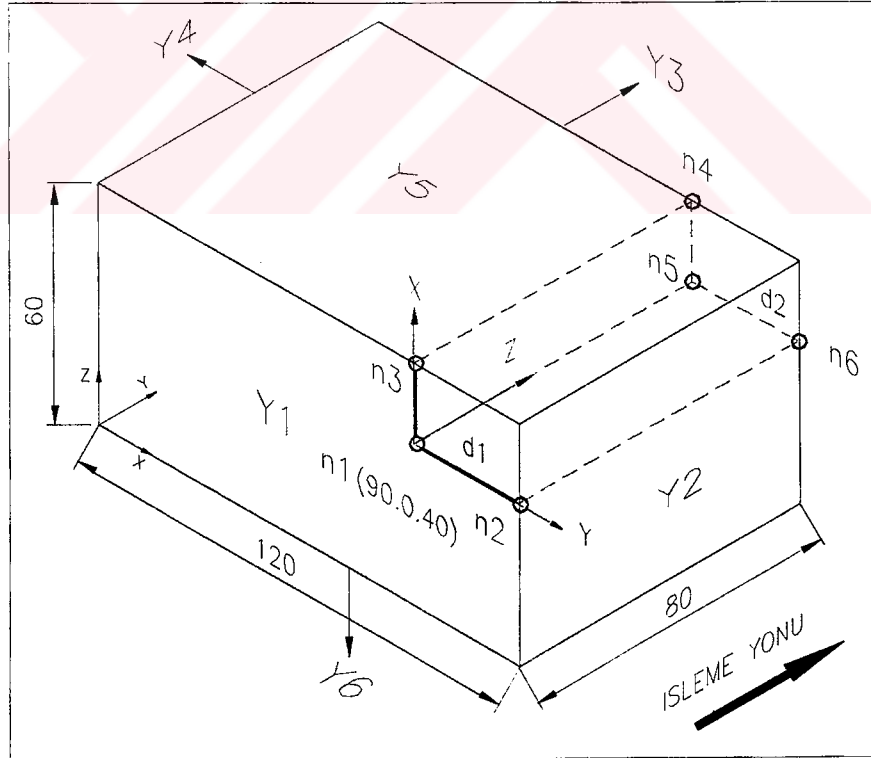
### 6.3.3. Özelliklerin işlenmesi

STEP dosyasında kademeler, kanallar, delikler vs. gibi özellikler süpürme, dönme ve benzer işlemlerle şekillendirilir. Bu sebepten dolayı, özelliğe ait yüzeylerin uyumu, konumu ve bir özelliğin diğer yüzeyler üzerindeki etkisi gibi her özellik için gerekli olan bilgiler elde edilmelidir.

#### 6.3.3.1. Kademe özelliği işleme

Bölüm 5.5'te bahsedildiği gibi, L bir kademe özelliği, süpürme profilidir ve Şekil 6.13'te gösterilen bir kademeyi şekillendirilmek üzere yerleştirilir ve süpürülür.

$n1$ , L profilinin bölgesel koordinat sisteminin orijini.  $n1$  ve  $n3$  arasındaki mesafe ve süpürme uzunluğu, STEP dosyasında bulunmaktadır. Öte yandan  $n2$ ,  $n3$ ,  $n4$ ,  $n5$  ve  $n6$  noktalarının koordinatları ile  $d1$  ve  $d2$  mesafelerinin hesaplanması gerekir.  $d1$  ve  $d2$  mesafeleri, Ek-C'de eşitlik (C.22)'den hesaplanır.

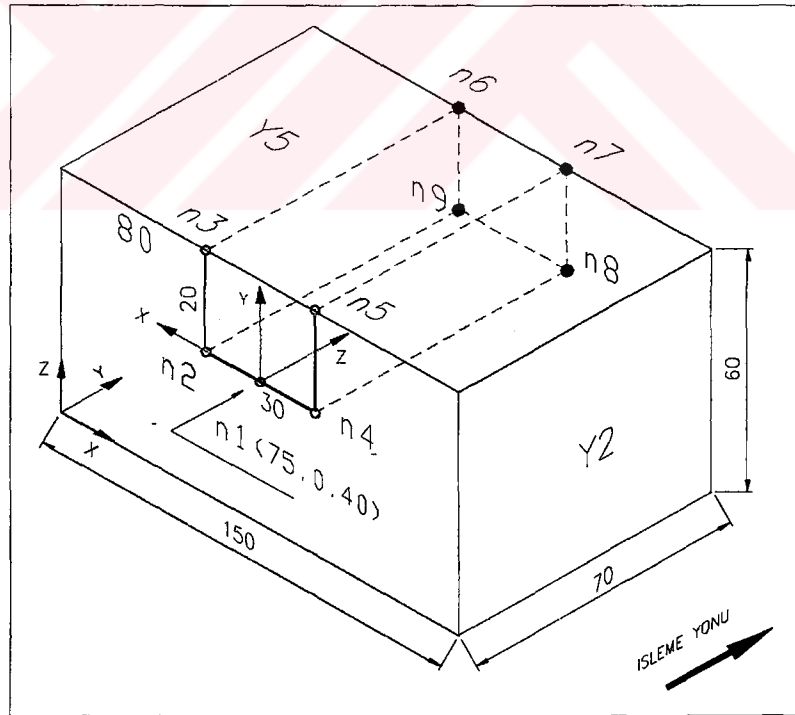


Şekil 6.13. L kademe özelliğinin yerleştirilmesi, uyumu ve işlenmesi

**n2, n3, n4** ve **n6** noktalarının koordinatları (Şekil 6.13), Ek-C'de eşitlik (C.17)'den hesaplanır. Süpürme uzunluğu **80** mm olduğundan, bu durumda **n5**'in koordinatları **(90,80,40)** olur.

### 6.3.3.2. Kanal özelliği işleme

Kademe özelliğine benzer bir kanal özelliği, bir **U** profilin süpürülmesiyle şekillendirilir (Şekil 6.14). Oluşturulacak kanalın genişliği, yüksekliği, süpürme uzunluğu ve **n1** orijininin koordinatları ile bölgesel koordinat sistemi, STEP dosyasında ayrıntılı bir şekilde bulunur. Bununla beraber **n2, n3, n4, n5, n6, n7** ve **n8** köşelerine ait koordinatların hesaplanması gerekir. **n2**'nin koordinatları, **(60,0,40)** ve **n4**'ün koordinatları **(90,0,40)** olarak kolayca bulunabilir. Süpürme uzunluğu **(70)** bilindiğinden, **n9**'un koordinatları **(60,70,40)** ve **n8**'in koordinatları da **(90,70,40)** olur. Diğer taraftan **n3, n5, n6** ve **n7**'nin koordinatları, Ek-C'de açıklanan eşitlik (C.17)'den hesaplanır.



Şekil 6.14. U şekil özelliğinin yerleştirilmesi ve işlenmesi

### 6.3.4. Son modeli dosyaya yazma

Yeni yüzeyler, özelliklere ait köşelerin koordinatları bulunduktan sonra, (kanal ve kademe yüzeyleri) komşu yüzeyleri, yönleri ve kenar çevrimleri, yüzeyler arasındaki içbükeylik-dışbükeylik ilişkileri ve açılarını içerecek şekilde belirlenir. Bir özelliğin yerleşmesi değişirse, komşuluk ilişkisi yeni duruma göre değerlendirilir. Örneğin Şekil 6.15'te gösterildiği gibi, **Y2** yüzeyi, üst sağ kademe özelliği yerleşmeden önce, **Y5** yüzeyine komşuydu; fakat üst kademe özelliğinin yerleşmesi sonucu, bunların komşuluğu sona ermiştir. Bu durumda **Y2** ve **Y5**, yeni kademe yüzeylerine komşudur.

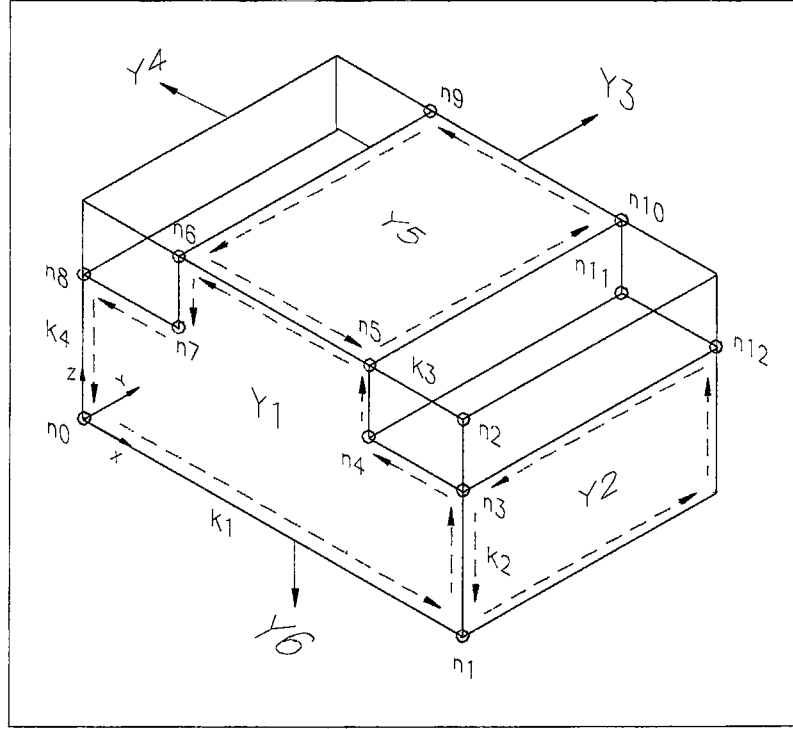
Özelliklerin parçaya girmesiyle (Şekil 6.16) yüzeylerin şekilleri değişeceğinden, her yüzeyin yeni şekli belirlenir ve KYG'de temsil edilir. Sonra son model, kademe ve kanal özelliklerinin yüzeylerini içerecek şekilde dosyaya yazılır.

Kademe ve kanal özelliklerinin köşe noktaları bulunduktan sonra, yüzey listesinden yüzeyler, sırayla alınarak bulunan noktaların yüzey üzerinde olup olmadığı Ek-C'de eşitlik (C.24)'ten kontrol edilir. Özelliklerin yerleşmesiyle üzerinde ortak nokta tespit edilen her yüzey, bu **ortak noktalarla** birlikte veri tabanına yazılır. Böylece ortak noktalara sahip yüzeylerle ortak noktalara sahip olmayan yüzeyler belirlenmiş olur.

Yüzeylere ait kenarların dosyaya yazılması, saat ibresinin ters yönünde olacak şekilde yerine getirilir (Şekil 6.15). Yüzeyler yazılırken yüzeyin ilk kenarı, ilk köşesi yazılmadan önce ele alınır. Böylece yüzeyin ilk köşesinin kaldırılıp kaldırılmadığı kontrol edilmiş olur. Eğer kaldırılmışsa, yeni oluşan köşeye ait nokta, kaldırılan köşenin yerine yazılır. İşlem gören yüzeyin **ortak noktaları** varsa, bunlar bulunmalıdır. Şayet kenar üzerinde herhangi bir ortak nokta varsa, bu ortak nokta, dosyaya yazılarak kenarın ortak noktadan sonra gelen diğer ucu dikkate alınmaz.

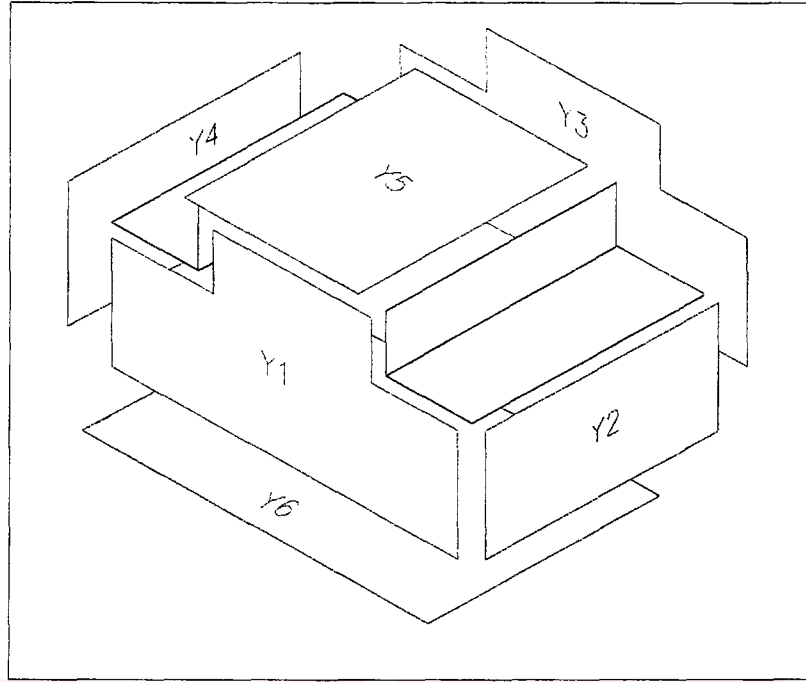
Ortak noktalar işlem gördükten sonra, bu noktalar veri tabanından çıkarılır. Örneğin **Y1** yüzeyi yazılırken (Şekil 6.15) ilk olarak **k1** kenarından başlanır. Program **n0** ve **n1** noktaları ile başlar, bu köşeler kaldırılmadığı için, dosyaya olduğu gibi yazılır. **Y1** yüzeyi ( **n3, n4, n5, n6, n7, n8**) **ortak noktalarına** sahiptir. Bu noktaların **k1** kenarı üzerinde olup olmadığı Ek-C'de açıklanan eşitlik (C.3/C.4/C.5 ve C.6)'dan kontrol edilir.





Şekil 6.15. Son modeli dosyaya yazma

Bu noktalardan herhangi birisi  $k1$ 'in üzerinde olmadığından,  $k2$  kenarına geçilir. Burada  $n3$  noktası kenar üzerindedir. Bu nokta **kademe ortak noktasıdır** ve bu özellik için  $n3$ ,  $n4$  ve  $n5$  olmak üzere üç nokta vardır. İlk önce  $n3$ 'ün koordinatı dosyaya yazılır, sonra  $n4$  ve  $n5$ 'in yazımı bunu takip eder. Kenar üzerinde başka ortak nokta yoksa, bu kenarın işlemine son verilip ardından  $k3$  kenarı ele alınır. Bu kenarın başlangıç noktası olan  $n2$  kaldırıldığından, bunun yerine dosyaya  $n5$  yazılır. Bu kenarın üzerinde ortak herhangi bir noktanın olup olmadığı kontrol edilir. Bu durumda  $n6$  diğer bir kademe ortak noktasıdır. Bu kademe özelliği için toplam kademe ortak noktası üçtür. İlk önce  $n6$ , sonra  $n7$  ve ardından da  $n8$  yazılır.  $k3$  üzerinde başka nokta olmadığından, diğer kenar  $k4$  ele alınır. Son kenar olan  $k4$ 'ün başlangıç noktası kaldırıldığından bunun yerine  $n8$  yazılır. Son olarak  $n0$  noktası yazılarak  $Y1$  yüzeyinin işlemi tamamlanır ve ardından ikinci bir yüzey ele alınır. Benzer şekilde tüm yüzeylerin yeni formları (Şekil 6.16) dosyaya yazılır.



Şekil 6.16. Kademe özellikleri yerleştikten sonra meydana gelen yüzeyler

## 7. UZMAN SİSTEM, REFERANS YÜZEYİNİN SEÇİMİ VE OPERASYON SIRALAMA

### 7.1. Uzman Sistem

US, uzman bir işlem planlayıcının yaptığı gibi, programda yer alan olgulara ve kurallara başvurarak sistem için gerekli sonuçları çıkarır. Sistemin temel yapısı grafik olarak, Şekil 7.1'de olduğu gibi gösterilebilir. Uzman sistem aşağıdaki ana elemanlardan meydana gelir:

- 1- Bilgi tabanı,
- 2- Sonuç çıkarma mekanizması (kontrol mekanizması),
- 3- Kullanıcı arayüzü,
- 4- Bilgi edinme.

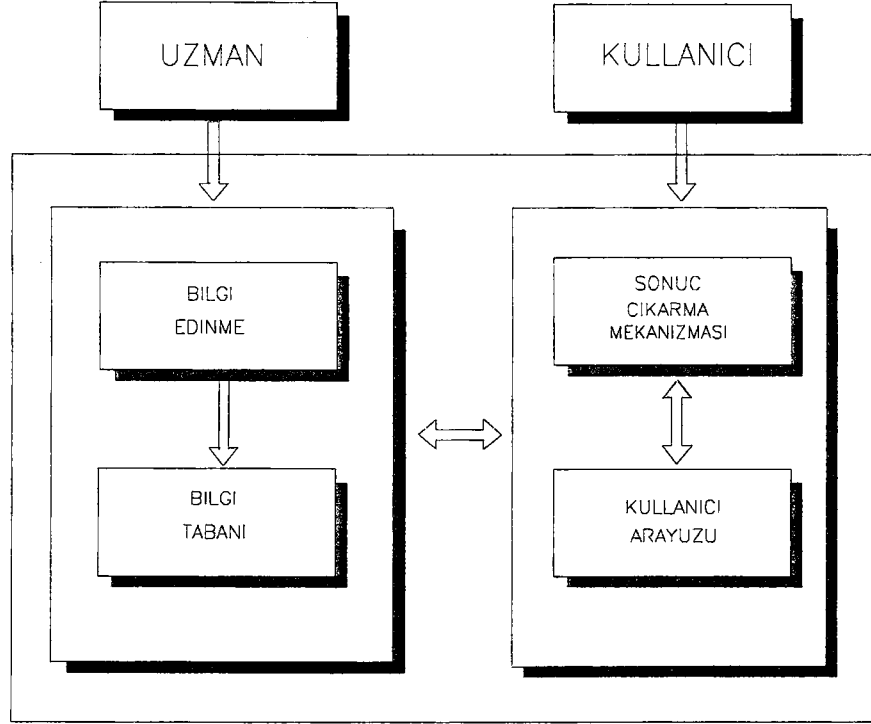
#### 7.1.1. Bilgi tabanı

Bilgi tabanı üretim kuralları, çerçeveler ve bilgi ağları gibi farklı formatlarda temsil edilebilir. Üretim kuralları bilgiyi açıklamanın doğal formu olduğu için, bu çalışmada bilgiyi temsil etmede kullanılmıştır.

Kurallar ve olgulardan oluşan bilgi tabanının amacı, kuralların doğruluğu ve uygulanabilirliğini tayin etmek için, sonuç çıkarma mekanizması tarafından kullanılacak veriler olarak hizmet vermektir. Bir üretim kuralının genel formu aşağıdaki şekildedir.

#### **Eğer (koşul) O halde (eylem ve sonuç)**

Bilgi tabanı, VE/ VEYA grafik tekniği kullanılarak Türkçe terimlerle hazırlanmıştır. Bilgi tabanı, özel alan bilgisi içeren kurallar ve olgulardan meydana gelir. Olgular, objeler arasındaki ilişkileri; kurallar ise, problem alanının kavramları arasındaki mantıksal ilişkileri tanımlar. Kurallar bir veya daha fazla koşul içerebilir.



Şekil 7.1. Uzman sistemin grafik olarak gösterimi

### Eğer

Koşul 1      **ve**

Koşul 2      **ve**

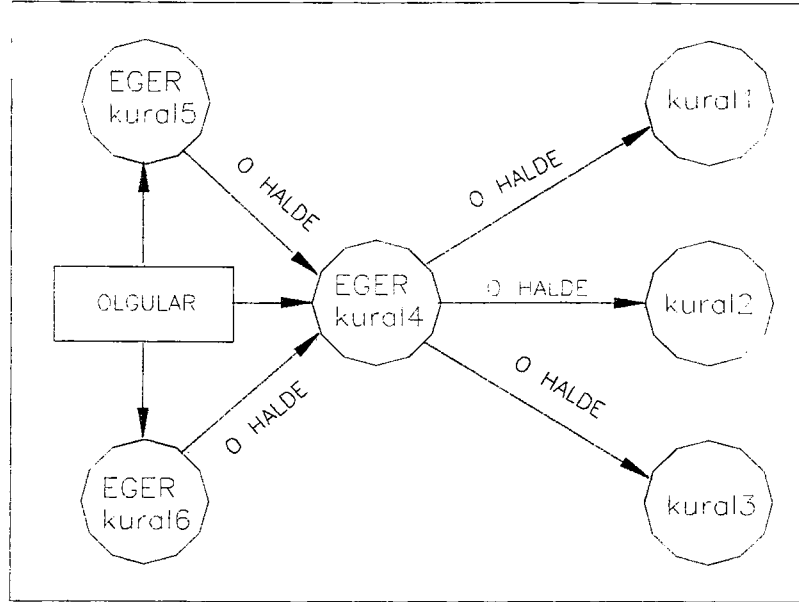
.....

Koşul      **n**

### O halde

Sonuç (eylem)

Bu ifadenin anlamı: **Koşul 1, Koşul 2,.....,Koşul n** doğru ise **Sonuç** doğrudur. **Eğer** kısmındaki her koşul bir **alt hedef** olarak değerlendirilir. Bilgi tabanındaki bir kural, bir problemi çözmeye yeterli olmazsa, diğer kuralları referans alabilir. Şekil 7.2'de görüldüğü gibi kuralların her biri diğerleriyle ilişkilendirilebilir.



Şekil 7.2. Kuralların diğer kuralları referans alması

**Olguları temsil etme:** Daha öncede belirtildiği gibi bir durumu ifade eden olgu, bilgi tabanında aşağıdaki şekilde temsil edilir.

yüzey\_pürüzlülüğü > 0.2 µm.

Burada:

^	^	^	^
A	B	C	D

A: olgunun ilişki ismini,

B: bir karşılaştırma operatörünü,

C: bir sayı değerini,

D: birimi (mikron) ifade etmektedir. Buna benzer çeşitli örnekler aşağıda verilmiştir.

uzunluk > 50 mm.

uzunluk = < 200 mm.

pozitif\_yerleştirme\_toleransı > = 0.075 mm.

negatif\_yerleştirme\_toleransı > = 0.075 mm.

**Kuralları temsil etme:** Bir kural **Eğer** kısmı ve **O halde** kısmından oluşur. Bir kuralda **Eğer** kısmı koşulları içerir, **O halde** kısmı ise sonucu oluşturur. Uygulamada kullanılan kurallara ait bazı örnekler aşağıdaki gibi verilebilir.

EĞER

özellik düzlem\_yüzey VE  
 yüzey\_pürüzlülüğü > 0.2 µm VE  
 sertlik < 56 HRC VE  
 tolerans yüzey\_frezeleme\_için\_geçerli

O HALDE

operasyon yüzey\_frezelemedir

EĞER

uzunluk > 0 mm VE  
 uzunluk = < 50 mm VE  
 pozitif\_yerleştirme\_toleransı > = 0.05 mm VE  
 negatif\_yerleştirme\_toleransı > = 0.05 mm

O HALDE

tolerans yüzey\_frezeleme\_için\_geçerlidir

EĞER

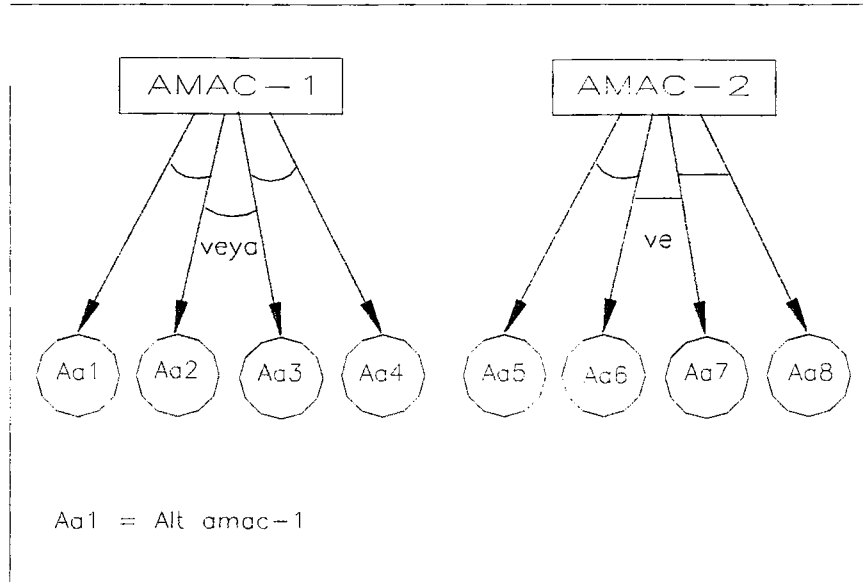
özellik kanal VE  
 sertlik = < 56 HRC VE  
 yüzey\_pürüzlülüğü > = 0.2 µm VE  
 yerleştirme\_pozitif\_toleransı > = 0.05 VE  
 yerleştirme\_negatif\_toleransı > = 0.05 VE  
 genişlik > = 0.7 mm

O HALDE

operasyon kanal\_frezelemedir

### 7.1.2. Kuralları yapılandırmak için VE/ VEYA grafiği

VE/ VEYA grafiği şeklinde hazırlanan kurallar, problemin özelliğine göre alt problemlere ayrılır. Eğer alt problemler birbirinden bağımsız ise, problemi alt problemlere ayırmak bir avantajdır. Bu nedenle, alt problemler birbirlerinden bağımsız olarak çözülebilir. VE/ VEYA grafiğinin hazırlanması ve problemlere uygulanmasına dair geniş bir açıklama Bölüm 3'te açıklanmıştır. Burada sadece araştırma konusu bazı kuralların VE-VEYA grafikleri ve bunların ilişkileri gösterilecektir. VE/ VEYA grafiğine göre yapılan probleme çözüm aranırken problemin VEYA kollarından herhangi birisi, VE kollarının ise hepsi çözülmelidir. Böyle bir VE/ VEYA ilişkisi Şekil 7.3'te gösterilmiştir.



Şekil 7.3. Amaç1 ve amaç2 için VE/ VEYA grafiği

Burada VE bağlantısı iki alt amacı birbirine yatay bir çizgi ile bağlarken, VEYA bağlantısı bir yayla gösterilmiştir. Amaç1'i çözmek için Aa1 veya Aa2 veya Aa3 veya Aa4'ten herhangi birisi çözülür, Amaç 2'yi çözmek için ise Aa6, Aa7, Aa8'in hepsi veya Aa5 çözülür. Örneğin aşağıda verilen kurallar için VE/ VEYA grafiği Şekil 7.4'te gösterildiği gibi hazırlanır.

**EĞER**

özellik düzlem_yüzey	VE -----	A
yüzey_pürüzlülüğü > 0.2 µm	VE -----	B
sertlik < 56 HRC	VE -----	C
tolerans yüzey_frezeleme_için_geçerli	-----	D

**O HALDE**

operasyon yüzey_frezelemedir	-----	Amaç1
------------------------------	-------	-------

**EĞER**

uzunluk > 0 mm	VE -----	E
uzunluk = < 50 mm	VE -----	F
pozitif_yerleştirme_toleransı > = 0.05 mm	VE -----	G
negatif_yerleştirme_toleransı > = 0.05 mm	-----	H

**O HALDE**

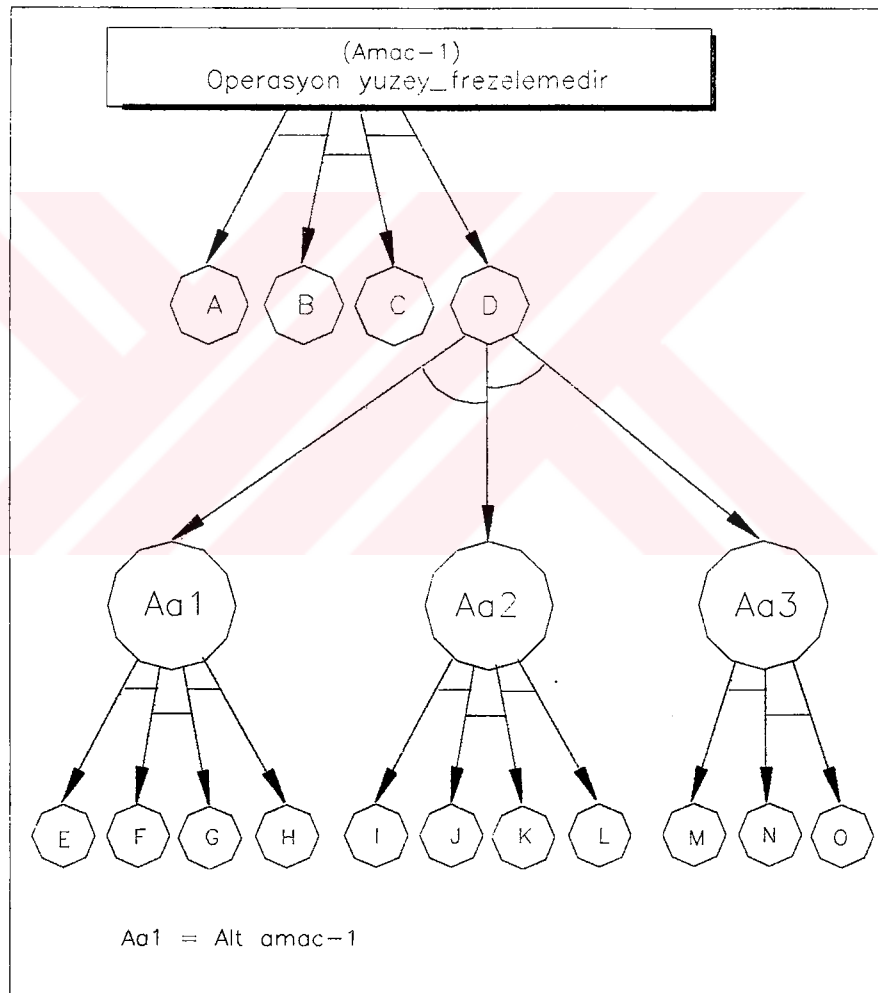
tolerans yüzey_frezeleme_için_geçerlidir	-----	Aa1
--	-------	-----

**EĞER**

uzunluk > 50 mm	VE -----	I
uzunluk = < 200 mm	VE -----	J



pozitif_yerleşirme_toleransı $\geq 0.075$ mm	VE -----	K
negatif_yerleşirme_toleransı $\geq 0.075$ mm	-----	L
O HALDE		
tolerans yüzey_frezeleme_için_geçerlidir	-----	Aa2
EĞER		
uzunluk $> 200$ mm	VE -----	M
pozitif_yerleşirme_toleransı $\geq 0.1$ mm	VE -----	N
negatif_yerleşirme_toleransı $\geq 0.1$ mm	-----	O
O HALDE		
tolerans yüzey_frezeleme_için_geçerlidir	-----	Aa3



Şekil 7.4. Üretim kurallarının VE/VEYA grafiği

### 7.1.3. Bilgi tabanının Prolog cümleciklerine çevrilmesi

Bilgi tabanında yer alan kurallar ve olgular, sistem tarafından yorumlanabilmesi için, Prolog cümlecikleri haline dönüştürülür. Bilgi tabanında yer alan kurallardan birisinin Türkçe ifade yapısı ve Prolog formatı aşağıda verilmiştir.

Kural: Türkçe formatı

Eğer

özellik düzlem\_yüzeysel VE

yüzey\_pürüzlülüğü > 0.2 µm ise VE

sertlik < 56 HRC ise

O halde

İşlem yüzey\_frezelemedir

Kural: Prolog formatı

işlem (yüzey\_frezeleme) :-

özellik(düzlem\_yüzey),

yüzey\_pürüzlülüğü (>,0.2 , "µm"),

sertlik (<, 56, "HRC").

Bilgi tabanında yer alan olgular da, kuralların dönüştürülmesine benzer şekilde, Prolog formata dönüştürülür. Buna ait bir örnek aşağıda verilmiştir.

yüzey\_pürüzlülüğü > 0.2 µm. (Türkçe yapısı)

yüzey\_pürüzlülüğü (>, 0.2, "µm"). (Prolog yapısı)

### 7.1.4. Sonuç çıkarma mekanizması

Veri tabanından kuralları ve olguları okuyabilme yeteneğine sahip olan sonuç çıkarma mekanizması, geriye zincirleme yapabilir ve sonuçlara ulaşmak için derinlik ilk araştırma stratejisi uygular. Bu çalışmada T. Prolog'un kontrol yapısını oluşturan **geriye zincirleme** tekniği kullanılmıştır. Burada sistem bir kuralın başını (sonuç kısmı) ispatlanacak bir amaç olarak ele alır ve bu amaca ulaşmak için kuralın her koşulunu kontrol ederek sonucu sağlamaya çalışır. Örneğin sonuç çıkarma mekanizması "**operasyon kademe\_frezelemedir**" ifadesinin doğru, ya da yanlış bir amaç olup olmadığını ispat etmek isterse, bu sonuca ulaşmak için geriye zincirleme tekniği uygular. Kademe frezeleme için bir kural aşağıdaki şekilde verilmiş olsun:

## EĞER

özellik kademe	VE -----	A
sertlik = < 56 HRC	VE -----	B
yüzey_pürüzlülüğü > = 0.2 µm	VE -----	C
pozitif_yerleştirme_toleransı > = 0.05	VE -----	D
negatif_yerleştirme_toleransı > = 0.05	-----	E

O HALDE  
operasyon kademe\_frezelemedir

Sonuç çıkarma mekanizması geriye zincirleme tekniği uygularken bu kuralın **O halde** kısmını sağlamak için, **Eğer** kısmında yer alan tüm koşulları sağlaması gerekir. Sonuç çıkarma mekanizması bu koşulların tamamını sağlarsa, "operasyon kademe\_frezelemedir" sonucuna vararak işleme son verir. Eğer koşulların birisini sağlayamazsa, bu durumda geri izleme yaparak aynı sonuca sahip başka bir kuralın olup olmadığını araştırır. Şayet bir kural bulursa, aynı prosedürü bu kurala da uygular. Öte yandan sistem ileriye doğru zincirleme tekniği kullanırsa, o zaman kuralın **Eğer** kısmında yer alan A, B, C, D ve E koşullarıyla başlar ve sırayla bunları sağlayan olguları araştırır. Bu koşulların tamamını sağladıktan sonra sonuç kısmı da sağlanmış olur.

Geriye ve ileriye zincirlemenin her ikisinde de sorgulama işlemi aşağıdaki prensiplere göre yerine getirilir. İspat edilmesi gereken hedef, daima bir **soru** olduğundan bu soruya bir **cevap** bulmak için şu işlemlerden birisi kullanılır.

- Sistem tarafından yönlendirilen **soru** bilgi tabanında bir olgu olarak bulunursa, o halde **cevap "soru doğrudur"** ifadesi olur.

- Sistem tarafından yönlendirilen **soruya** programda uygulanabilir bir kural varsa, o zaman problemin tipine göre, ileriye veya geriye zincirleme uygulayarak **cevabı** bulmak için bu kural sağlanmaya çalışılır.

- İşlem gören kuraldaki tüm VE koşulları sağlanmalıdır. Eğer bunlardan birisi sağlanamazsa **cevap "soru yanlıştır"** şeklindedir.

- İşlem gören kuralda VEYA alternatifleri varsa ve bunlardan herhangi birisi sağlanırsa, bu durumda **cevap** sağlanan VEYA alternatifidir. Tüm VEYA alternatifleri yanlış ise, **cevap** yanlıştır.

-Yukarıdaki işlemlerden herhangi birisiyle bir çözüm sağlanamazsa, o zaman bu problemin çözülemez olduğu kabul edilir.

### 7.1.5. Kullanıcı arayüzü

Kullanıcı arayüzü, ana menü yardımıyla sistem ve kullanıcı arasındaki iletişimi sağlayan sistemin bir parçasıdır. Kullanıcı arayüzü, stok, referans yüzeyi, takım tezgahı ve kesici seçme modülleri ile bağlama işlemi ve araparça modelleri oluşturma gibi sistemi meydana getiren diğer modüllerle iletişimi sağlar. Kullanıcı arayüzü sistemi sorgulama, veri tabanını araştırma, dosyaları muhafaza etme, çıktıları yazdırma ve işlem planlama kartlarını gösterme gibi işlemleri yerine getirir.

### 7.1.6. Bilgi edinme ve sürdürme

Deneyimli bir işlem planlamacıdan elde edilen bilgiler, bir editör yardımıyla ilgili program modüllerinde depolanıp uygulanacağı ortama elverişli hale getirmek için modifiye edilir.

## 7.2. Referans Yüzeyi Seçimi

Parçanın referans yüzeyi, bir özellik işlenirken önemli rol oynar. Genellikle işleme ve bağlama yüzeylerinin belirlenmesi referans yüzeyine göre yapılır. Bunun için talaş kaldırma ve bağlama işlemleri yapılmadan önce referans yüzeyinin seçilmesi gerekir.

Referans yüzeyi seçimi için ilgili kurallar, bilgi tabanında üretim kuralları formundadır. Sonuç çıkarma mekanizması her yüzeyi kontrol ederek bunlarla ilgili kuralları uygular. Geriye zincirleme tekniği ve derinlik ilk araştırma stratejisi uygulamak suretiyle, hangi yüzeyin referans yüzeyi olacağına karar verir. Referans yüzeyi seçimi ile ilgili bazı örnek kurallar aşağıda verilmiştir.

EĞER  
 yüzey düzlem\_yüzey VE  
 yüzey dış\_yüzey  
 O HALDE  
 yüzey işleme\_referans\_yüzeyidir

EĞER  
 yüzey düzlem\_yüzey VE  
 yüzey en\_büyük  
 O HALDE  
 yüzey işleme\_referans\_yüzeyidir

EĞER

yüzey referans\_yüzeyi\_olarak\_verilmiş  
 O HALDE  
 yüzey işleme\_referans\_yüzeyidir

Referans yüzeyi seçiminde kullanılan her kural, öncelik seviye değerine sahip bir kuralla eşleştirilmiştir. Örnek olarak yukarıda verilen kurallarla eşleşen öncelik seviye değerli kurallar aşağıda verilmiştir.

EĞER  
 yüzey düzlem\_yüzey VE  
 yüzey dış\_yüzey  
 O HALDE  
 referans\_yüzeyi\_olma\_önceliği 90

EĞER  
 yüzey düzlem\_yüzey VE  
 yüzey en\_büyük  
 O HALDE  
 referans\_yüzeyi\_olma\_önceliği 80

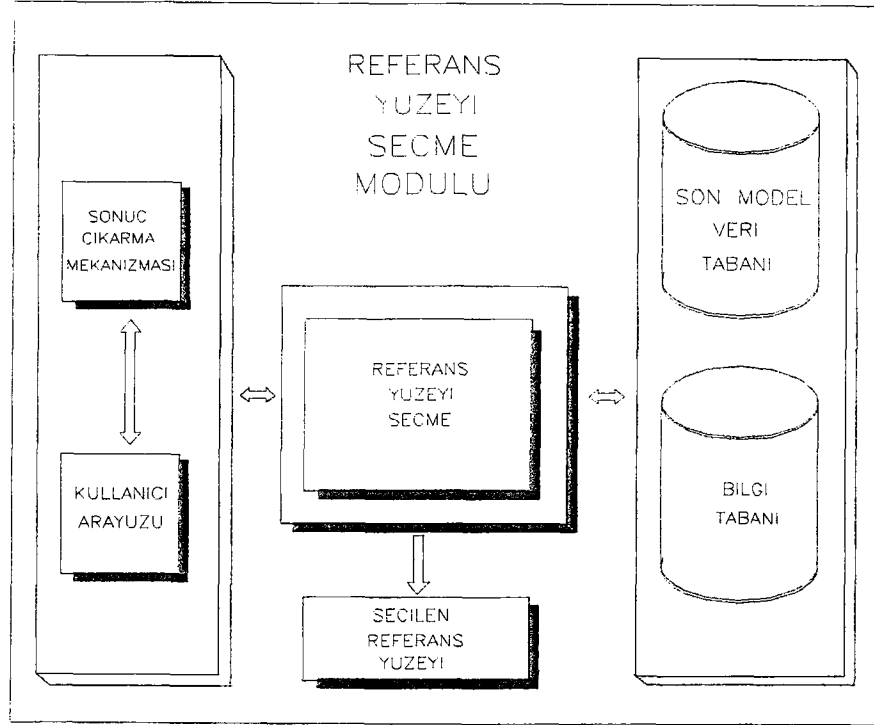
EĞER  
 yüzey referans\_yüzeyi\_olarak\_verilmiş  
 O HALDE  
 referans\_yüzeyi\_olma\_önceliği 100

Sistem referans yüzeyini belirlemek üzere işleme başladığında, veri tabanından özelliklerin listesini alır ve ilk özellik için öncelik seviye değerine sahip kuralları sağlamaya çalışır (Şekil 7.5).

Sağlanan kuralların öncelik seviye değerleri bir listeye yazılarak toplanır. Bu prosedür her özelliğe uygulandıktan sonra, özelliklerin elde edilen toplam öncelik seviye değerleri karşılaştırılır. En büyük toplam öncelik seviye değerine sahip özellik, referans yüzeyi olarak seçilerek veri tabanına yazılır. Referans yüzeyi seçme işleminin ana hatları Şekil 7.6'da gösterilmiştir.

### 7.3. Operasyonların Sıralanması

Operasyonların sıralanması, işlem planlamada önemli bir rol oynar. Operasyon sıralama, tasarım özelliklerine göre bitmiş parçayı üretmek için, ham malzemeye uygulanacak imalat işlemlerinin sırasını tayin eder [36]. Özellikleri işlemek için gerekli diğer işlem planlama faaliyetleri, operasyonların sıralanmasına göre yerine getirilir. Operasyonların sıralanması, parçayı imal etmek için uygulanacak işlemleri düzenleme yanında, bir işlem planını da elde etmeyi sağlar.

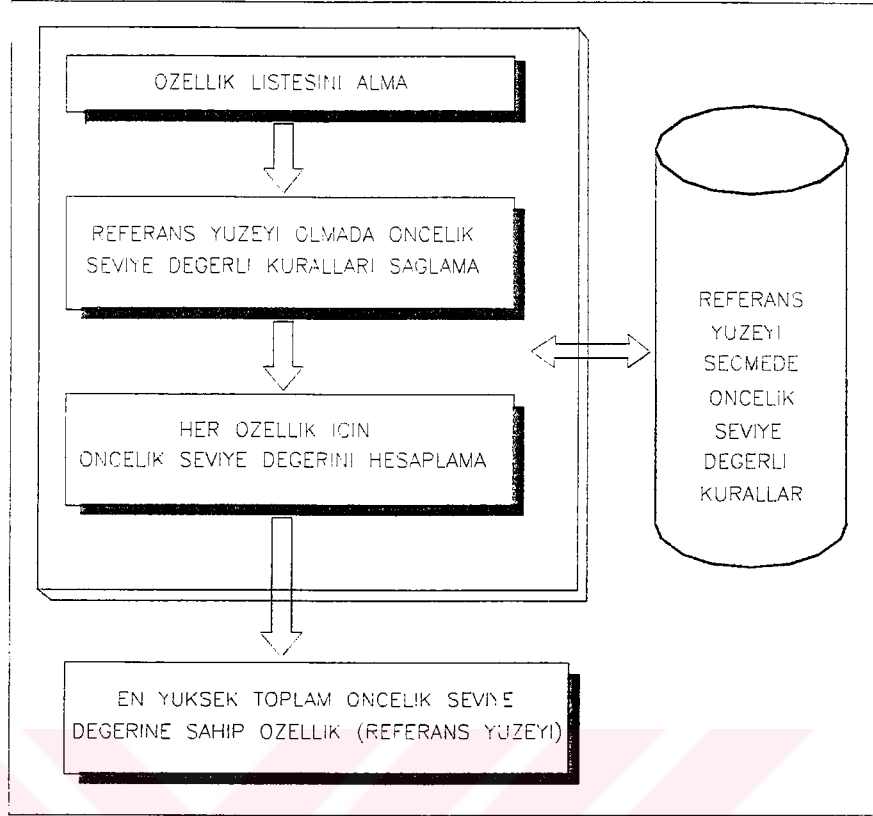


Şekil 7.5. Referans yüzeyi seçme modülü

Sistem tarafından referans yüzeyi seçildikten sonra, operasyonların sırası belirlenir. Daha sonra işlem planlamada takip eden görevler yerine getirilir (Şekil 7.7). İşlemlerin sıralanmasında kullanılan bilgi tabanı, referans yüzeyi seçmede kullanılan kurallara benzer tarzda **sıralama kuralları** ve özellikler arasındaki öncelikli ilişkiyi gösteren, öncelik seviye değerli kurallardan (**öncelikli kurallar**) meydana gelir.

Bilgi tabanındaki sıralama kuralları, özellikler arasındaki öncelik ilişkisini tayin eder. Yani bu kurallar özelliklerin işleme önceliğini belirler. Kurallara göre her özellik bir diğer özellikten önce işlenmelidir. Bazen bir özellik diğer özelliklerle birlikte işlenebilir. Örneğin genelde bir düzlem yüzey onun içindeki bir delikten önce işlenir. Benzer şekilde bir T-kanal başlangıçta bir dikdörtgen kanal açıldıktan sonra üretilebilir. Kurallar aşağıdaki faktörlere bağlı olarak konstrükte edilir.

1. Geometri,
2. Referans yüzeyi,
3. Tolerans,
4. Yüzeyin büyüklüğü,
5. Yüzey pürüzlülüğü.



Şekil 7.6. Referans yüzeyi seçme işlemi

Operasyon sıralama esas itibariyle özelliklerin sıralamasına dayanır. Parçanın modeli, sistem tarafından incelenerek, işlenecek özellikler belirlenir. Sonuç çıkarma mekanizması, geriye zincirleme tekniği ve derinlik ilk araştırma stratejisi kullanarak her özellik için ilgili **öncelik seviye değerli kuralları** sağlamaya çalışır. Sağlanan kuralların öncelik seviye değerleri, bir listeye yazılarak toplanır. Bu prosedür her özelliğe uygulandıktan sonra, özelliklerin elde edilen toplam öncelik seviye değerleri karşılaştırılır. En büyük toplam öncelik seviye değerine sahip özellik, en başta olacak şekilde, özelliklerin toplam öncelik seviye değerleri, varlık numaralarıyla birlikte, bir listeye yerleştirilir ve veri tabanına yazılır. Böylece işlem planlama sistemi, daha sonraki görevlere giriş olarak bunları dikkate alır. **Sıralama kurallarının** bazı örnekleri aşağıda verilmiştir.



EĞER

özellik düzlem\_yüzey VE  
özellik referans\_yüzeyi VE  
özellik dış\_yüzey

O HALDE

özellik ilk\_önce\_işlenecektir

EĞER

özellik düzlem\_yüzey VE  
özellik dış\_yüzey

O HALDE

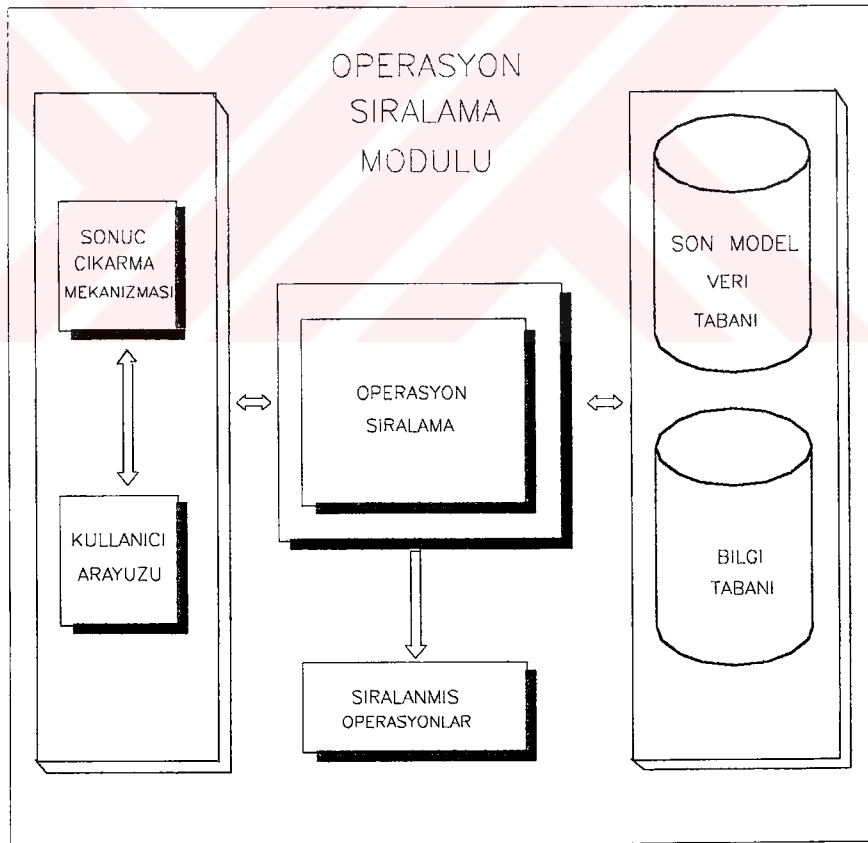
özellik ilk\_önce\_işlenecektir

EĞER

özellik düzlem\_yüzey VE  
tolerans en\_yüksek

O HALDE

özellik ilk\_önce\_işlenecektir



Şekil 7.7. Operasyonların sıralanması

Karşılık gelen **öncelikli kurallar**'a ait bazı örnekler şu şekilde verilebilir.

EĞER

özellik düzlem\_yüzey VE  
özellik referans\_yüzeyi VE  
özellik dış\_yüzey

O HALDE

operasyon\_sıralaması\_önceliği 90

EĞER

özellik düzlem\_yüzey VE  
özellik dış\_yüzey

O HALDE

operasyon\_sıralaması\_önceliği 85

EĞER

özellik düzlem\_yüzey VE  
tolerans en\_yüksek

O HALDE

operasyon\_sıralaması\_önceliği 80

Örneğin Bölüm-6 Şekil 6.2'de verilen parça için özelliklerin sıralaması aşağıdaki gibi belirlenmiştir.

özellik\_listesi([top(455,175),top(435,165),top(365,90),top(360,90),  
top(355,90),top(350,90),top(450,55),top(445,55),top(440,35)]).

Sonra, istenen geometriyi imal etmek için, sıralanan özelliklere göre gereken imalat işlemleri, **operasyon seçme** modülü tarafından yerine getirilir. Bağlama işlemi, kesiciler ve takım tezgahlarının seçimi ve işleme parametrelerinin hesaplanması tamamlandıktan sonra, çıktı, kullanıcı arayüzü ve sonuç çıkarma mekanizması yardımı ile, sıralanmış işlemleri içeren bir işlem planlama kartı haline dönüştürülür.

## 8. OPERASYON, TAKIM VE TAKIM TEZGAHI SEÇİMİ

### 8.1. İmalat Operasyonu Seçimi

Özellikler sıralandıktan sonra, her özelliğe uygulanacak operasyon belirlenir. Operasyon seçimi, kaba malzemedan talaş kaldırmak suretiyle, parçanın özelliklerini istenen şekle getirmek için, uygun işlemleri seçme görevidir. Kaba malzemenin durumunu değiştiren her talaş kaldırma operasyonu, özel bir metodla yerine getirilir [107].

Her yüzey/özellik için operasyonlar, **kullanıcı arayüzü** ve **sonuç çıkarma mekanizması** yardımıyla, bilgi tabanındaki ilgili kuralları ve olguları kullanarak seçilir (Şekil 8.1). İşlem seçimi, taslak parçadan bitmiş parça şekli oluşuncaya kadar tekrarlanır ve sürdürülür. Sonuç çıkarma mekanizması istenen geometriyi üretmek için, parçanın özelliklerine uygun kuralları icra eder ve gerekli işleme operasyonlarını belirler. Özellik için ilgili kuralların uygulanmasında gerekli teknolojik bilgi veri tabanından alınır. Eğer yeterli bilgi varsa, sonuç çıkarma mekanizması, bir özelliği işlemek için gereken çözümü bulabilir.

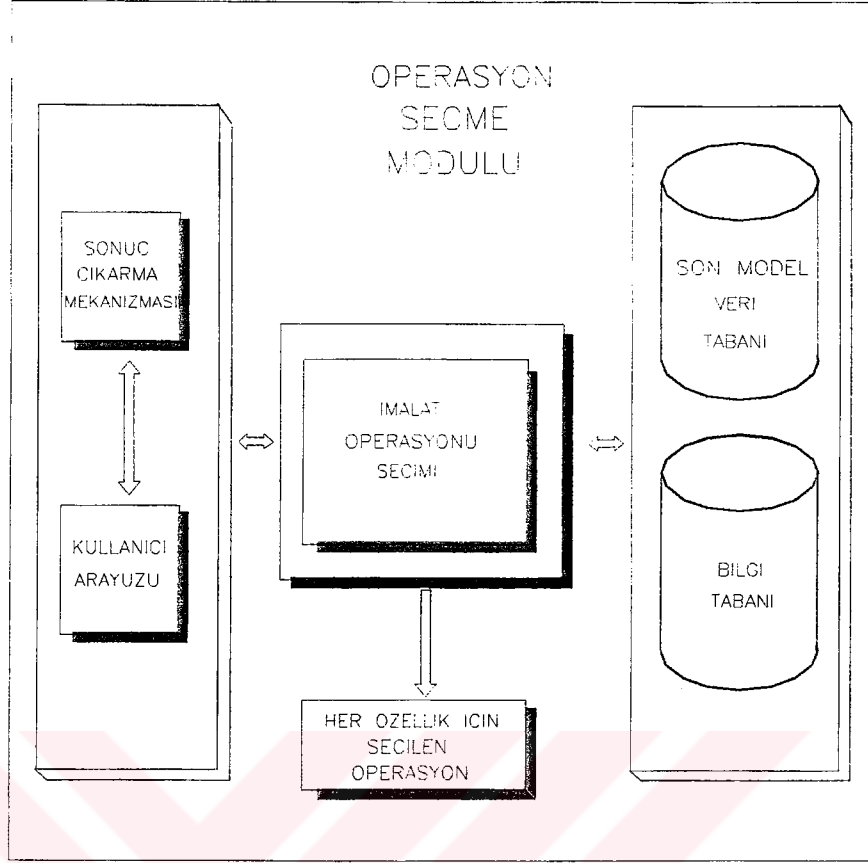
#### 8.1.1. Operasyonları seçmede kullanılan bilgi tabanı

Bir özelliğe uygulanacak operasyonu belirleyebilmek için, imalat işlemlerinin kapasiteleri bilinmelidir [8]. Operasyonları seçmede kullanılan bilgi tabanı, her işlemin işlem kapasitesine göre hazırlanmıştır. Bu işlemle ilgili bazı kısıtlar aşağıda sıralanmıştır:

- Bir operasyonla üretilebilecek şekil ve ölçü,
- Farklı işlemlerle sağlanabilecek geometrik toleranslar,
- Erişilebilir yüzey kalitesi,
- Geometrik ve teknolojik kısıtlar,
- Sertlik.

Talaş kaldırma işlemleri, iş parçasından malzeme kaldırarak yeni yüzeyler oluşturmayı amaçlar. Elde edilen yüzeylerin karakteristikleri, uygulanan işlemin kapasitesine bağlıdır ve her operasyon belirli bir şekli üretebilir.

İşlemin **şekil** oluşturma kapasitesi, talaş kaldırma işlemini gerçekleştiren takımın geometrisiyle tayin edilir. Örneğin yüzey frezeleme, bir düzlem yüzeyi,



Şekil 8.1. İmalat operasyonları seçme modülü

helisel matkapla delme ise, silindirik bir deliği üretebilir. Yüzeyin/özelliğın ölçüsü ve yüzey kalitesi, uygulanan operasyonlara, kullanılan takıma ve takım tezgahının iş yapma hassasiyetine bağlıdır. Örneğın belirli bir matkapla ne sonsuz genişlik ve derinlikte bir delik, ne de en küçük ölçüsünden daha küçük bir delik delinemez. Deliğın derinliğı, kullanılan takımın çapıyla ilgilidir. Helisel bir matkapla, çapının 10 katından daha fazla derinlikte bir delik delinemez. Eğer derinlik/çap oranı 10/1'den daha büyükse, delik, namlu delmede uygulanan derin delikleri delme işlemleriyle, yerine getirilmelidir.

İşlenen yüzeyden/özelliğten beklenen **tolerans** ve **yüzey kalitesi**, seçilen operasyonla yakından ilgilidir. Her işlem, belirli bir yüzey kalitesi ve tolerans değeri elde edebilir. Gerçek bir üretim ortamında, bir özellik için seçilen operasyonun istenen toleransı sağlayıp sağlayamayacağını dikkate almak gerekir. Bunun için toleranslama kavramı önemlidir. Seçilen operasyonlar, istenen yüzey kalitesini ve hassasiyeti yerine getirme kapasitesine sahip

olmalıdır. Örneğin helisel matkap, delme burcu kullanılarak belirli doğruluk ve yüzey pürüzlülüğünde bir deliği imal edebilir. Kullanılan matkabın çapı 6.3 ile 19 mm arasında ise, çap toleransı  $\pm 0.076$  mm olarak alınabilir. Çapı 19 ile 75 mm arasında olan matkaplarda, delme burcu ve punta matkabı kullanılmadığı takdirde, çap toleransı  $\pm 0.2$  mm'dir. Yüzey frezeleme operasyonları ile yüzey pürüzlülüğü değeri  $0.2 \mu\text{m}$ 'den daha büyük yüzeyler elde edilebilirken, helisel matkapla delmede, yüzey pürüzlülüğü değeri  $0.8 \mu\text{m}$ 'ye kadar olan delikler elde edilebilir. Öte yandan düzlem yüzey taşlama ve delik honlama işlemi ile,  $0.025 \mu\text{m}$ 'ye kadar yüzey pürüzlülüğü değeri elde edilebilir [7].

Uygulanan operasyonlar, malzemelerin sertlik değerlerine bağlı olarak çeşitli kesici takımları gerektirdiği için, işlem seçme kuralları hazırlanırken **sertlik** kısıtı da dikkate alınmalıdır. Yüksek Hız Çeliği (HSS) veya değiştirilebilir karbid uçlu takım kullanıldığı zaman, işlenecek malzemenin sertliği 22 HRC'den az olmalıdır. Eğer parçanın sertliği 22 HRC'den büyükse, karbid takımlar kullanılmalıdır. Operasyon seçimi için kuralların hazırlanmasında, yukarıda açıklanan kısıtlardan başka, **geometrik** ve **teknolojik** kısıtlar da dikkate alınmalıdır.

**Geometrik kısıtlar**, özelliklerin geometrik ilişkileri ile belirlenir. Örneğin delik büyültme ve raybalama işlemleri, bir ön delik olmadıkça bir parçaya doğrudan uygulanamaz. Benzer şekilde, T-kanalı işlemeden önce, takım sapının girebileceği genişlikte bir kanal açılmış olmalıdır. Delik düz uçlu olarak işlenecekse, işlem sonunda delik tabanı bir parmak freze ile işlenmelidir.

**Teknolojik kısıtlar**, tezgah-takım seçme modülü ile değerlendirilen güç tüketimi, talaş kaldırma miktarı, kesme hızı vs. gibi parametrelerle belirlenen kısıtlardır. Operasyon seçimi için hazırlanan kurallar, bir çözümü bulurken diğer kuralları referans alabilir. Başka bir ifadeyle kurallar arasındaki dallanma, arzu edilen sıklıkta yapılabilir. Bilgi tabanında aynı veya benzer operasyonlar için kullanılan kurallar, kullanıcı tarafından kolayca bulunabilmesi için, özel bir başlık altında toplanır. Örneğin **cep** ve **kademe** frezeleme için kullanılan kurallar aşağıdaki şekilde verilebilir.

#### /\* Cep frezeleme \*/

EĞER

özellik cep VE  
 sertlik = < 56 HRC VE  
 yüzey\_pürüzlülüğü > =  $0.2 \mu\text{m}$  VE  
 pozitif\_yerleştirme\_toleransı > = 0.05 VE  
 negatif\_yerleştirme\_toleransı > = 0.05 VE  
 derinlik = < 180 mm

O HALDE  
işlem cep\_frezelemedir

**/\* Kademe frezeleme \*/**

EĞER  
özellik kademe VE  
sertlik = < 56 HRC VE  
yüzey\_pürüzlülüğü > = 0.2 µm VE  
pozitif\_yerleştirme\_toleransı > = 0.05 VE  
negatif\_yerleştirme\_toleransı > = 0.05  
O HALDE  
işlem kademe\_frezelemedir

İmalat operasyonuna karar vermek üzere hazırlanan kural, bu operasyonun gerektirdiği tüm özellikleri kapsamalıdır. Kuralın yapısındaki doğru ayrıntıların zenginliği, direkt olarak bulunan sonucun hassasiyetine ve doğruluğuna etki eder. İmalat operasyonlarının seçiminde kullanılan kural örnekleri aşağıda verilmiştir.

EĞER  
özellik konik\_uçlu\_kör\_delik VE  
sertlik = < 56 HRC VE  
ön\_delik açılmış VE  
yüzey\_pürüzlülüğü > = 0.4 µm VE  
derinlik > = 3 \* çap VE  
derinlik = < 55 \* çap VE  
çap > = 1 mm VE  
çap = < 75 mm VE  
burç kullanılıyor VE  
çap\_toleransı raybalamada\_geçerli VE  
yerleştirme\_toleransı raybalamada\_geçerli VE  
doğrusallık\_toleransı > = 0.3 mm VE  
yuvarlaklık\_toleransı > = 0.01 mm  
O HALDE  
işlem raybalamadır

EĞER  
özellik yarım\_ay\_kama\_kanalı VE  
sertlik = < 50 HRC VE  
yüzey\_pürüzlülüğü > = 0.2 µm VE  
pozitif\_yerleştirme\_toleransı > = 0.05 VE  
negatif\_yerleştirme\_toleransı > = 0.05 VE  
genişlik > = 1 mm VE  
genişlik = < 10 mm  
O HALDE  
işlem yarım\_ay\_kama\_kanalı\_frezelemedir

### 8.1.1.1. Frezeleme operasyonları

**Yüzey frezeleme:** Yüzey pürüzlülüğü değeri, 0.2  $\mu$ m'den daha büyük yüzeyleri işlemek için uygulanır. Frezelenen yüzeyin ölçüsüne bağlı olarak 0.075 ile 0.2 mm arasında tolerans değerleri elde edilebilir. Yüzey frezeleme için modellenen kurallardan bazıları aşağıda verilmiştir.

EĞER

özellik düzlem\_yüzey VE  
 yüzey\_pürüzlülüğü > 0.2  $\mu$ m VE  
 sertlik < 56 HRC VE  
 tolerans yüzey\_frezeleme\_için\_geçerli

O HALDE

operasyon yüzey\_frezelemedir

EĞER

uzunluk > 0 mm VE  
 uzunluk = < 50 mm VE  
 pozitif\_yerleştirme\_toleransı > = 0.05 mm VE  
 negatif\_yerleştirme\_toleransı > = 0.05 mm

O HALDE

tolerans yüzey\_frezeleme\_için\_geçerli

**Kanal frezeleme:** Düz, kırılmaç kuyruğu, T-kanal, V-kanal gibi kanal frezeleme operasyonlarından oluşur. T-kanallar iki kademedeki üretilir. İlk olarak parmak, ya da kanal freze takımıyla bir ön kanal açılır; daha sonra T-freze ile işlem tamamlanır. Kanal frezeleme uygulamalarına ait bazı kural örnekleri aşağıda verilmiştir.

#### /\* Kırılmaç kuyruğu kanal frezeleme \*/

EĞER

özellik kırılmaç\_kuyruğu\_kanal VE  
 sertlik = < 50 HRC VE  
 yüzey\_pürüzlülüğü > = 0.2  $\mu$ m VE  
 pozitif\_yerleştirme\_toleransı > = 0.05 VE  
 negatif\_yerleştirme\_toleransı > = 0.05 VE  
 ön\_kanal açılmış

O HALDE

operasyon kırılmaç\_kuyruğu\_kanal\_frezelemedir

#### /\* T- kanal frezeleme \*/

EĞER

özellik t\_kanal VE



sertlik = < 50 HRC VE  
 yüzey\_pürüzlülüğü > = 0.2 µm VE  
 pozitif\_yerleştirme\_toleransı > = 0.05 VE  
 negatif\_yerleştirme\_toleransı > = 0.05 VE  
 genişlik > = 11 mm VE  
 genişlik = < 60 mm VE  
 kalınlık > = 3.5 VE  
 kalınlık = < 28 mm VE  
 ön\_kanal açılmış  
 O HALDE  
 operasyon t\_kanal\_frezelemedir

### 8.1.1.2. Delme operasyonları

Dairesel delikleri üretmek ve bu delikler üzerinde çeşitli işlemleri uygulamak üzere, delme operasyonu kullanılır (Şekil 8.2). Dairesel kesitli delik elde etmede, en etkili ve ekonomik metod delme işlemidir. Delme, diğer imalat işlemlerine göre hem çok basittir, hem de kabul edilebilir bir hassasiyeti sağlar. Delik hızlı bir şekilde delinirken takım da kolaylıkla kontrol edilebilir. Doğrudan matkapla delinerek elde edilen yüzey, kaba olduğundan, ölçü hassasiyeti ve yerleştirme toleransı iyi değildir. Bu yüzden, delme burçları kullanarak ölçü hassasiyeti önemli ölçüde arttırılır. Kalıplardaki delme burçları, üniform bir şekilde matkabın merkezlemesini sağlar. Böylece, yerleştirme hassasiyeti, doğrusalık ve dairesellik daha iyi sağlanmış olur. Çapları 1 ile 76 mm arasında olan delikler, standart helisel matkaplarla delinir.

Matkapla bir delik, boydan boya veya kör delik olarak delinir. Elde edilecek deliğin kalitesi punta matkabı, ya da delme burcunun kullanılıp kullanılmamasına karar vermede etkili olur. Delme burçlarının kullanılmasıyla, daha dar tolerans değerleri elde edilir. Helisel matkapla delme işlemlerinde kullanılan bazı kural örnekleri aşağıda verilmiştir.

EĞER  
 özellik boydan\_boya\_delik VE  
 sertlik = < 22 HRC VE  
 yüzey\_pürüzlülüğü > = 0.80 µm VE  
 derinlik = < 10 \* çap VE  
 çap > = 1 mm VE  
 çap = < 75 mm VE  
 çap\_toleransı helisel\_matkapla\_delmede\_geçerli VE  
 yerleştirme\_toleransı helisel\_matkapla\_delmede\_geçerli VE  
 burç kullanılıyor VE  
 doğrusalık\_toleransı > = 0.6 mm VE  
 yuvarlaklık\_toleransı > = 0.05 mm

O HALDE

operasyon helisel\_maktkapla\_delmedir

EĞER

çap > 6.3 mm VE

çap = < 19 mm VE

pozitif\_çap\_toleransı > = 0.076 mm VE

negatif\_çap\_toleransı > = 0.076 mm

O HALDE

çap\_toleransı helisel\_maktkapla\_delmede\_geçerlidir

**Sert madenle delme:** Sert madenle delme operasyonu üç kategoride sınıflandırılır. Bunlar:

- Burç kullanarak yapılan delme,
- Punta matkabı kullanarak yapılan delme,
- Burç ve punta matkabı kullanmadan yapılan delme.

Bu işleme ait bir kural örneği aşağıdaki gibi verilebilir.

EĞER

özellik boydan\_boya\_delik VE

sertlik > 22 HRC VE

sertlik = < 56 HRC VE

yüzey\_pürüzlülüğü > = 0.80 µm VE

derinlik = < 5 \* çap VE

çap > = 4 mm VE

çap = < 76 mm VE

burç kullanılıyor VE

çap\_toleransı helisel\_matkapla\_delmede\_geçerli VE

yerleştirme\_toleransı helisel\_matkapla\_delmede\_geçerli VE

doğrusallık\_toleransı > = 0.6 mm VE

yuvarlaklık\_toleransı > = 0.05 mm

O HALDE

operasyon karbid\_uçlu\_maktkapla\_delmedir

**Namlu matkabıyla delme:** Namlu matkapları, çapta oldukça dar tolerans ve doğrusallık gerektiren çok uzun delikleri delmede kullanılır. Yüzey kalitesi, 1 µm'den daha iyi olması istenen uzun delikler, genellikle namlu matkabıyla delinerek oluşturulabilir. Delme sırasında namlu matkapları, burçlarla klavuzlanmalıdır. Bu matkaplarla yapılan delme işlemleri, bazı deliklere uygulanması gereken raybalama işlemine olan ihtiyacı ortadan kaldırabilir. Namlu matkapları, delme burçlarıyla klavuzlandığı takdirde, delinen deliğin eksen çizgisinden sapma miktarı en fazla 0.03 mm olmaktadır. Bu matkaplarla, 0.25 µm değerinde yüzey kalitesi elde edilebilir. Derin delikleri delmede,

doğrusallık, dairesellik ve boyut gibi değerler, namlu matkabıyla, helisel matkaba göre daha dar toleranslar içinde elde edilebilir. Bu işleme ait bir örnek aşağıdaki şekilde verilebilir.

EĞER

özellik boydan\_boya\_delik VE  
sertlik = < 56 HRC VE  
yüzey\_pürüzlülüğü > = 0.25 µm VE  
derinlik > = 3 \* çap VE  
derinlik = < 120 \* çap VE  
çap > = 2 mm VE  
çap = < 50.8 mm VE  
burç kullanılıyor VE  
çap\_toleransı namlu\_matkapla\_delmede\_geçerli VE  
yerleştirme\_toleransı namlu\_matkapla\_delmede\_geçerli VE  
doğrusallık\_toleransı > = 0.025 mm VE  
yuvarlaklık\_toleransı > = 0.005 mm

O HALDE

operasyon namlu\_maktkapla\_delmedir

**Yağ delikli matkapla delme:**Yağ delikli matkapla delme operasyonu, derin delikleri elde etmede oldukça uygundur. Bu işlemle, doğrusallık toleransı ve yüzey kalitesi, helisel matkaplardan daha iyi bir şekilde elde edilir. Bu işleme ait örnek bir kural aşağıdaki şekilde verilebilir.

EĞER

özellik konik\_uçlu\_kör\_delik VE  
sertlik = < 56 HRC VE  
yüzey\_pürüzlülüğü > = 0.4 µm VE  
derinlik > = 3 \* çap VE  
derinlik = < 12 \* çap VE  
çap > = 4 mm VE  
çap = < 38 mm VE  
çap\_toleransı yağ\_delikli\_matkapla\_delmede\_geçerli VE  
yerleştirme\_toleransı yağ\_delikli\_matkapla\_delmede\_geçerli VE  
doğrusallık\_toleransı > = 0.4 mm VE  
yuvarlaklık\_toleransı > = 0.01 mm

O HALDE

operasyon yağ\_delikli\_maktkapla\_delmedir

**Çok ağızlı matkapla delme (Delik genişletme):** Bu işlem, maça konarak üretilen, zımbayla açılan veya önceden delinen delikleri büyütmede kullanılır. Bu nedenle orjinal delikler (ilk delikler) bu metodla üretilemez. Bu işlemle, adi helisel matkapla elde edilen kesme işlemlerinden daha iyi kalitede delikler üretilir. Bununla ilgili bir kural örneği aşağıda verilmiştir.

## EĞER

özellik konik_uçlu_kör_delik	VE
ön_delik açılmış	VE
sertlik = < 56 HRC	VE
yüzey_pürüzlülüğü > = 0.6 µm	VE
derinlik = < 10 * çap	VE
çap > = 6.3 mm	VE
çap = < 31.7 mm	VE
burç kullanılıyor	VE
pozitif_çap_toleransı > = 0.071 mm	VE
negatif_çap_toleransı > = 0.071 mm	VE
pozitif_yerleştirme_toleransı > = 0.06 mm	VE
negatif_yerleştirme_toleransı > = 0.06 mm	VE
doğrusallık_toleransı > = 0.5 mm	VE
yuvarlaklık_toleransı > = 0.04 mm	VE

## O HALDE

operasyon çok\_ağızlı\_matkapla\_delmedir

**Punta matkabıyla delme:** Punta matkabıyla, bir merkezleme deliği delinip genellikle tamlık yükseltilebilir. Çünkü merkezleme deliğini referans alan bir matkap, başlangıç noktasından ayrılmayacaktır. Punta matkabıyla delme operasyonu, takip eden delme operasyonlarına klavuzluk edip delinen deliklerin yerleştirme doğruluğunu arttırmak ve malzemedeki merkezleme delikleri elde etmede kullanılır.

### 8.1.1.3. Raybalama operasyonları

Tam ölçüsünde bir delik elde etmek için kullanılan raybalama, doğruluğu ve yüzey kalitesini arttıran bir bitirme işlemidir. Bu işlemle, optimum şartlar altında yüzey kalitesini, 0.8 µm veya daha az olacak şekilde elde etmek mümkündür. Raybalama operasyonu için kurallar burçlu ve burçsuz raybalama olmak üzere ikiye ayrılır. Bu işlem için hazırlanan bir kural örneği aşağıda verilmiştir.

## EĞER

özellik konik_uçlu_kör_delik	VE
sertlik = < 56 HRC	VE
ön_delik açılmış	VE
yüzey_pürüzlülüğü > = 0.4 µm	VE
derinlik > = 3 * çap	VE
derinlik = < 55 * çap	VE
çap > = 1 mm	VE
çap = < 75 mm	VE
burç kullanılıyor	VE

çap\_toleransı raybalamada\_geçerli VE  
 yerleştirme\_toleransı raybalamada\_geçerli VE  
 doğrusalık\_toleransı  $\geq 0.3$  mm VE  
 yuvarlaklık\_toleransı  $\geq 0.01$  mm  
 O HALDE  
 operasyon raybalamadır

#### 8.1.1.4. Delik işleme operasyonu

Delik işleme, mevcut bir deliği büyüterek tamlığını yükseltmek amacıyla yapılan bir işlemdir. Bu işlemle daha doğru ölçü ve daha iyi yüzey kalitesi sağlanır. Delik işleme, delik büyütme başlığı ile veya delik büyütme çubuğu ile gerçekleştirilir.

#### 8.1.1.5. Taşlama operasyonları

Taşlama, daha dar tolerans ve hassas yüzey kalitesi elde etmek üzere, uygulanan bir işlemdir. Ayrıca, taşlama işlemiyle, frezeleme ve öteki imalat işlemlerine göre, daha sert malzemelerden talaş kaldırılabilir. Genellikle taşlama operasyonu için hazırlanan kurallar, yüzey taşlama ile ilgilidir. Yüzey taşlama, daha dar tolerans ve hassas yüzey kalitesi elde etmek için, düzlem yüzeylere uygulanır. Bu işlemle ilgili bir kural örneği aşağıdaki şekilde verilebilir.

EĞER  
 özellik düzlem\_yüzey VE  
 sertlik  $< 91$  HRC VE  
 yüzey\_pürüzlülüğü  $< 0.2$   $\mu$ m VE  
 yüzey\_pürüzlülüğü  $\geq 0.025$   $\mu$ m VE  
 toleransı yüzey\_taşlamada\_geçerli  
 O HALDE  
 operasyon yüzey\_taşlamadır

#### 8.1.1.6. Honlama operasyonları

Honlama ve lepleme işlemleri, çok iyi yüzey kalitesi ve çok dar tolerans gerektiren yüzeylere uygulanır. Honlamaya ait bir kural örneği aşağıda verilmiştir.

EĞER  
 özellik konik\_uçlu\_kör\_delik VE  
 sertlik  $= < 82$  HRC VE  
 ön\_delik açılmış VE

yüzey\_pürüzlülüğü  $\geq 0.025 \mu\text{m}$  VE  
 derinlik  $\geq 15 * \text{çap}$  VE  
 çap  $\geq 4 \text{ mm}$  VE  
 çap  $= < 127 \text{ mm}$  VE  
 çap\_toleransı honlamada\_geçerli VE  
 yerleştirme\_toleransı honlamada\_geçerli VE  
 doğrusalılık\_toleransı  $\geq 0.025 \text{ mm}$  VE  
 yuvarlaklık\_toleransı  $\geq 0.005 \text{ mm}$   
 O HALDE  
 operasyon honlamadır

### 8.1.1.7. Klavuz çekme, silindirik ve konik havşa açma, oturma yüzeyi işleme

Konik havşa açma, silindirik havşa açma ve oturma yüzeyi işleme, genellikle delme işleminden sonra yapılır. Oturma yüzeyi işleme, bazen delme işleminden önce de yapılabilir.

**Klavuz çekme**, iç vida meydana getirmek için yapılan talaş kaldırma işlemidir.

**Silindirik havşalama**, sınırlı bir derinlik için delik ağzını genişletme operasyonu olarak tanımlanabilir.

**Konik havşalama**, deliğin girişini konik olarak meydana getirmek üzere uygulanan bir işlemdir.

**Oturma yüzeyi işleme**, bir deliğin başlama ve merkezlenmesini kolaylaştırmak için meyilli yüzeylere uygulanır. İşlem sonunda elde edilen yüzey düz olur.

Daha fazla bilgiyi temsil etmek için çeşitli kuralları ve olguları, sisteme ilave etmek mümkündür. Böylece operasyon seçme modülünün performansı artmış olur.

### 8.1.2. İşlem seçiminde sonuç çıkarma

**Sonuç çıkarma mekanizması:** Bitmiş parçayı üretmek için, sistemde mevcut uygulanabilir kuralları bularak her özelliğe ait işlemi belirler. Bunu yaparken kuralları sürekli birbiriyle ilişkilendirip bir çözüm elde etmek üzere araştırır. Sistem, işlem seçimi için geriye zincirleme ve derinlik ilk araştırma stratejisi kullanır. Referans yüzeyi seçme ve işlemlerin sıralanmasında olduğu gibi, sistem, özelliğe uygulanabilir bir kuralın sonuç kısmıyla başlar ve bunu sağlamak için tüm koşulları sağlamaya çalışır. Burada elde edilecek olan karar veya amaç, araştırılan kuralların bir parçasıdır. Herhangi bir kuralın tüm koşullarını sağlayan bu karar, bir sonuç olarak doğru olmalıdır.

İşlem seçimi için asıl giriş bilgisi, veri tabanında KYG'de temsil edilen parça model dosyasından alınır. Boyutlar, yerleştirme toleransları, ölçüler vs gibi belirli bilgiler istenildiğinde, program tarafından STEP dosyasından çıkarılabilir. Ancak sistemin ihtiyaç duyduğu tüm bilgiler, veri tabanına önceden yerleştirilirse, sistem sorgulama sırasında bu bilgileri kullanabilir.

## 8.2. Takım Seçimi

Bu bölümde, her operasyona uygun takım seçimi ele alınmıştır. İşlem planlamada, özelliği oluşturacak takımın seçimi, oldukça önemlidir. Çünkü seçilen takım belirli bir özelliği şekillendirir. Sonuçta uygulanan işlem, maliyeti ve yüzey kalitesini etkiler [8].

İlgili modüllerde referans yüzeyi seçimi ve işlemlerin sıralanması tamamlandıktan sonra, istenen özellikleri oluşturabilecek kesici takımların seçimi yapılır. Takım seçiminde, özelliklerin geometrisi, kesme payları, yüzey kalitesi, işlenen malzemenin sertliği ve tolerans değerleri önemli rol oynar.

Kullanıcı arayüzü ve sonuç çıkarma mekanizmasıyla sistemde yer alan takım seçimiyle ilgili kurallar işletilir. Sistem, sorgulama sırasında geriye zincirleme ve derinlik ilk araştırma tekniği uygulayarak operasyonlara uygun gelen takımları seçer.

Sistem, işlenecek parça malzemesi, özellikleri, geometrisi ve işlemlerin tipine bağlı olarak seçilecek takımın karakteristiklerini belirler. Sonra istenen takım malzemesi, takım tipi, takım geometrisi gibi karakteristikler belirlenerek buna uyan bir takımın takım dosyasında olup olmadığı araştırılır. Operasyonlara uygun gelen takımlar bulunduktan sonra, bunlarla ilgili bilgiler, operasyonlarla ilişkilendirilerek planlama kartına yazılır. Veri tabanındaki takım dosyası, mevcut işlemleri yerine getirebilecek takımlardan meydana gelir. Takım dosyası modüler bir yapıda hazırlandığından, yeni takımlara ihtiyaç duyulduğunda, bilgi tabanı etkilenmeden gerekli yenilikler yapılabilir. Kesme takımlarına ait özellikler Prolog olguları olarak veri tabanında depolanır.

### 8.2.1. Veri tabanında kesme takımlarını temsil etme

Kesme takımları veri tabanında, Prolog olguları olarak temsil edilir. Tipik olarak bir kesme takımı: Takımın adı/numarası; aksenal talaş, radyal talaş ve yerleştirme açıları ile, çapı, takma uç sayısı (diş), efektif uzunluğu, takma uç şekli ve takım malzemesi ile tanımlanır. Takım ve iş parçası çifti için kesme hızı, diş başına ilerleme ve kesicinin/dişin ISO veya TS kod numarası istendiğinde veri



tabanından elde edilebilir. Veri tabanında temsil edilen takımlar aşağıda verilmiştir.

**/\* Kesici takım veri tabanı \*/**

takim(alin\_freze1, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(45),  
cap(130), dis\_sayisi(8), efektif\_boy(63), takma\_uc\_sekli(kare),  
takim\_malzemesi(karbid), takim\_kod\_no(p\_20)).

takim(alin\_freze1, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(45),  
cap(140), dis\_sayisi(10), efektif\_boy(75), takma\_uc\_sekli(kare),  
takim\_malzemesi(karbid), takim\_kod\_no(p\_20)).

takim(hss\_alin\_freze1, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(0),  
cap(100),dis\_sayisi(6), efektif\_boy(50), takma\_uc\_sekli(y),  
takim\_malzemesi(hss), takim\_kod\_no(ts303\_11)).

takim(parmak\_freze1,aksenel\_meyil(p),radyal\_meyil(p),klavuz\_acisi(0),  
cap(24), dis\_sayisi(4), efektif\_boy(75), takma\_uc\_sekli(y),  
takim\_malzemesi(hss),takim\_kod\_no(ts303\_16)).

takim(parmak\_freze1, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(0),  
cap(30), dis\_sayisi(4), efektif\_boy(100), takma\_uc\_sekli(y),  
takim\_malzemesi(hss), takim\_kod\_no(ts303\_16)).

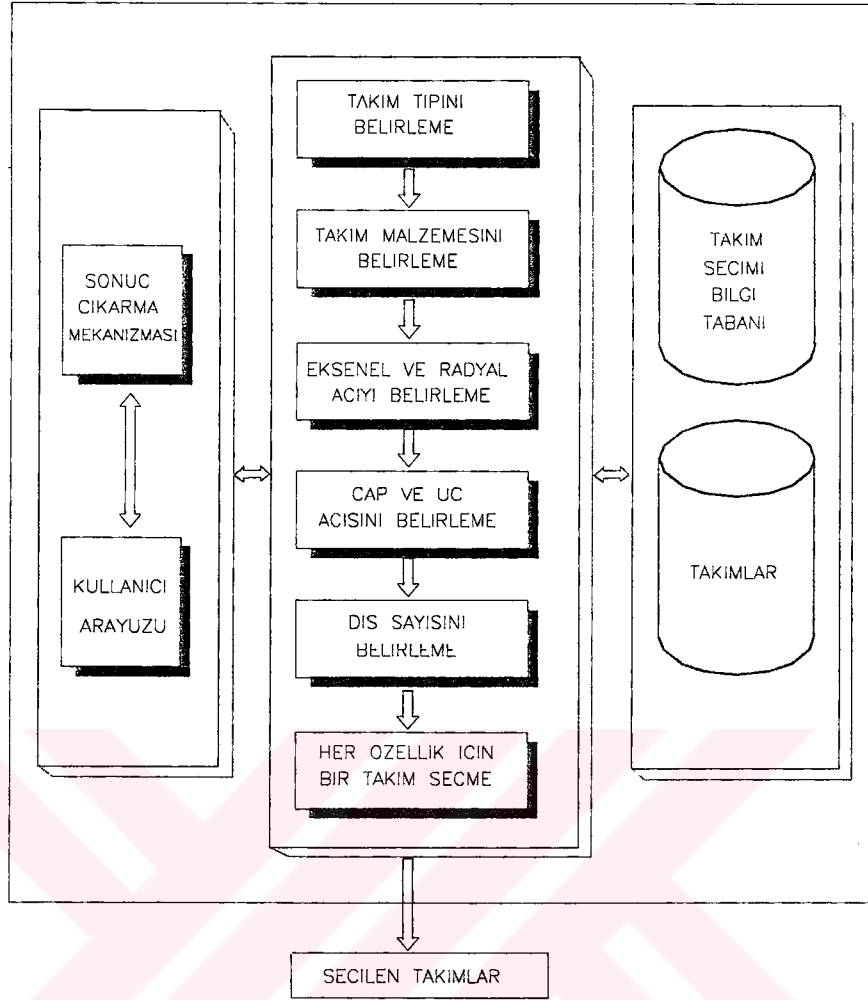
takim(helisel\_matkap1, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(0),  
cap(20), dis\_sayisi(2), efektif\_boy(100), takma\_uc\_sekli(y),  
takim\_malzemesi(hss), takim\_kod\_no(ts62\_4)).

takim(helisel\_matkap2, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(0),  
cap(18),dis\_sayisi(2), efektif\_boy(100), takma\_uc\_sekli(y),  
takim\_malzemesi(hss), takim\_kod\_no(ts62\_4)).

takim(helisel\_matkap3, aksenel\_meyil(p), radyal\_meyil(p), klavuz\_acisi(0),  
cap(16), dis\_sayisi(2), efektif\_boy(100), takma\_uc\_sekli(y),  
takim\_malzemesi(hss), takim\_kod\_no(ts62\_4)).

### **8.2.2. Kesme takımı seçme prosedürü**

Operasyona uygun düşecek bir kesme takımını seçmede aşağıdaki prosedür yerine getirilir (Şekil 8.2).



Şekil 8.2. Kesici takım seçme modülü

- 1- Kesici tipini seçme,
- 2- Takım malzemesini tayin etme,
- 3- Talaş açısını belirleme,
- 4- Çap ve klavuz açısını tayin etme,
- 5- Diş sayısını hesaplama,
- 6- Takım kodunu seçme,
- 7- Her özellik için kesicileri seçme.

**Kesici tipini seçme:** Genel olarak uygun bir takım tipini seçmek için, bazı kurallar hazırlanmıştır. Kesme takımları, belirli operasyonları yapmak üzere tasarlanır. Böylece takımın tipini seçmede kullanılacak kurallar, operasyonların

tipine bağılı olarak hazırlanır; yani takımın tipi, işlemin tipine göre belirlenir. Takım seçimiyle ilgili olarak hazırlanan kurallar aşağıda verilmiştir.

EĞER  
işlem yüzey\_frezeleme  
O HALDE  
kesici\_takım alın\_frezedir

EĞER  
işlem cep\_frezeleme  
O HALDE  
kesici\_takım parmak\_frezedir

**Takım malzemesini belirleme:** Genellikle frezeleme ve delme işlemlerinde en çok kullanılan iki ortak kesme takım malzemesi, Yüksek Hız Çeliği (HSS) ve karbidlerdir [108]. Genellikle HSS takımlar, yumuşak ve sünek; karbidler ise, daha sert malzemelerde kullanılır. Takım malzemesinin seçimi, işlenecek malzemenin sertliğine bağılıdır. Takım malzemesi seçimi için genel kurallar aşağıdaki gibidir.

EĞER  
sertlik > 22 HRC  
O HALDE  
kesici\_takım\_malzemesi karpiddir

EĞER  
sertlik =< 22 HRC  
O HALDE  
kesici\_takım\_malzemesi yüksek\_hız\_çeligidir

Bununla beraber, kurallar uygulanan operasyonlara bağılı olarak biraz değişir. Karbid takımlar, mümkün olan her yerde tercih edilir. Yüzey, kanal ve kademe frezeleme için sertlik 22 HRC'den daha fazla ise, takım malzemesi olarak karbid takımlar seçilir. Öte yandan sertlik 22 HRC'ye eşit veya daha az ise, karbid veya HSS takımlar seçilebilir. V-kanal, T-kanal, kırlangıç kuyruğu kanal frezeleme ve helisel matkapla delme gibi, bazı özel operasyonlarda genellikle HSS takımlar kullanılır. Delme işlemlerinde takım malzemesi seçimi için bazı kurallar aşağıda verilmiştir.

EĞER  
işlem helisel\_matkapla\_delme VE  
sertlik =< 22 HRC  
O HALDE  
kesici\_takım\_malzemesi HSS dir

EĞER

işlem takma\_uclu\_matkapla\_delme  
sertlik > 22 HRC

VE

O HALDE

kesici\_takım\_malzemesi karbiddir

**Talaş açılarını belirleme:** Talaş açısı, takım tabanına paralel düzlem ile talaş yüzeyi arasındaki açı olarak tanımlanır. Eğer yüzey, kesme kenarından aşağıya doğru meyilli ise, talaş açısı pozitif; yukarıya doğru meyilli ise, talaş açısı negatiftir. Talaş açısı ne kadar büyükse daha düşük kesme kuvvetleri, daha düşük güç tüketimi ve daha iyi yüzey kalitesi sağlar. Bununla beraber büyük talaş açısı, kesme kenarına sağlanan desteği azaltır. Genelde sert malzemeler için küçük talaş açıları kullanılırken yumuşak dövülebilir malzemelerde daha büyük talaş açıları kullanılır. Çift pozitif geometri (eksenel ve radyal pozitif talaş açısı), yumuşak malzemeler için oldukça uygundur. Çift negatif geometri ise sert malzemeler için oldukça uygun düşer. Bir freze çakısına ait ekstenel ve radyal talaş açıları, Şekil 8.3'te gösterilmiştir.

Pozitif ekstenel açı ve negatif radyal açı, tüm malzemelerde iyi bir kesme sağlar. Eğer çift pozitif veya çift negatif açı yoksa, sert ve yumuşak malzemeler için pozitif ekstenel açı ve negatif radyal açı, sürekli geçerli olan açılardır. Talaş açılarının seçiminde kullanılan kurallar aşağıda verilmiştir.

EĞER

sertlik = &lt; 22 HRC

VE

O HALDE

ekstenel\_radyal\_talaş\_açıları\_pozitifdir

EĞER

sertlik &gt; 22 HRC

VE

sertlik &lt; 40 HRC

O HALDE

ekstenel\_radyal\_talaş\_açıları\_pozitifdir\_ve\_negatifdir

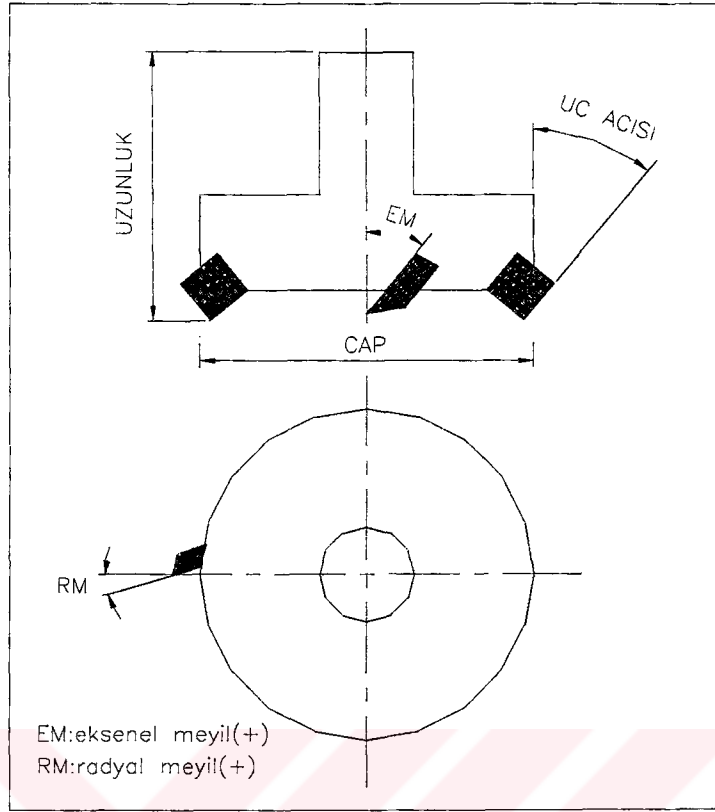
EĞER

sertlik &gt;= 40 HRC

O HALDE

ekstenel\_radyal\_talaş\_açıları\_negatifdir

**Klavuz açısını belirleme:** Klavuz açısı, tezgah, kesme takımı ve iş parçası üzerinde oluşan kesme kuvvetlerini etkiler. Klavuz açısı büyüdükçe radyal kuvvetler azalırken ekstenel kuvvetler artar. Klavuz açısı sadece Şekil 8.3'te görülen yüzey frezesi için kullanılır. Klavuz açısı tayin edilirken genellikle şu ifade



Şekil 8.3. Kesme takımının geometrisi

daima geçerlidir: Frezeleme sonucunda birbirine dik iki yüzey elde edilecekse, klavuz açısı daima  $0^\circ$  olmalıdır. Aksi halde bu açı genellikle  $45^\circ$ dir.

**Takım çapı ve boyunu belirleme:** Bir operasyon için seçilen takım çapı, kesicinin etkin çapıdır. Takım çapının belirlenmesinde, özelliğin boyutları önemli rol oynar. Delik özelliği için takım çapı, işlenecek deliğin çapıdır. Kanal özelliği için seçilen takım, kanalın genişliğine eşittir. Frezelemede en iyi sonucu elde etmek için, takım çapı işlenen alanın genişliğinin 1.6 katı olmalıdır [109].

Takım çapı,  $\text{Çap} = 1.6 \times \text{Genişlik}$  formülü kullanılarak hesaplanır. Eğer hesaplanan çap, takım dosyasında bulunan en büyük çaplı takımdan daha küçükse, bu çap; aksi halde mevcut takımlardan en büyük çapa sahip olanı seçilir. Etkin uzunluk, işlenecek özelliğin derinliğine bağlı olarak tayin edilir.

**Diş sayısının belirlenmesi:** Kesici takımın adımı, dişler arasındaki mesafeyi gösterir. Genellikle iş parçası ile en az iki dişin temasını sağlayacak bir

takımın seçiminde, kesici adımı etkili olur. Bir takımın çapı sabit kalmak kaydıyla, diş sayısının artırılması, adımın küçülmesine sebep olur. Bu da, aynı anda iş parçasıyla temas eden diş sayısının artmasını sağlar. Elde edilecek yüzeyin kalitesine bağlı olarak, en fazla diş sayısına veya en küçük adıma sahip takım, tavsiye edilir. Kesici takımın diş sayısı, aşağıdaki eşitlikten hesaplanır.

$$Z = \frac{1.4 \Pi D}{W} \quad (8.1)$$

Bu eşitlikte: Z, Diş sayısı; D, Kesici çapı; W, Kesme genişliğidir.

**Takım kodu seçimi:** Takımın TSE veya ISO kod numarası, kesme takımı ve iş parçası malzemesi ile operasyonlara bağlı olarak veri tabanında takım\_kod\_no olarak isimlendirilen olguları araştırarak seçilir.

**Her özellik için takım seçimi:** Yukarıda açıklanan takımın özellikleri, ele alınan temel özellikleri karşılayacak şekilde veri tabanında tayin edilir. Takım çapı, diş sayısı ve uzunluk kontrol edilir. Eğer takım tayin edilen özellikleri karşılırsa, seçilir; aksi halde veri tabanında mevcut olan başka bir takımın özellikleri kontrol edilmek üzere ele alınır.

### 8.3. Takım Tezgahı Seçimi

Önceki bölümde ele alınan kesme takımlarının seçiminden sonra, işlem planlamada takip edilen adım, parçayı imal edecek takım tezgahının seçimidir. Her özellik için ilgili takım tezgahının seçimi, işlem planlama faaliyetlerinde önemli görevlerden biridir. Bir özellik için belirlenen operasyonu yerine getirmede, uygun bir takım tezgahı seçilmelidir. Operasyonları yapmak üzere seçilen takım tezgahları, işleme tamlığını, yüzey pürüzlülüğünü ve önemli derecede maliyeti etkiler.

Takım tezgahı seçme modülü için giriş bilgileri, sıralanmış özellik listesi, belirlenen operasyonlar, parça tasviri ve seçilen kesme takımlardan ibarettir. Takım tezgahı seçme programı icra edilirken sonuç çıkarma mekanizması, her operasyona göre uygun tezgahı belirlemede, tezgah seçimini sağlayan kuralları bulur ve uygular. Sonuç çıkarma mekanizması, geriye zincirleme ve kurallara ulaşmak için derinlik ilk araştırma stratejisi kullanır. Veri tabanında yeterli bilgi olduğu sürece sistem, operasyonlar için gereken tezgahları otomatik olarak

seçer. Operasyonlara göre uygun takım tezgahlarını seçmek için bazı koşullar karşılanmalıdır. Bu koşullar:

- Takım tezgahının tamlığı, istenen işleme tamlığına uygun olmalıdır.
- Takım tezgahının iş yapma kapasitesi, iş parçasının en büyük boyutlarını yerleştirmek için yeterli büyüklükte olmalıdır.
- Tezgahın gücü operasyonları yapmaya yeterli olmalıdır.

Sistemde takım tezgahı seçimi aşağıdaki faktörlere bağlı olarak yapılır.

- Parça ölçüsü,
- Operasyon tipi,
- Özellik tipi,
- Doğruluk,
- Yüzey pürüzlülüğü,
- Sertlik.

Bir takım tezgahı yüzeyleri, belirli yüzey pürüzlülüğü değerinde yapabilir. İş parçasının sertliği, takım tezgahında yapılacak operasyon seçimini etkiler. Örneğin çok iyi yüzey kalitesi isteniyorsa, bu yüzey taşlama tezgahlarında işlenebilir. Benzer şekilde yüzey pürüzlülüğü iyi değil ve freze tezgahı ile yapılabilir durumda ise, bu tezgah tercih edilir. Freze tezgahı, operasyona bağlı olarak yüzeyi işlemek üzere seçilir. Yüzey pürüzlülüğü ve sertlik faktörleri, operasyon seçiminde dikkate alınır. Sertlik faktörü, takım tipi seçimini etkilediğinden, takım seçme modülünde de dikkate alınır. Tezgah seçimini etkileyen diğerler faktörler, takım tezgahı seçim modülünde ele alınır.

Parça, seçilen takımlar, sıralanan özellik listesi ve seçilen operasyonlar hakkında bilgi elde edildikten sonra, bilgi tabanındaki ilgili kuralları uygulayarak bir tezgah tipi belirlenir. Daha sonra işleme parametreleri amprik formülleri kullanarak hesaplanır. Tezgah, karar verilen tezgah tipine göre, takım tezgahı dosyasından elde edilir ve taslak modelin ölçüsü, takım tezgahının iş kapasitesine göre kontrol edilir. Eğer tezgah iş parçasını yerleştirebilecek ölçüde değilse, aynı isme sahip başka bir tezgah tipi alınır. İkinci olarak takım tezgahının tamlığı ile özellik için belirlenen tolerans aralığını yapma kapasitesinde olup olmadığı kontrol edilir. Aday tezgah, istenen gücü de sağlıyorsa, o zaman operasyon için bu tezgah seçilir. Tüm işlem planlama süreci tamamlandığında, seçilen takım tezgahları, özellikler, operasyonlar, işleme parametreleri ve kesme takımları işlem planlama dosyasına kaydedilir.



### 8.3.1. Veri tabanında takım tezgahlarını temsil etme

Takım tezgahı seçimi veri tabanı, tezgah seçimi için kurallar ve olgularla, takım tezgahları ve kesme hızı dosyaları gibi veri dosyalarından meydana gelir.

Fabrika ortamındaki takım tezgahları, teknolojik gelişmelere bağlı olduğundan, takım tezgahı veri tabanı, gerekli değişiklikleri yapmaya izin verecek esneklikte olmalıdır. Hatta takım tezgahı veri tabanı, planlama için tek tek veri elemanlarını elde etmek üzere, kolay ve hızlı bir şekilde erişilebilen formatlarda depolanmalıdır [110].

Veri tabanında takım tezgahları Prolog olguları olarak temsil edilir. Bir takım tezgahı, tezgahın ismi ve özellikleriyle birlikte okunaklı bir formatta belirlenir. Tezgahın özellikleri, tezgahın ismi/numarası, iş kapasitesi (X,Y,Z eksenlerinde hareket kapasitesi), ilerleme ve hız kademesi, hızlı hareket, maksimum takım çapı ve uzunluk, takım değiştirme zamanı, doğruluk, güç, şirket kodu, çalışma pozisyonu (düşey ve yatay) ve kontrol mekanizması (BSD, manüel) gibi parametreleri içerir. Verilerle ilgili bu özellikler, tezgah ismi ve istenen veriler için bir değişken ismi kullanılarak elde edilebilir. Böylece her takım tezgahı için, istenen belirli verilere kolayca erişilebilir. Veri tabanında temsil edilen takım tezgahlarına ait bazı örnekler aşağıda verilmiştir.

#### **/\* Takım tezgahı veri tabanı \*/**

```
takim_tezgahi(freze1,x_ksen_hareketi(400,mm), y_ksen_hareketi(350, mm),
z_ksen_hareketi(300, mm), hiz_araligi(1,7000, mm_min),
devir_araligi(45, 3000, dev_dak), max_takim_capi(180, mm),
max_takim_uzunlugu(250, mm), takim_degistirme_zamani(40, s),
tamlik(0.008), guc(20, kw), calisma_konumu(dusey), kontrol(cnc)).
```

```
takim_tezgahi(isleme_merkezi1, x_ksen_hareketi(500, mm),
y_ksen_hareketi(450, mm), z_ksen_hareketi(400, mm),
hiz_araligi(1, 6000, mm_min), devir_araligi(45, 5000, dev_dak),
max_takim_capi(100, mm), max_takim_uzunlugu(250, mm),
takim_degistirme_zamani(25, s), tamlik(0.005), guc(30, kw),
calisma_konumu(yatay), kontrol(cnc)).
```

```
takim_tezgahi(matkap1, x_ksen_hareketi(400, mm), y_ksen_hareketi
(350,mm), z_ksen_hareketi(500, mm), hiz_araligi(0, 300, mm_dk),
devir_araligi(60, 2000, dev_dak), max_takim_capi(36, mm),
max_takim_uzunlugu(250, mm), takim_degistirme_zamani(30, s),
tamlik(0.05), guc(6, kw), calisma_konumu(dusey), kontrol(elle)).
```

### 8.3.2. Takım tezgahı seçme prosedürü

Özellik listesini ve seçilen operasyonları elde ettikten sonra, prosedür, her özellik için uygulanır. Takım tezgahı seçme modülü Şekil 8.4'te verilmiştir. Takım tezgahının seçiminde uygulanan prosedür ana hatlarıyla aşağıdaki gibi yerine getirilir:

1. Sıralanan özellik listesini alma,
2. Tezgah tipini belirleme,
3. İşleme parametrelerini hesaplama,
4. Takım tezgahını belirleme,
5. Ölçüyü kontrol etme,
6. Tamlığı kontrol etme,
7. Güç kontrolü,
8. Uygun takım tezgahı seçimi.

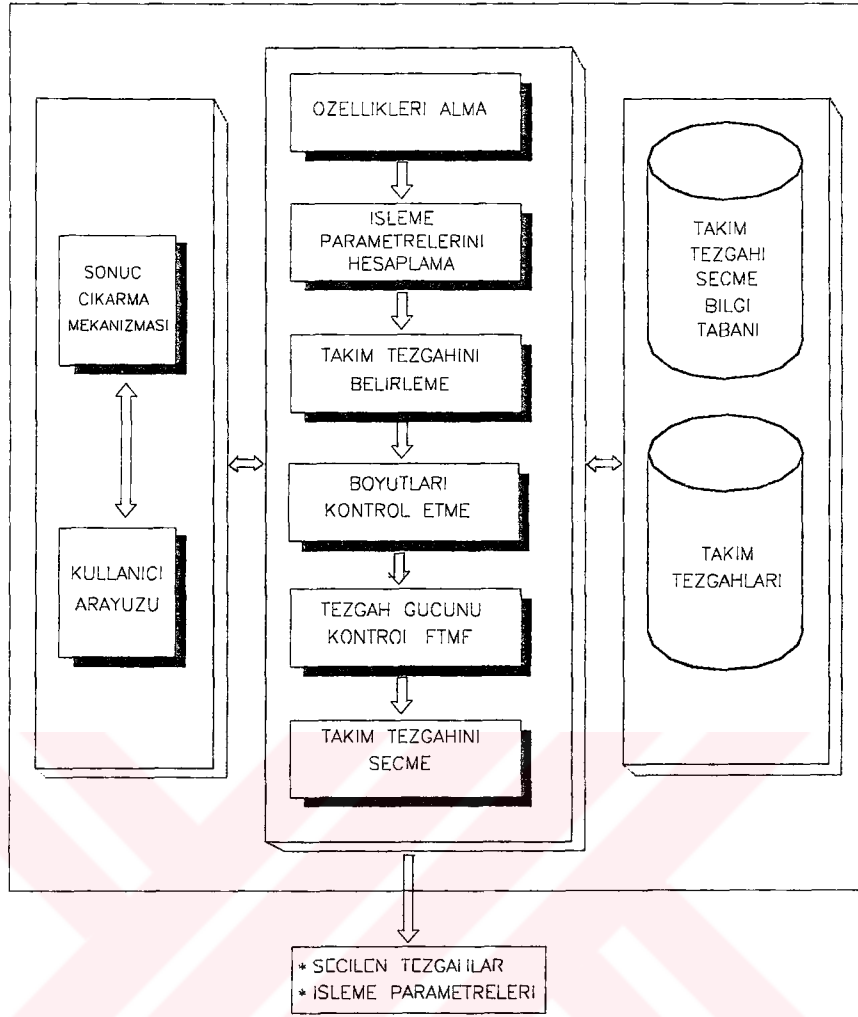
#### 8.3.2.1. Tezgah tipini belirleme

Tipik bir takım tezgahı, belirli operasyonları yapma kapasitesinde olacak şekilde tasarlanır. Örneğin matkap tezgahı, delme operasyonlarını; freze tezgahı ise, frezeleme operasyonlarını yapar. Takım tezgahı seçimi için hazırlanan kurallar, tezgahın yerine getirebileceği işleme operasyonlarının tipine bağlı olarak hazırlanır.

İşlenen özellik için, veri tabanından operasyonun ismi elde edildikten sonra, sonuç çıkarma mekanizması, tezgah tipi hakkında bir karar vermek için, geriye zincirleme tekniği ve derinlik ilk araştırma stratejisi kullanarak tezgah tiplerine ait kuralları uygular. Sonuç çıkarma mekanizması, sonucu çıkardıktan sonra, özellik için istenen operasyonu, hangi tezgahın yerine getirme kapasitesine sahip olduğunu belirler. Tezgah tipi seçimi kurallarına ait bazı örnekler aşağıda verilmiştir.

EĞER  
operasyon yüzey\_frezeleme  
O HALDE  
tezgah frezedir

EĞER  
operasyon delik\_delme  
O HALDE  
tezgah matkaptır



Şekil 8.4. Takım tezgahı seçme modülü

### 8.3.2.2. İşleme parametrelerinin hesaplanması

Operasyonlar, kesme takımları ve işleme parametrelerine karar verilmedikçe, takım tezgahı seçilemez. Örneğin güç, tezgah seçimi için önemlidir [8]. Tezgah seçme prosedüründe sonraki adım ise, güç, hızlar ve takım ömrü vs gibi işleme parametrelerini belirlemektir. Sistem, parçanın imalat isteklerini karşılayan işleme parametrelerini belirler.

İşleme parametrelerinin seçimi, takım malzemesi, iş parçası malzemesi, takım çapı ve kesme derinliğine bağlıdır. İşlem planlamada, belirli işleme operasyonları için, doğru işleme parametrelerini belirlemek zorunludur [4].

Diğer koşullar sabit tutulduğunda, kesme hızı veya ilerleme artırılırsa, gerçek işleme zamanı azalır, fakat takım aşınma miktarı artar. Bu yüzden çok

düşük hızlar ve ilerlemeler üretim zamanını arttırır. Alternatif olarak, çok yüksek kesme hızları, sık sık takım değiştirmeyi gerektirdiğinden, üretim süresini ve takım maliyetini yükseltir. Açıkcası minimum üretim zamanı elde etmek için, optimum koşullar sağlanmalıdır. Benzer şekilde optimum koşullar, minimum üretim maliyetini ortaya çıkaracaktır. Düşük hızlar ve ilerlemeler, uzun işleme zamanları gerektirdiği için, yüksek hızlar ve ilerlemeler de, sık sık takım değiştirme ihtiyacı yüzünden maliyetleri yükseltecektir. Genelde bu amaçlara eş zamanlı olarak erişilemediğinden dolayı, burada problem üretim zamanı ve maliyetinin birlikte nasıl optimize edileceğidir. Etkin talaş kaldırma için işlem parametreleri, genellikle üretim miktarını maksimum veya üretim maliyetini minimum yapma esasına dayanır [61]. Sistemdeki işleme parametreleri seti maksimum üretim miktarı esasına göre belirlenmiştir.

Sistem, optimum işleme parametrelerini belirlemek için, ampirik eşitlikleri kullanır. İşleme parametreleri aşağıda sıralanan değerlerden meydana gelir.

- Takım ömrü,
- Kesme hızı,
- İş mili hızı,
- İlerleme,
- Kaldırılan talaş miktarı,
- Güç.

**Takım ömrü ve kesme hızının hesaplanması:** Parça üretiminde diğer işleme parametrelerini hesaplamak için, **kesme hızı** ve **takım ömrü** arasındaki ilişkiyi bilmek gerekir. Takım ömrü, kesme takımının verimli bir şekilde kullanıldığı, zaman periyodu olarak tanımlanabilir. Taylor'un, Şekil 8.5'teki takım ömrü ve kesme hızı değişkenleri arasındaki ampirik ilişkiyi gösteren çalışması şu şekildedir.

$$\frac{v}{v_r} = \left( \frac{t_r}{t} \right)^c \quad (8.2)$$

$v$  = kesme hızı (m/dk),

$v_r$  = verilen kesme hızı (m/dk),

$t$  = takım ömrü (dk),

$c$  = sabit,

$t_r$  = verilen kesme hızı  $v_r$  için belirli takım ömrü (dk).

$t_r$ 'nin değeri, belirli bir parça-takım malzemesi çifti ve belirli bir hız için bulunabilir.  $c$ , esas itibariyle takım malzemesine bağlı bir üsttür. HSS kesme takımları için,  $c = 0.125$ ; karbid takımlar için de  $0.25 < c < 0.3$  alınır. Şekil 8.6'da takım ömrü 60 sn verildiğinde, farklı takım ve iş parçası malzemeleri için  $v_r$  değerleri görülmektedir. Eşitlik 8.2'den belirli bir durum için takım ömrü  $t$ , aşağıdaki gibi hesaplanır.

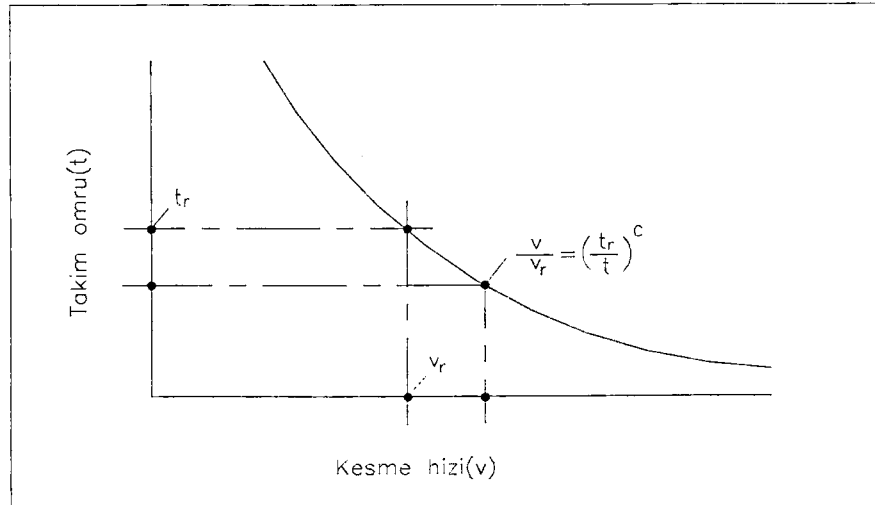
$$t = t_r \left( \frac{v_r}{v} \right)^{\frac{1}{c}} \quad (8.3)$$

Minimum üretim zamanı için kesme hızı  $v_p$  ise şöyle hesaplanır.

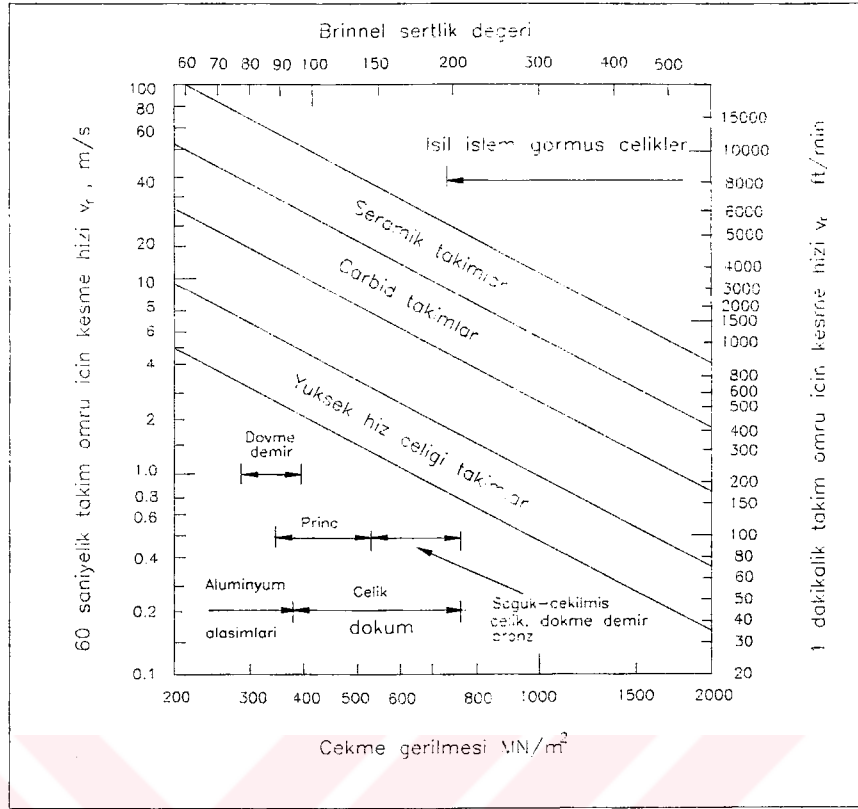
$$v_p = v_r \left( \frac{c}{1-c} \cdot \frac{t_r}{t_{ct}} \right)^c \quad (8.4)$$

$t_{ct}$  = takım değiştirme zamanı.

Minimum üretim zamanı için optimum takım ömrü  $t_p$ , Taylor'un takım ömrü eşitliği, eşitlik 8.3'te, eşitlik 8.4'ü yerine koymak suretiyle elde edilebilir.



Şekil 8.5. Taylor'un takım-ömür bağıntısı [111]



Şekil 8.6. Takım ömrü ( $t_r$ ) 60 sn için kesme hızı ( $v_r$ ) değerleri [111]

$$t_p = \left( \frac{1-c}{c} \right) t_{ct} \quad (8.5)$$

Eşitlik 8.5'te  $(1-c)/c$  faktörünün pratik olarak kullanımı, yüksek hız çeliği takımlar için 7, karbid takımlar için 3 olarak alınır. Sonuç olarak optimum kesme hızlarına karşılık gelen  $v_p$ 'nin değeri, aşağıdaki eşitlikten bulunabilir.

$$v_p = v_r \left( \frac{t_r}{t_p} \right)^c \quad (8.6)$$

**Devir sayısı:** Kesme hızı tayin edildikten sonra, devir sayısı (iş mili hızı) aşağıdaki eşitlikten hesaplanabilir.

$$n = \frac{1000 v_p}{\pi D} \quad (8.7)$$

$n$  = Devir sayısı (dev/dk),

$D$  = Takım çapı (mm).

**İlerleme:** Takip eden iki diş arasındaki iş parçası hareket mesafesidir ve önceden bulunan iş mili hızına bağlı olarak hesaplanır. Frezede ilerleme, eşitlik 8.8'den, matkapta ilerleme ise eşitlik 8.9'dan hesaplanır.

$$s = s_z \cdot z \cdot n \quad (8.8)$$

$s$ : ilerleme hızı (mm/dk),

$s_z$ : diş başına ilerleme miktarı (mm/diş),

$z$ : takımın diş sayısı.

$$s = s_n \cdot n \quad (8.9)$$

$s_n$ : her dönüş için ilerleme (mm/dev).

**Kaldırılan talaş miktarı:** İşlem sırasında kaldırılacak talaşın hesaplanması için, kullanılan kesme derinliği ve genişliği belirlenmelidir. Kaldırılan talaş miktarı, kesme derinliği, genişlik ve ilerleme ile ilgilidir. Frezede talaş kaldırma miktarı, eşitlik 8.10'dan, matkapta ise eşitlik 8.11'den bulunur.

$$V = \frac{w \cdot h \cdot s}{1000} \quad (8.10)$$

$V$ : Kaldırılan talaş miktarı (cm<sup>3</sup>/dk),

$w$ : Kesme genişliği (mm),

$h$ : Kesme derinliği (mm).

$$V = \frac{\pi D^2}{4000} s \quad (8.11)$$

**Güç:** Birim güç, kaldırılan birim talaş hacmi için gerekli enerji olarak tanımlanır ve kW/cm<sup>3</sup>/dk olarak ifade edilir. Birim güç iş parçası malzemesi,



operasyon tipi, kesme takımının talaş açısı, talaş kesiti ve hıza bağlı olarak değişir. Operasyonların güç ihtiyacını hesaplamak için kullanılan birim güç değerleri, işleme verileri el kitaplarından alınır [112]. Belirli bir hızda ilerlemede ve kesme derinliğinde bir malzemeyi işlemek için, takım malzemesi ve geometrisi, kesme kuvvetleri oluşturur ve güç tüketir. Bu yüzden takım tezgahının seçiminde gücün hesaplanması gerekir [108]. Seçilen tezgah, parçayı işlemek için yeterli gücü sağlamalıdır. Bir operasyonda gerekli güç, takım tezgahı seçiminde önemli bir faktördür. Tezgah bu istekleri karşılamak için yeterli güce sahip olmalıdır [112].

Herhangi bir takım tezgahında güç kayıpları meydana gelir. Bu bakımdan tezgahın motor gücü, kesme için gerekli olan güçten daha fazla olmalıdır. Genel şartlarda %80'lik verim doğru bir tahmindir. Güç, talaş kaldırma miktarının, birim güç ile çarpımının, tezgah verimine bölünmesi ile hesaplanır. Hızlar, ilerlemeler ve talaş kaldırma miktarı hesaplandıktan sonra, güç tüketimi aşağıda verilen formülden tayin edilebilir.

$$P_m = \frac{V \cdot p_u}{\eta} \quad (8.12)$$

$P_m$ : Motor gücü (kW),

$P_u$ : Birim talaş hacmi için güç (kW/cm<sup>3</sup>/dk),

$\eta$ : Tezgahın verimi.

### 8.3.2.3. Takım tezgahının belirlenmesi

Tespit edilen özelliklere uyan bir tezgah, veri tabanında bulunan takım tezgahlarından seçilir.

**Ölçü kontrolü:** Tezgahlar, üretilen parçaların ölçüsünü kısıtlayacaktır. Bir tezgahın iş kapasitesi boyutları, üretilebilecek en büyük iş parçası boyutlarını belirler [112].

Takım tezgahının iş yapma kapasitesi ve stok ölçüsü karşılaştırılır. Eğer parça boyutları, tezgahın iş kapasitesinden daha büyükse, o zaman aynı türden başka bir takım tezgahı bulunur ve ölçüsü kontrol edilir. Aksi halde program, tezgah seçim prosedüründeki takip eden adıma geçer.

**Tamlık kontrolü:** Takım tezgahı seçiminde dikkate alınması gereken diğer bir faktör tamlıktır. Takım tezgahı, istenen tamlığı sağlayacak özellikte

olmalıdır. Her takım tezgahı, belirli tolerans sınırları içinde parçaları işleyebilir. Örneğin freze tezgahı, 25'ten 50 µm'ye kadar bir tolerans içinde, parçaları işleme kapasitesindedir. Seçilen tezgah, gereken toleransları karşılayabilmelidir.

Program, özelliğin tolerans sınırlarını ve tezgahın tamlığını karşılaştırarak tezgahının istenen toleransı sağlayıp sağlayamadığını kontrol eder.

**Güç kontrolü:** Program, parçayı işlemede gerekli olan güç ile takım tezgahının gücünü karşılaştırır. Eğer istenen güç, tezgahın gücünden daha fazla ise, o zaman veri tabanındaki başka bir takım tezgahı araştırılır. Aksi halde program, tezgah seçiminde takip eden adımı sürdürür.

**Uygun takım tezgahını seçme:** Tezgah seçimi için gerekli olan faktörler kontrol edildikten sonra, bu faktörlere sahip tezgah, operasyon için seçilir ve çıkış dosyasına yazılır.

Üretilen özellik için takım tezgahı seçimi ve işleme parametrelerinin hesaplanması tamamlandıktan sonra, tüm prosedür, kalan özelliklere göre, uygun takım tezgahlarını seçmek için tekrarlanır.

Bir parça için, işlem planlama süreci sırasında, her operasyon, hesaplanan işleme parametreleri, işlem planlama kartındaki seçilen takım tezgahı ismi ve numarası ile birleştirilir.

## 9. BAĞLAMA İŞLEMİ VE ARAPARÇA OLUŞTURMA

### 9.1. Bağlama İşleminin Yapılması

Daha önceki bölümlerde, referans yüzeyi seçimi, operasyon seçimi, operasyonların sıralanması, kesme takımları ve takım tezgahlarının seçimi açıklanmıştı. Bu bölümde de, her operasyon için bağlama işlemi ve araparça modellerin oluşturulması açıklanmıştır.

Operasyonların sıralanması, operasyon seçimi, kesiciler ve takım tezgahlarından başka; bağlama işlemi, imalatta tamlığı etkilediğinden işlem planlamada önemli faaliyetlerden biridir. Bağlama işlemi, iş parçalarını takım tezgahlarına yerleştirmek ve tespit etmek için gereklidir [108].

Takım ve iş bağlama düzenleri, iş parçalarını emniyetli ve hassas bir şekilde tutmak için kullanılır. **İş bağlama düzeni**, iş parçalarını yerleştirmeye ve tespit etmeye yarayan bir ayardır. **İş bağlama ve takım klavuzlama düzeni** ise, iş parçalarını yerleştirme ve tespit etme yanında, üzerinde bulunan delme burçları ile, operasyon sırasında, kesme takımlarına klavuzluk yapar. İş bağlama ve takım klavuzlama düzeni, genellikle delme, raybalama ve ilgili operasyonlar için kullanılır. Delik delme burçları, bir parçada operasyon yapılırken, işleme doğruluğunu sağlamak için kullanılır [113].

Esas itibariyle parçaları pozisyona getirmek ve tespit etmek için, iki tip bağlama sistemi kullanılır. Bunlar, **modüler** ve **özel bağlama sistemleridir**. Modüler bağlama sistemi, bir plaka üzerine çok sayıda delik ve T-kanalı yerleştirilmiş bağlama elemanlarından oluşur. Çok değişik konfigürasyonlar, bu elemanların farklı bileşimlerini kullanmak suretiyle elde edilir. Aynı zamanda, temel plaka üzerine tespit edilen elemanların, çeşitli konfigürasyonları sayesinde, modüler bağlama sistemi, büyük ölçüde esneklik sağlar. Özel bağlama sistemleri, işlenecek belirli iş parçaları için tasarlanmış, özel amaçlı bağlama aparatlarıdır [114]. Bağlama işleminde yerleştirme, destekleme ve tespit etme elemanları olmak üzere üç tip eleman kullanılır. Bağlama işlemi ile ilgili önemli kurallar, parçayı yerleştirme ve tespit etmeden ibarettir. Bundan dolayı bağlama modülü, istenen desteklemeyi, yerleştirmeyi ve tespit etme pozisyonlarını belirlemek için, parçayı işleme sırasında destekleme, yerleştirme ve tespit etmede, uygun yüzeyleri tanıma üzerinde odaklanır.

Sistemde bağlama işlemi, seçilen operasyonlara ve işlenecek özelliğin geometrik karakteristiklerine göre, yerine getirilir. İşlenecek parçayı yerleştirmek ve tespit etmek için, Prolog cümlecikleri formundaki bağlama işlemi ile ilgili

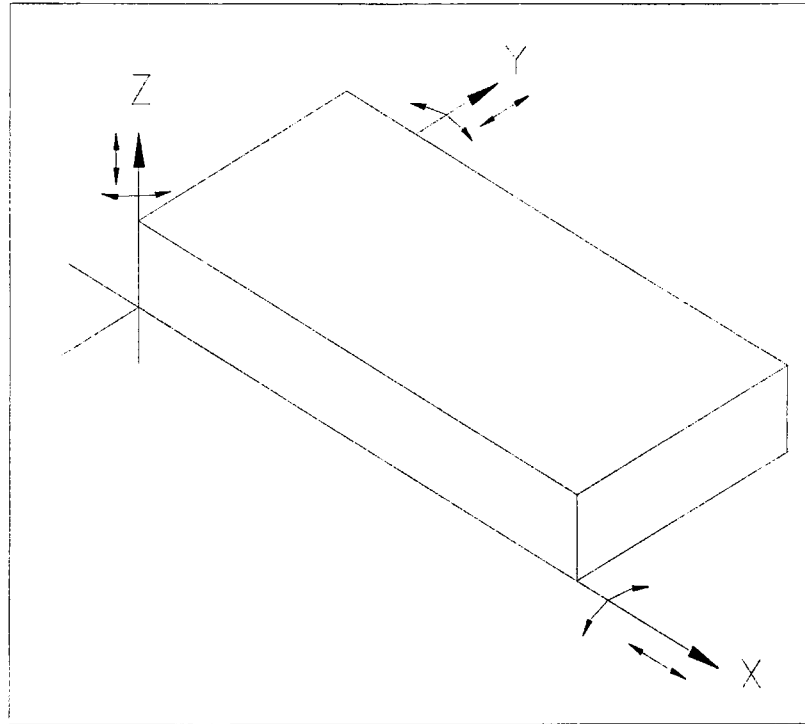
kurallar icra edilerek **6 nokta (3-2-1) metodu** uygulanır. Parçayı bağlamadan önce, taslak ve bitmiş parça modelleri, KYG'de temsil edilir ve veri tabanında depolanır. Sistem, ileriye planlama stratejisi işletir. Yani sistem, taslak parça modeli ile başlar ve bitmiş parça modeline ulaşıncaya kadar, her özellik için operasyonları uygular. Taslak parçanın geometrisini değiştiren ilk operasyon için bağlama işlemi, taslak parça modelin geometrisine göre yerine getirilir. Güvenilir kararlar vermek ve takip eden operasyon için uygun yerleştirme, destekleme ve tespit yüzeylerini belirlemek için, ilk operasyondan sonra oluşturulan yeni araparça model, modellendirilmeli ve ikinci operasyon için bağlama işlemi yapılırken mevcut olmalıdır. Bu sebeple her operasyondan sonra işlenen bölge, parça modelinden kaldırılır ve yeni araparça model oluşturularak KYG'de temsil edilir. Böylece takip eden operasyon için bağlama işlemi başlamadan önce, geçerli model, işlem için hazır olur. Operasyondan sonra oluşturulan yeni yüzeyler, takip eden operasyon için yerleştirme ve tespit noktaları belirlenirken dikkate alınır.

#### 9.1.1. 6 Nokta metodu

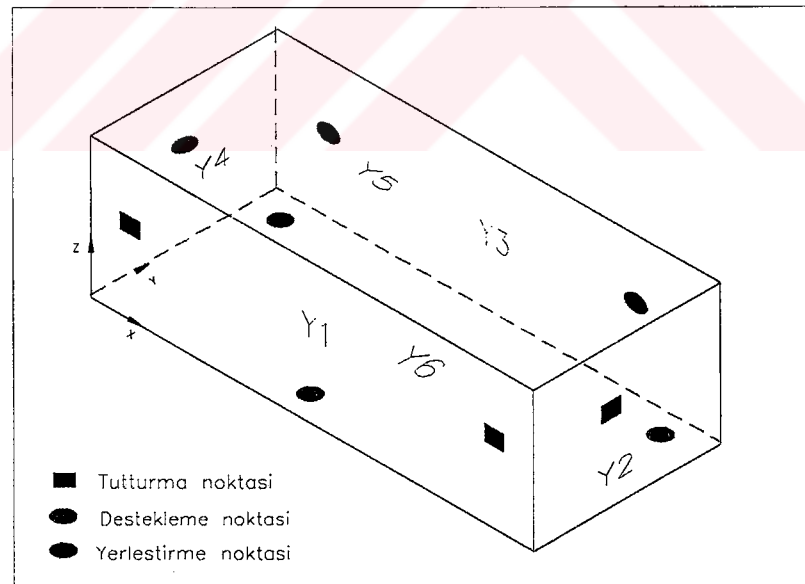
**6 nokta metodu**, parçayı birbirine dik üç düzleme yerleştirmek ve destekleme ve yerleştirme noktaları üzerine tespit etmek için kullanılır. Bir parça 12 serbestlik derecesine sahiptir. Yani üç boyutlu uzayda herhangi bir yönde hareket etmek için serbesttir (Şekil 9.1). Başka bir ifadeyle parça üç doğrusal (**x**, **y**, **z**) eksen boyunca hareket etmek veya bu eksenlerin etrafında dönmek üzere serbesttir [115].

İşleme sırasında, iş parçasının 12 serbestlik derecesi, bir iş bağlama düzeni veya iş bağlama ve takım klavuzlama düzeniyle tekrarlanabilir bir şekilde, parçayı aynı yere yerleştirmek ve iş parçasının hareketine engel olmak için sınırlandırılmalıdır. Parçanın 12 serbestlik derecesi, altı uygun seçilmiş **destekleme, yerleştirme ve tespit etme** noktalarıyla sınırlandırılabilir. Bu, **6 nokta** yerleştirme metodu veya **3-2-1** yerleştirme metodu olarak bilinir.

Çoğu durumlarda, yerleştirme ve destekleme elemanları istenen referans konumunu tesis etmek için, aynı amaca hizmet eder. Destekleme elemanları, yatay düzlemde parçayı yerleştirmek ve desteklemek için, iş parçasının altına yerleştirilen basit konumlandırma elemanlarıdır. Yerleştirme elemanları ise, iş parçasını düşey eksen üzerinde pozisyona getirir. Bir iş parçası, 6 nokta prensibini kullanmak suretiyle (eğer düzlemler birbirine dik ise) **üç** noktadan bir düzleme, **iki** noktadan diğer bir düzleme ve **bir** noktadan da üçüncü bir düzleme yerleştirilebilir [113]. Şekil 9.2'de gösterildiği gibi üç



Şekil 9.1. Tespit edilmemiş bir bloğun uzaydaki serbestlik derecesi



Şekil 9.2. Bir parçayı 6 nokta prensibine göre yerleştirme ve bağlama

elemanı, **1** doğrusal ve **4** dönme hareketini sınırlamak için, **destekleme yüzeyi (Y6)** üzerine yerleştirilir. İki yerleştirme elemanı, **1** doğrusal ve **2** dönme hareketini sınırlandıran ve destekleme yüzeyine dik olan **ilk yerleştirme yüzeyi (Y3)** üzerine yerleştirilir. Kalan bir yerleştirme elemanı da, destekleme ve ilk yerleştirme yüzeyine birlikte dik olan **ikinci yerleştirme yüzeyi (Y4)**'e tahsis edilir. Bu durumda **Y4**, **1** doğrusal hareketi sınırlandırır. **6** destekleme ve yerleştirme noktasının kullanılması ile **9** hareket sınırlandırılır. Geriye kalan üç doğrusal hareket, destekleme ve yerleştirme noktalarına karşı bağlama işlemiyle sınırlandırılır. Destekleme ve yerleştirme elemanları, iş parçasının stabilitesini arttırmak için mümkün olduğu kadar aralıklı yerleştirilmelidir.

### 9.1.2. Bağlama prosedürü

Bağlama prosedürü başlamadan önce, yüzeylerin boyutları, komşu yüzeyler, birbirine dik olan yüzeyler, karşılıklı yüzeyler, üzerine özellik yerleşen yüzeyler, taslak ve bitmiş parça modelleri Prolog olguları şeklinde belirlenerek veri tabanında temsil edilir ve depolanır. Bağlama programı başladığında sıralanmış özellik listesi veri tabanından alınır. Prosedür genellikle, sıralanmış özelliklerden ilk olarak referans yüzeyi olarak belirlenen yüzeye/özellığe uygulanır. Bağlama prosedürü ana hatlarıyla aşağıdaki gibi sıralanır ve parçaya uygulanır.

- Özelliğin tipini belirleme,
- Özelliğe uygulanacak operasyonu belirleme,
- Destekleme yüzeyini bulma,
- Destekleme ve yerleştirme yüzeyleri için en uygun kenarı belirleme,
- İlk ve ikinci yerleştirme yüzeylerini belirleme,
- Bağlama yüzeylerini bulma.

#### 9.1.2.1. Yüzey frezeleme için bağlama işlemi

**Destekleme yüzeyinin seçimi:** İşlenecek yüzeyin yönü bulunduğundan sonra, bunun karşısındaki yüzey elde edilir ve destekleme yüzeyini belirlemek için ilgili kural uygulanır. Bu işlem için kullanılan kuralın genel formu aşağıdaki şekildedir.

" EĞER bir yüzey işlenecek yüzeyin karşısında ise ve bu yüzey referans yüzeyi ise, O HALDE bu yüzeyi destekleme yüzeyi olarak seç, AKSİ HALDE başka bir yüzeyi



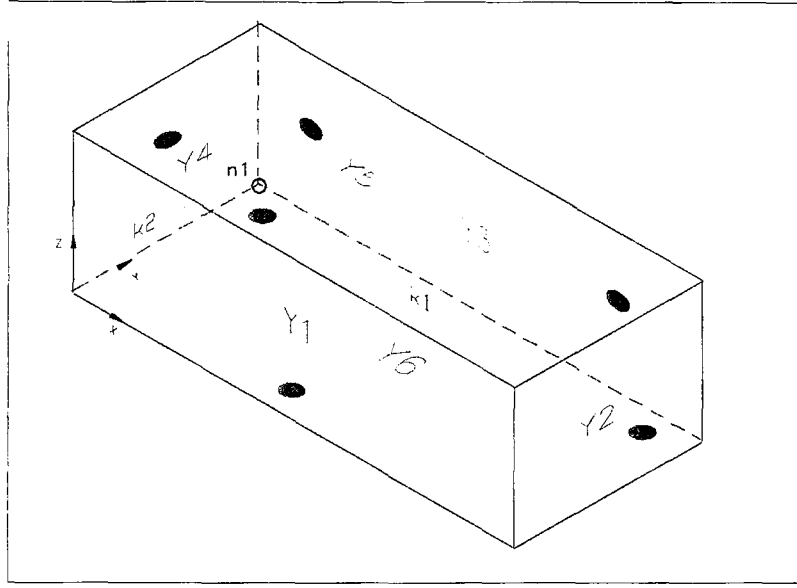
dene. EĞER kontrol edilen yüzeylerin hiçbirisi işlenecek yüzeyin ve referans yüzeyinin karşısında değilse, O HALDE işlenecek yüzeyin karşısındaki yüzeyi bul ve onu destekleme yüzeyi olarak deklere et."

**Destekleme ve yerleştirme yüzeylerinin belirlenmesi:** Destekleme yüzeyi belirlendikten sonra, birinci ve ikinci yerleştirme yüzeylerini tayin etmek üzere, destekleme yüzeyi ile ortak kenarları bulunan iki komşu yüzey, veri tabanından alınır. Destekleme elemanlarını kolayca yerleştirmek ve daha stabil hale getirmek için, iki yerleştirme ve destekleme yüzeyi tarafından paylaşılan kenarların uzunlukları kontrol edilir. Şekil 9.3'teki parçada **Y5** yüzeyinin işleneceği ve **Y6** yüzeyinin de destekleme yüzeyi olarak seçildiği kabul edilsin. Uygun kenar seçildikten sonra, destekleme yüzeyi ile ilk ve ikinci yerleştirme yüzeylerinin birbirine dik olup olmadıkları kontrol edilir. Seçilen destekleme ve yerleştirme yüzeyleri birbirlerine dik değilse, dikliği sağlayan başka bir kenar araştırılır.

**İlk ve ikinci yerleştirme yüzeylerinin belirlenmesi:** Destekleme yüzeyinin en uygun kenarını seçen kurullarla, yerleştirme yüzeyleri de belirlenir. Belirlenen yüzeylerden **ilk yerleştirme yüzeyi olan** yüzeye, iki yerleştirme noktası; **ikinci yerleştirme yüzeyi** olarak belirlenen yüzeye de bir yerleştirme noktası tahsis edilir. İş parçasının stabilitesini ve iyi yerleştirilmesini sağlamak için, daha uzun olan yüzey, ilk yerleştirme yüzeyi olarak tercih edilir. İlk yerleştirme yüzeyinin seçimi için dikkate alınması gereken diğer bir husus da referans yüzeyidir. En uygun kenarı paylaşan yüzey referans yüzeyi ise, bu yüzey, ilk yerleştirme yüzeyi olma önceliğine sahiptir. Bu mümkün değilse referans yüzeyi ikinci yerleştirme yüzeyi olarak denenir.

Şekil 9.3'te görülen parçada, **Y6** yüzeyinin destekleme yüzeyi ve bu yüzeyin **n1** köşesini paylaşan **k1** ve **k2** kenarları da, yerleştirme yüzeylerini tayin etmek üzere ortak kenarlar olarak belirlensin. Bu iki kenarın uzunluklarının kontrol edilmesiyle, **k1**'in daha uzun kenar olduğu bulunur. **k1** ve **k2** kenarlarına sahip yerleştirme yüzeyi olmaya aday, **Y3** ve **Y4** yüzeyleri kontrol edilerek, daha uzun kenarı paylaşan **Y3** yüzeyi ilk yerleştirme yüzeyi olarak seçilir. Daha kısa kenarı paylaşan **Y4** yüzeyi ise, ikinci yerleştirme yüzeyi olarak belirlenir. İlk ve ikinci yerleştirme yüzeyi belirlendikten sonra, yerleştirme noktalarının sayısı ilgili yüzeylere atanır. **Y6** destekleme yüzeyine, ikisi daha uzun kenar olan **k1** boyunca uzanacak şekilde, üç destekleme noktası tahsis edilir. İlk yerleştirme yüzeyi **Y3** üzerine iki yerleştirme noktası ve ikinci yerleştirme yüzeyi **Y4** üzerine de bir yerleştirme noktası yerleştirilir.





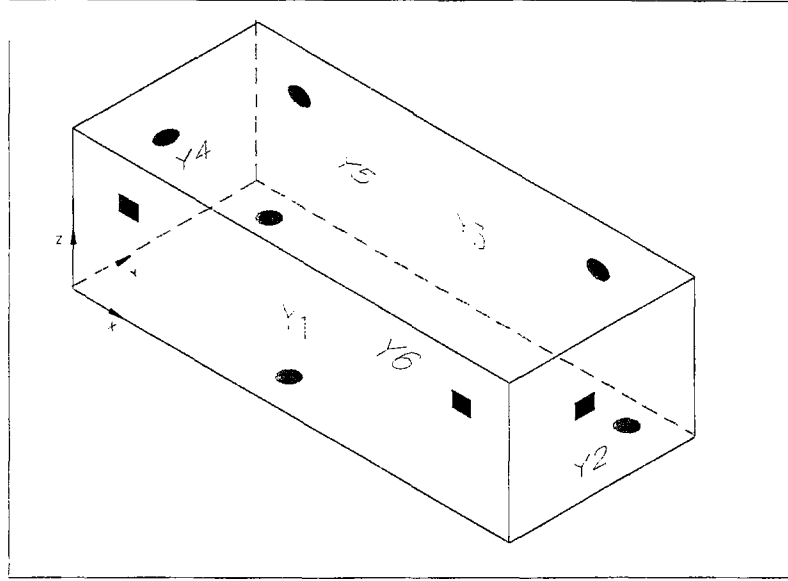
Şekil 9.3. Birinci ve ikinci yerleştirme yüzeylerinin belirlenmesi

**Tespit etme yüzeylerinin belirlenmesi:** Tespit etme yüzeylerinin belirlenmesi, yüzey/özellik üzerinde uygulanacak yüzey, kademe, kanal frezeleme; delme operasyonları vs. gibi, operasyonun tipine bağlı olarak gerçekleştirilir. Bir yüzey, yüzey frezeleme ile işlenecekse, bu yüzey üzerine tespit noktaları yerleştirilmemelidir.

Şekil 9.4'te gösterildiği gibi **Y1** yüzeyi, ilk yerleştirme yüzeyi **Y3**'ün karşısında olduğundan, **ilk tespit yüzeyi**; **Y2** yüzeyi de ikinci yerleştirme yüzeyi **Y4**'ün karşısında olduğundan **ikinci tespit yüzeyi** olarak seçilir. İlk yerleştirme yüzeyi **Y3** üzerindeki yerleştirme noktalarının karşısına, iki tespit noktası atanır. Bu durumda üzerine iki tespit noktası tahsis edilen ilk tespit yüzeyi **Y1**'dir. Bir tespit noktası da, ikinci yerleştirme yüzeyi **Y4** üzerindeki yerleştirme noktasının karşısında bulunan, ikinci tespit yüzeyi **Y2** üzerine yerleştirilir.

#### 9.1.2.2. Kademe frezeleme için bağlama işlemi

Kademe frezeleme işleminde destekleme, yerleştirme ve tespit yüzeyleri, yüzey frezeleme için bağlamaya benzer tarzda belirlenir.

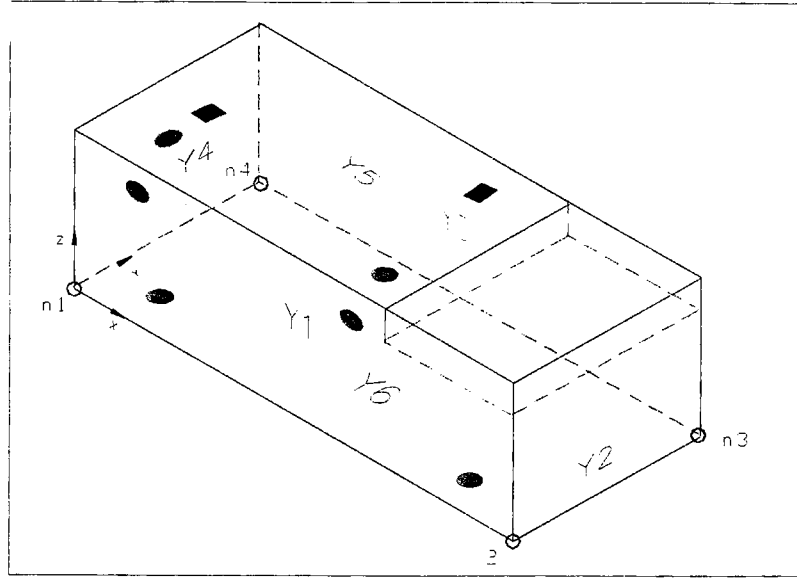


Şekil 9.4. Yüzey frezeleme için bağlama yüzeylerinin belirlenmesi

**Destekleme yüzeyinin belirlenmesi:** Destekleme yüzeyi, bitmiş modelden alınan kademe yüzeyleri ve yönlerine bağlı olarak belirlenir. Öncelikle kademe yüzeyleri ve referans yüzeyinin yönleri karşılaştırılır. Eğer referans yüzeyinin yönü, kademe yüzeylerinin birinin karşısında ise, o zaman referans yüzeyi, destekleme yüzeyi olarak seçilir. Aksi halde kademe yüzeylerinin birinin karşısında olan düz yüzey, destekleme yüzeyi olarak belirlenir.

Yüzey frezelemede bağlama işlemi için uygulanan prosedür, burada da kullanılarak, destekleme yüzeyi için en uygun kenar belirlenir. Örneğin Şekil 9.5'te gösterilen kademe özelliğinde, kademe yüzeylerinin yönünün biri ile aynı yönde olan **Y2** yüzeyinin, **n2** ve **n3** köşeleri arasında kalan kenar seçilmemelidir. Çünkü yerleştirme yüzeyi, işleme sahasından uzakta ise, parçayı yerleştirmek çok kolay olacaktır. Böylece geriye kalan **Y4** yüzeyinin **n1** ve **n4** köşeleri arasında kalan kenar, en uygun kenar olarak belirlenir.

**Yerleştirme ve tespit etme yüzeylerinin belirlenmesi:** Destekleme yüzeyi olarak belirlenen **Y6**'nın daha uzun kenarı bulunduktan sonra (Şekil 9.5), üç destekleme noktası bu yüzeye atanır. Destekleme yüzeyinin daha uzun kenarını paylaşan **Y1** yüzeyi, ilk yerleştirme yüzeyi olarak, daha kısa kenarını paylaşan **Y4** yüzeyi de, ikinci yerleştirme yüzeyi olarak seçilir. İlk yerleştirme yüzeyine iki yerleştirme noktası ve ikinci yerleştirme yüzeyine de bir yerleştirme noktası tahsis edilir.



Şekil 9.5. Kademe frezeleme için bağlama işlemi

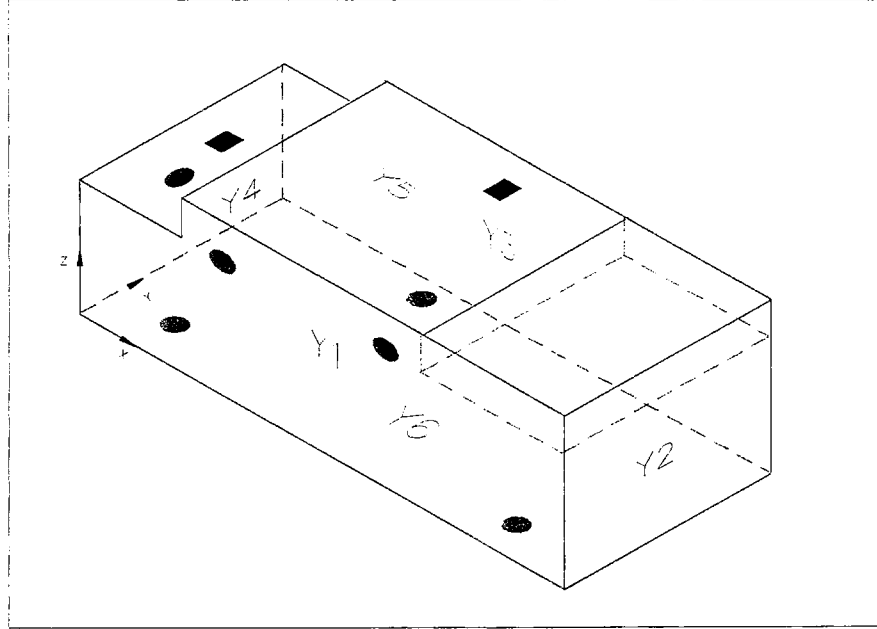
Destekleme yüzeyinin karşısında bulunan **Y5** yüzeyi, ilk tespit yüzeyi olarak seçilerek destekleme elemanlarına karşı, iki tespit noktası tahsis edilir. Destekleme yüzeyi karşısında olan diğer yüzeyler ise, alternatif bağlama yüzeyleri olarak tanımlanır (örneğin Şekil 9.6'nın sol tarafındaki kademe yüzeyi).

#### 9.1.2.3. Delme için bağlama işlemi

Delinecek deliği içeren **Y5** yüzeyi veri tabanından alınır (Şekil 9.7). Bunun karşısında bulunan **Y6** yüzeyi ise destekleme yüzeyi olarak seçilir. En uygun kenar bulunduktan sonra, yerleştirme ve bağlama yüzeyleri için kademe frezelemede kullanılan prosedür uygulanır.

#### 9.1.2.4. Kanal frezeleme için bağlama işlemi

İşlenecek kanalın yönü belirlendikten sonra, destekleme yüzeyi, kanal taban yüzeyinin yönüne bağlı olarak belirlenir (örneğin Şekil 9.7'de gösterilen **Y6**). Destekleme ve yerleştirme yüzeyleri için en uygun kenar bulunurken kesici yaklaşım yönündeki köşeler arasında kalan kenar (bu örnekte n1 ve n2 arası), dikkate alınmaz. Kesici yaklaşım yönünün karşısında bulunan **Y3** yüzeyini birinci yerleştirme yüzeyi olarak belirlemek için, öteki kenarlar kontrol edilir. Takip eden prosedür, kademe frezeleme için bağlama işleminde olduğu gibi yerine getirilir.

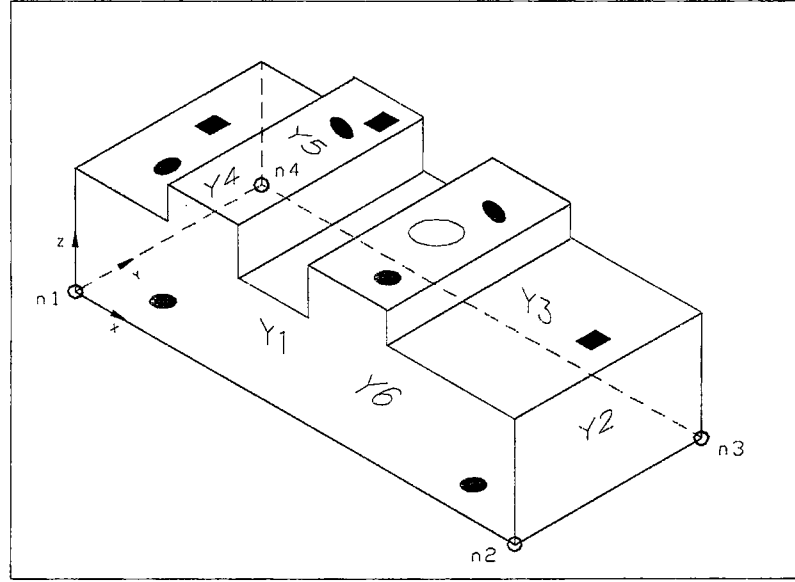


Şekil 9.6. Sol kademelin alternatif bağlama yüzeyi olarak kullanılması

## 9.2. Araparça Modelleri Oluşturma

Bağlama işleminin doğru bir şekilde yerine getirilebilmesi için, araparça modeller, iş parçasının son durumunu tanımlamada gereklidir. Parçanın herhangi bir özelliğine uygulanan bir operasyon, ilerde uygulanacak operasyonlar sırasında, bağlama işlemini etkilerse, destekleme, yerleştirme ve tespit yüzeylerinin seçimi araparça modellere bağlı olarak yerine getirilir [116]. Taslak, bitmiş ve her araparça model, KYG'de temsil edilerek veri tabanında depolanır. Araparçalar, özelliği işlemek için yapılan operasyona bağlı olarak oluşturulur. Her operasyon parçanın geometrik şeklini değiştirir. Operasyonlar yapıldıktan sonra, parça modeli operasyon tipine göre güncelleştirilir. Yeni araparça model, takip eden bağlama prosedürü başlamadan önce veri tabanında temsil edilmelidir. En sonunda taslak parça modeli, bitmiş parça modeline dönüştürülür. Bu dönüşüm sırasında araparça modeller icra edilen operasyona göre üretilir.

Bağlama prosedürü destekleme, yerleştirme ve tespit yüzeylerini belirlemede operasyona göre yerine getirilir; yerleştirme ve tespit etme işlemi ise, yüzeylere bağlı olarak yapıldığından, parça yüzeyleri hakkında yeterli bilgiye sahip olmak gerekir.



Şekil 9.7. Delik ve kanal frezeleme için bağlama işlemi

Taslak ve bitmiş parça modeline ait yüzey bilgisi yanında, ayrıca her araparça modelin yüzeyi hakkındaki bilgi de açık bir şekilde veri tabanında depolanmalıdır. Parçaya bir operasyon uygulanırken operasyonun tipine göre parçadan talaş kaldırılır. Üzerine operasyon uygulanan esas yüzeyler ve etkilediği yüzeyler belirlenerek model güncelleştirilir ve KYG'de temsil edilir. Parça modelinin bu dinamik değişikliği, takip eden operasyon için yeni üretilen yüzeyleri dikkate almak suretiyle, bağlama programına imkan sağlar.

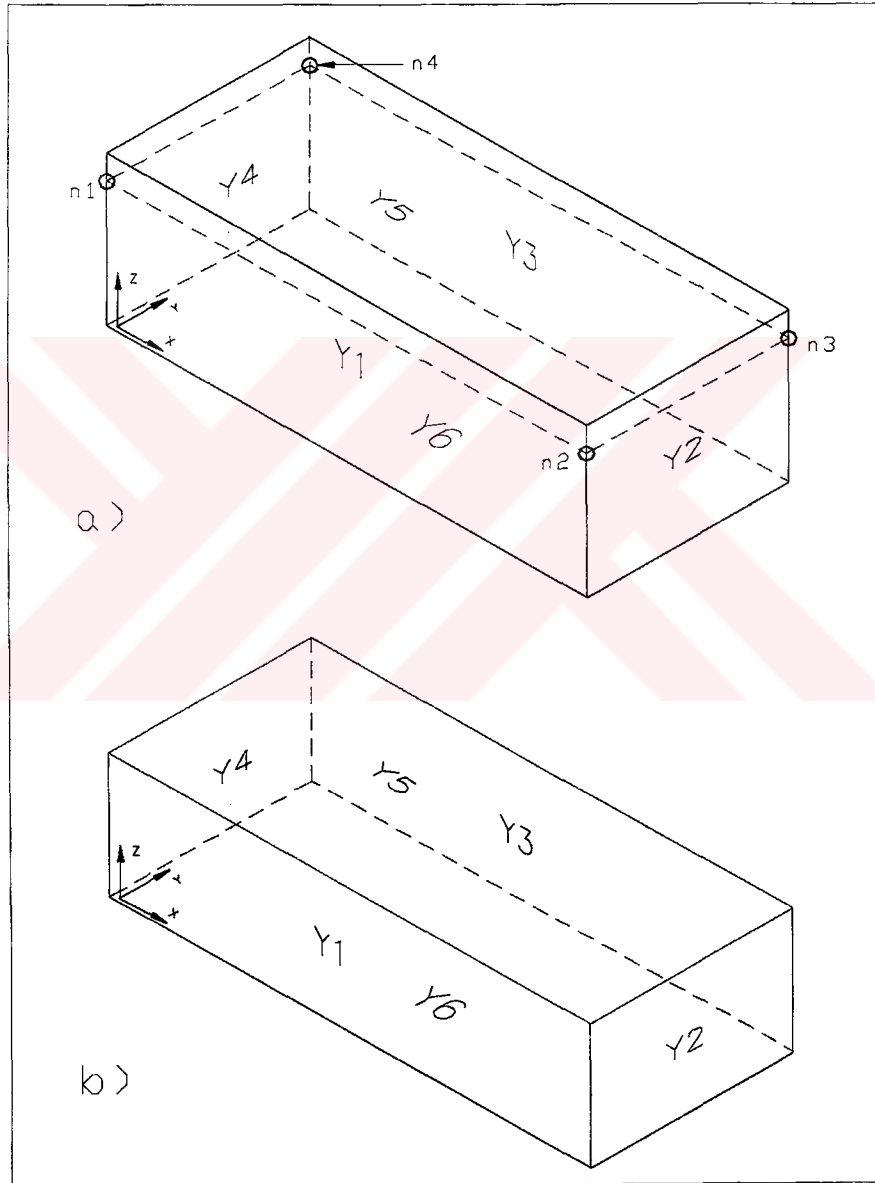
Her operasyon, yüzeyleri farklı şekillerde etkiler. Yeni araparça modelleri oluşturma prosedürü, operasyonun tipine bağlı olup ana hatlarıyla aşağıdaki gibidir.

- Operasyonla oluşturulacak yeni noktaların (köşeler) koordinatlarını belirleme,
- Parça modeline yeni yüzeyleri yerleştirme,
- Operasyonla etkilenen parça modelinin her yüzeyini güncelleştirme.

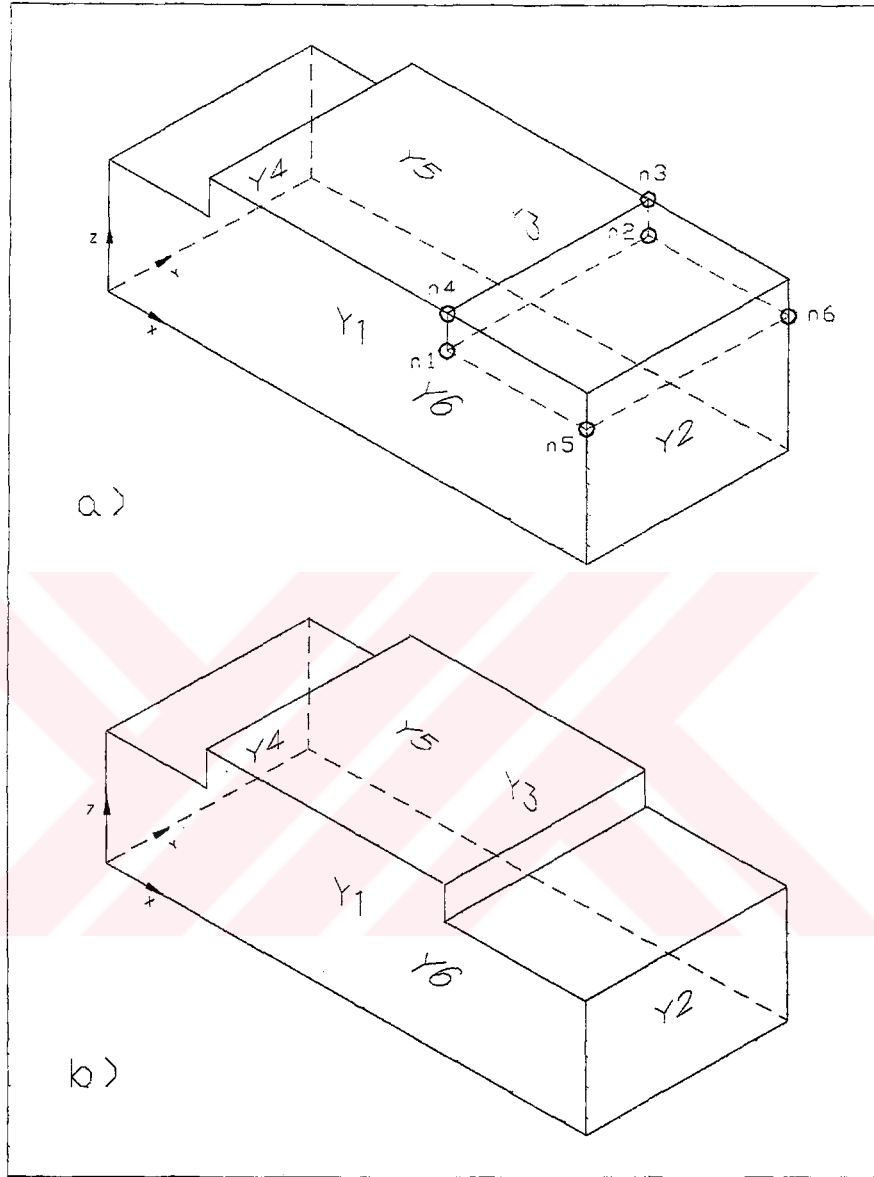
Bu prosedür yeni yüzeyleri veya özellikleri yerleştirmek ve eski yüzeyleri kaldırmak için, Bölüm 6'da açıklandığı gibi yerine getirilir. Örneğin Şekil 9.8.a'da gösterilen **Y5** yüzeyi üzerine yüzey frezeleme uygulandığında, oluşacak yeni noktaların koordinatları (**n1**, **n2**, **n3**, **n4**) kesme derinliğine bağlı olarak belirlenir. İşlenen **Y5** yüzeyi ve etkilenen (**Y1**, **Y2**, **Y3**, **Y4**) yüzeyleri değiştirildikten sonra, yeni araparça model KYG'de temsil edilir ve veri

tabanında depolanır (Şekil 9.8.b). Benzer şekilde kademe frezeleme operasyonu için yüzeyler üzerindeki yeni noktalar (**n1**, **n2**, **n3**, **n4**, **n5**, **n6**) yerleştirilir (Şekil 9.9.a). Etkilenen yüzeyler düzenlenir ve üretilen yeni model ve kademe yüzeyleri ile ilgili yeni modelde tanımlanır (Şekil 9.9.b).

Çalışmada ele alınan üç örnek parçanın işlem planlama, takım ve bağlama işlemi ile ilgili plan kartları Ek-D'de verilmiştir.



Şekil 9.8. Y5 yüzeyinin işlenmeden önce (a) ve işledikten sonraki (b) durumları



Şekil 9.9. Sağ kademe özelliğini işlemeden önce(a) ve işledikten sonra(b) oluşan araparça model



## 10. ARAŞTIRMANIN DEĞERLENDİRİLMESİ VE TARTIŞMA

Bu çalışmanın amacı, US yaklaşımı kullanarak işlem planlama ve bağlama faaliyetleri için karar verme işlemi otomatik hale getirmektir. Çünkü işlem planlama faaliyetleri, önemli ölçüde insan tecrübesi ve sezgisel bilgiye ihtiyaç duymaktadır. Algoritmik programlar, zamanla imalat işlemlerinde meydana gelen değişmelere paralel olarak, işlem planlama sistemlerini değiştirmede genelde yeterli esnekliğe sahip değildir. Çünkü programdaki herhangi bir değişiklik, yazılımın tüm yapısını etkiler [7]. Bu sebeple imalat bilgisi, işlem planlama mantığı ve tecrübenin en uygun durumunu elde etmek için US yaklaşımı tartışılmıştır.

İşlem planlama faaliyetlerine, US yaklaşımının uygunluğunu denemek amacıyla, US yaklaşımı kullanılarak işlem planlama ve bağlama işlemi için bir yazılım geliştirilmiştir. Geliştirilen sistem, stok seçimi, taslak parça tasarımı, bitmiş parça oluşturma, referans yüzeyi, operasyon seçimi ve sıralaması, kesici ve takım tezgahı seçimi, bağlama ve araparça model oluşturma modüllerinden meydana gelir.

Geliştirilen sistem, Prolog cümlecikleri formundaki STEP dosyasını alarak, örnek parça için bir taslak tasarlar ve stok ölçüsünü seçer. Yüzeyler arasındaki açılar hesaplandıktan sonra, yüzeylerin bulunan komşuluk ve içbükeylik-dışbükeylik ilişkileri, taslak ve bitmiş parça modelleri, Komşu Yüzey Grafiği (KYG) adlı özel bir formatta temsil edilir. Bu format, sistem için geliştirilmiş olup işlem planlama sisteminde doğrudan kullanılabilir. KYG, taslak, bitmiş ve araparça modelleri temsil etmek üzere kullanılan parça bilgisini sağlar. Ayrıca geliştirilen sistemle, referans yüzeyi seçimi, her özellik için operasyon seçimi ve operasyonların sıralanması, işleme parametrelerinin hesaplanması, kesici ve takım tezgahlarının seçimi ve her operasyon için bağlama işlemi belirlenir.

Geliştirilen sistem, uzman bir işlem planlayıcının yaptığı gibi, kurallar ve olgulardan oluşan sezgisel bilgiye başvurarak işlem planlama sistemi için gerekli olan sonuçları çıkarır.

Kurallar, VE/VEYA grafik teknikleri kullanılarak hazırlanmıştır. VE/VEYA grafik metodunun asıl avantajı, problemi alt problemler şeklinde ayrıştırmaya imkan sağlamasıdır. Böylece ayrıştırılan alt problemler, birbirinden bağımsız olarak çözülebilir.

Uygulanan US yaklaşımının avantajlarından birisi de kuralların herhangi bir sırada organize edilebilmesidir. Yeni kurallar ve olgular, programın tümünü değiştirmeksizin, ilgili program modüllerine yerleştirilebilir. Hazırlanan kurallar, birbirleriyle bağlantılı olarak uygulanabilir. Bu çalışmada elde edilen sonuçlar, işlem planlamada bulunan karmaşık muhakeme etme işleminin, US yaklaşımı kullanılarak otomatik hale getirilebileceğini göstermektedir.

Bu çalışmada, üretim kuralları EGER- O HALDE kuralları formundaki bilgiyi açıklamak için kullanılmıştır. Çünkü işlem planlama bilgisi genellikle "Eğer değişik koşullar sağlanırsa o halde gerçek eylem yerine getirilir" şeklinde ifade edilebilir.

Sistemde kullanıcı arayüzü, kullanıcı ve sistem modülleri arasındaki iletişimi kurarak sistem için gereken sonuçların elde edilmesini, işlem plan kartlarının gösterilmesini, düzenlenmesini ve çıktıların alınmasını sağlar.

Program modüllerinden kuralları ve olguları okuyarak gerekli sonuçları çıkartan sonuç çıkarma mekanizması, sonuçlara ulaşmak için geriye zincirleme ve derinlik ilk araştırma stratejisi uygulama yeteneğine sahiptir.

BDİP sistemler için bir parçanın geometrik açıklaması, GT esaslı parça kodları, özel parça tanımlama dilleri veya bir BDT modelden alınır (Bölüm 2). IGES, SET ve VDA-FS gibi çeşitli grafik standartları, BDT'den BDİP'e parçaların tabii formattaki geometrik bilgisini transfer etmek için kullanılmaktadır. STEP'in 1.0 versiyonu bir çalışma taslağı olarak 1988'de basılmış ve ISO tarafından uluslararası grafik verileri standardının ilk taslak teklifi olarak kabul edilmiştir [12]. Bu çalışmada parça tanımlamak için kullanılan STEP standardı hala geliştirilmekte olup ileriki yıllarda BDT sistemleri tarafından kullanılacağı beklenmektedir. Şu anda STEP standart formatın BDT çıktısı olmadığından, STEP'i kullanan ticari BDİP sistemler mevcut değildir. Diğer BDT grafik standartları, işlem planlamada gerekli olan tüm ürün verilerini temsil etmedikleri için, bu çalışmada, STEP standardı BDT verileri girişi için kullanılmıştır. Mevcut BDT grafik standartları bir ürünün sadece geometrik verilerini tasvir ederken STEP, işlem planlama faaliyetleri için gerekli olan geometrik veriler yanında, teknolojik bilgileri de temsil eder. Bu yüzden STEP'in ileriki yıllarda çok yaygın bir şekilde kullanılacak tabii veri değiştirme formatı olacağı düşünülmektedir.

İşlem planlama süreci sırasında, her operasyondan sonra parçadan malzeme kaldırılarak yeni bir araparça model oluşturulur. Bu araparça modeller, daha ileride yapılacak operasyonlarda bağlama işlemi ve işlem planlama hakkında güvenilir kararlar vermek için gereklidir. ST model (STEP dosyası) yüzeyleri, özellikleri, köşeleri, kenarları vs. içermekle beraber, imalat operasyon seçimi ve bağlama işlemlerine yardım etmek için, açılar, yüzeyler arasındaki

komşuluk ve içbükeylik-dışbükeylik ilişkileri gibi bazı ilave bilgilerin de açık bir şekilde bulunması gerekir.

Bu sebeple taslak, bitmiş ve araparça modelleri temsil etmek için KYG olarak anılan özel bir açıklama şeması geliştirilmiştir. KYG, operasyon seçimi ve bağlama işlemi için gerekli olan hazır ve kompakt bir formda, köşe ve kenar bilgisi içeren yüzeyler arasındaki açılar, komşuluk ve içbükeylik-dışbükeylik ilişkisi gibi daha açık bir şekilde bilgiye sahip olan parça modellerini mümkün kılar.

Taslak ve bitmiş parçanın temsil edilmesi sistem tarafından otomatik olarak tamamlandıktan sonra, bunlar veri tabanına yerleştirilir. Parça için bir referans yüzeyi seçilir ve parça özellikleri sıralanır ve seçilen operasyonlarla ilişkilendirilir. İlk yüzey/özellik işlenmeden önce yapılacak operasyona göre bağlama işleminin yerine getirilmesi gerekir. Operasyondan sonra işlenen bölge parçadan kaldırılarak yeni araparça model oluşturulur. KYG şeması, yüzeyler ve komşuluk ilişkilerine bağlı olduğundan, bir yüzey işlenirken bundan hangi yüzeylerin etkileneceği belli olur. Böylece sadece etkilenen yüzeyler değiştirilerek yeni modeller oluşturulur.

Geliştirilen sistem, minimum talaş kaldırma esasına göre, stok boyutlarını seçer ve taslak parçayı tasarlar. Belirlenen taslak parça boyutları, bitmiş parça modelini kapsayacak şekilde olur.

İşlem planlamada taslak parça tasarımı; döküm, dövme veya stok şekli olarak seçilen kare çubuk, lama ve platina gibi taslak parçanın imalat işlemine bağlı olarak yerine getirilir. Bu çalışmada taslak parçanın kare veya lama çubuk, ya da platinadan seçileceği kabul edilmiştir.

Bitmiş parçayı tanımlamadan önce, bazı gerekli bilgiler temin edilir. Yüzeylerin yönü çıkarıldıktan sonra, her yüzey için kumşu yüzeyler bulunur ve veri tabanında depolanır. Ortak kenarların kontrol edilmesiyle, yüzeyler arasındaki açılar hesaplanır ve içbükeylik-dışbükeylik ilişkileri belirlenir. Parça son olarak yüzey ve alt özellikleri de içerecek şekilde dosyaya yazılır.

Referans yüzeyini seçmek için sonuç çıkarma mekanizması, her yüzeyi kontrol ederek bunlarla ilgili kuralları uygular ve hangisinin referans yüzeyi olacağına karar verir. Referans yüzeyi, işlem planlama faaliyetlerinde operasyon sıralama ve bağlama işlemini doğrudan etkilediğinden, veri tabanında çok yüksek bir önceliğe sahiptir.

Operasyonların sıralanması için bilgi tabanı, **sıralama kuralları** ve - özellikler arasındaki öncelik ilişkisini göstermek üzere kullanılan- **öncelik seviye değerli kurallar** olmak üzere iki gruba bölünmüştür. Operasyon sıralama, esas itibarıyla özelliklerin sıralamasına ve hangi özelliğin diğerinden

önce işlenmesi gerektiğini belirlemek için veri tabanındaki yardımcı kurallara bağlıdır.

Operasyon sırası belirlendikten sonra, her yüzey/özellik için uygulanacak operasyonlar, **sonuç çıkarma mekanizması** ve **kullanıcı arayüzü** yardımıyla seçilir. Bunun için de sistemde yer alan ilgili kurallar uygulanır. Parçanın özellikleri tanıdıktan sonra, sonuç çıkarma mekanizması ilgili kuralları uydurarak icra eder ve istenen geometriyi üretmek için gerekli işleme operasyonlarını belirler. Bilgi tabanında yeterli bilgi varsa, sonuç çıkarma mekanizması, özelliği işlemek için bir çözüm bulabilir. Operasyon seçimi için bilgi tabanı, her işlemin işlem kapasitesine bağlı olarak hazırlanmıştır.

Özelliklere uygulanacak operasyonun tipi belirlendikten sonra, bu operasyonu yerine getirecek olan kesici takımlar, ilgili program modülündeki kurallar işletilerek belirlenir. Kesici takımların seçimi, genellikle iş parçası malzemesi, sertlik, özelliklerin geometrik karakteristikleri ve seçilen işleme operasyonlarının tipine bağlıdır.

İşleme parametreleri hesaplandıktan sonra, sonuç çıkarma mekanizması, her operasyona göre uygun tezgahı seçmek için, tezgah seçimi ile ilgili kuralları bulur ve her operasyon için gerekli olan tezgahları otomatik olarak seçer. Takım tezgahı seçimi, parça boyutu, operasyon tipi, özellik tipi, yüzey pürüzlülüğü, sertlik gibi faktörlere bağlı olarak yerine getirilir. Takım tezgahları ve kesme takımları ile ilgili veri tabanları okunabilir Prolog olguları şeklinde hazırlanmıştır.

Sistemde bağlama işlemi, işlenecek özelliğin geometrik karakteristiklerine ve seçilen operasyonlara göre yerine getirilir. İşlem görecektir iş parçasını yerleştirmek ve tesbit etmek için **6 nokta (3-2-1) metodu**, Prolog cümlecikleri formundaki bağlama işlemi için ilgili kuralları icra etmek suretiyle uygulanır. Sistem ileriye planlama stratejisi işletir. Yani taslak parça modeli ile başlar ve bitmiş parça modeline ulaşıncaya kadar her özellik için operasyonları uygular. Taslak model geometrisini değiştirecek olan ilk operasyon için bağlama işlemi, taslak parçanın geometrisine bağlı olarak yapılır. Güvenilir kararlar vermek ve takip eden operasyon için uygun yerleştirme, destekleme ve tesbit yüzeylerini seçmek için ilk operasyondan sonra oluşturulan yeni araparça model, modellendirilmeli ve takip eden operasyon için hazır olmalıdır. Bu yüzden her operasyondan sonra işlenen bölge, parça modelinden kaldırılır ve yeni araparça model oluşturularak KYG'de temsil edilir.

İşlem planlama ve bağlama işlemine göre, iş parçasının durumunu doğru bir şekilde tanımlamada araparça modeller gereklidir. Bu yüzden taslak, bitmiş ve araparça modeller KYG'de temsil edilir ve veri tabanında depolanır.

Operasyonun tipine göre bağlama prosedürü yerine getirilirken destekleme, yerleştirme ve tesbit yüzeylerini belirlemek için, parçanın yüzeyleri hakkında yeterli bilgiye sahip olmak gerekir. Çünkü yerleştirme ve tesbit işlemi, yüzeylere bağlı olarak yapıldığından yüzeyler hakkındaki bilgi veri tabanında hazır olarak bulunmalıdır.

Program modülleri, modüler yapıda hazırlandığından, sistem, mevcut modüllerin performansını etkilemeden başka modülleri ilave etmek suretiyle genişletilebilme potansiyeline sahiptir.

Sistemin eksikliklerinden birisi KYG şeması ve araparça oluşturma modülü tarafından kullanılan özelliklerin sayısı ve tipinde yatar. Sistem, düzlem yüzeyler, kanallar, kademeler ve delikleri ele alır. Daha ileri araştırmalar eğik yüzey, havşa açma, delik büyütme, vida çekme, T-kanal, V-kanal gibi daha fazla özelliği kapsayacak şekilde genişletilmelidir.

İşlem planlama sistemi, referans yüzeyini, operasyonları, kesicileri ve takım tezgahlarını ve operasyonların sıralamasını otomatik olarak yapar. Ayrıca her operasyondan sonra araparça model, operasyona göre otomatik olarak oluşturulur. Sonuçlar kabul edilebilir olmasına rağmen, bunlar optimizasyon teknikleri kullanılarak iyileştirilebilir. Geliştirilen sistem, kullanıcı etkileşimli olmaktan ziyade, program yönlendirmeli olarak tasarlandığından, sistem tarafından elde edilen sonuçlara müdahale edilemez. Şayet sistemde böyle bir özelliğin olması arzu edilirse, sistemin kullanıcı etkileşimli olacak şekilde tasarlanması gerekir. Mevcut sistemde kullanıcı işlem planlama kartında herhangi bir değişiklik yapmak isterse, örneğin takım tezgahı, kesme takımı değiştirme veya operasyonların sırasını değiştirme gibi, işlem planlama kartlarını düzenlemek için bir metin editörü kullanmaya ihtiyaç duyar.

Kullanılan algoritmalar ve tekniklerin uygunluğunu denemek için tecrübeye dayanan planlama sistemi, prizmatik parçalara uygulanmaktadır. Üç parçaya ait modelleme ve planlama sonuçları örnek olarak Ek-D'de verilmiştir.



## 11. SONUÇ VE ÖNERİLER

Bu çalışmada, esnek, otomatik ve STEP standardı kullanan US yaklaşımına dayalı üretken BDİP sistemi geliştirilmiştir. Sistem, özellikle bu çalışma için geliştirilmiş olan KYG adlı özel bir açıklama şemasını içerir. Şema, yüzeyleri ve özellikleri temsil etmekle beraber, operasyon seçimi ve bağlama işlemi için gerekli olan, yüzeyler ve özellikler arasındaki açılar, içbükeylik-dışbükeylik ilişkisi gibi ilgili bilgiyi de içerir. Sistem T. Prolog kullanılarak Kişisel Bilgisayar (PC)'de uygulanmıştır. Geliştirilen sistem tarafından otomatik olarak yerine getirilebilen işlem planlama faaliyetleri aşağıda sıralanmıştır:

- STEP standartta 3-B BDT verileri girişini değerlendirme,
- Stok seçimi, taslak parça tasarımı ve taslak parça oluşturma,
- Bitmiş parça modeli oluşturma,
- Referans yüzey seçimi,
- Operasyon seçimi ve sıralama,
- İşleme parametrelerini hesaplama,
- Kesme takımı ve takım tezgahı seçme,
- Her operasyon için bağlama ve araparça model oluşturma .

Sonuçlar, US yaklaşımına dayalı sistemde, işlem planlama için istenen bilgiyi elde etmek suretiyle, prizmatik parçalar için işlem planlarını otomatik olarak üretmede tatminkar olduğunu göstermektedir.

Sistem, temel geometrik elemanlardan oluşan üç boyutlu prizmatik parçaları ele alır. Daha ileri seviyede yapılacak çalışmalar, karmaşık parçalar için genişletilebilir. Böyle bir sistemin dönel parçalar için de geliştirilmesi mümkündür. Geliştirilen sistemi BDÜ sistemlere bağlamak için, STEP formatındaki BDT verilerini otomatik olarak Prolog olguları haline dönüştüren bir dönüşüm programı ve sistemin çıktısından SD kodlarının üretilmesi gerekir. Bu yüzden mevcut sisteme, SD kodu oluşturma modülü eklenebilir. Araparça modellerin oluşturulmasından elde edilen parça koordinatları, SD kodu oluşturma sistemi için giriş verileri sağlar. Her operasyondan sonra üretilen araparça modelleri esas alınarak SD programları elde edilebilir.

Maliyet analizi ve işleme zamanı hesabı sistemde yer almamıştır. İşleme parametrelerinin hesabı, parçalar için maksimum üretim miktarı esasına göredir.

Sistemde, takımlama ve imalat operasyonları için ekonomik bilgi yer almamıştır. Bu nedenle işlem planlama prosedürünün ekonomisi değerlendirilmemiştir. Bu tür bilgiler ekonomik düşüncelerle hazırlanan sistemlere eklenebilir.

Bağlama prosedürü sadece, parçaların, destekleme, yerleştirme ve tesbit etme yüzeylerini belirlemeyi esas almıştır. Bağlama donanımının tasarımı dikkate alınmamıştır. Sistemi daha etkin hale getirmek için bağlama donanım tasarım modülü eklenebilir.

Araparça modelleri, KYG'de otomatik olarak oluşturulur. Modellerin grafiksel çıktısını görüntülemek için, bir BDT sisteme parça verileri manüel olarak girilmelidir. Modelin mevcut durumunun hızlı bir grafiksel gösterilimini üretmek için bu işlem otomatize edilebilir.

Sistemin geliştirilmesinde kullanılan T. Prolog, programın işletilmesi sürecinde her aşamada elde edilen yeni veriler için, tip tanımlamasını zorunlu kıldığından bu durum, hem program modüllerinin büyümesine, hem de sistemin geliştirilmesinde bazı güçlüklerle neden olmaktadır. Şayet böyle bir sistemi geliştirmede bir Prolog versiyonu kullanılacaksa, Standart Prolog türlerinden birinin tercih edilmesi daha uygun olacaktır.



## 12. KAYNAKLAR

- [1] Kochan, A. and Cowan, D., "Implementing CIM-Computer Integrated Manufacturing, IFS Publications Ltd., UK, 1986.
- [2] Goetsch, D.L., "Fundamentals of CIM Technology", Delmar Publishers Inc., USA, 1988.
- [3] Chang, T.C., Anderson, D.C. and Mitchell, O.R., "QTC- An Integrated Design/Manufacturing/Inspection System For Prismatic Parts", Proceedings of ASME Computers in Engineering Conference, Vol. 1, pp. 417-426, 1988.
- [4] "Machining Data Handbook", Metcut Research Associates Inc. Machinability Data Center, Vol. 2, USA, 1980.
- [5] Wang, H.P. and Wysk, R.A., "Applications of Microcomputers in Automated Process Planning", CASA/SME Journal of Manufacturing Systems, Vol. 5, No. 2, 1985.
- [6] Besant, C.B., Lui, C.W.K., "Computer Aided Design and Manufacture", Elliswood Limited, USA, 1986.
- [7] Chang, T.C., Wysk, R.A. and Wang, H.P., "Computer Aided Manufacturing", Prentice Hall, USA, 1991.
- [8] Chang, T.C., "Expert Process Planning for Manufacturing", Addison-Wesley Publishing Company, USA, 1990.
- [9] Alting, L. and Lenau. T., "Integration of Engineering Functions in the CIM Philosophy", 15th International Symposium on Automotive Technology & Automation, Switzerland, pp. 1-19, 1986.
- [10] Ssemakula, M.E., Gill, J.S., "CAD/CAPP Integration Using IGES", AME-Advanced Manufacturing Engineering, Vol 1, No 5, pp. 264-270, 1989.
- [11] Smith, P., "CAD/CAM Data Exchange", CIM Review The Journal of Computer Integrated Manufacturing Management, Vol 6, No 2, pp. 31-33, Winter 1990.
- [12] STEP-Standard for the Exchange of Product Model Data, The First Working Draft of STEP Version 1.0, ISO TC 184/SC4/ WG 1, pp. 1-546, 1988.
- [13] Nolen, J., "Computer Automated Process Planning for World-Class Manufacturing", James Nole & Company, USA, 1989.
- [14] Wang, M. and Waldron, M.B., "A Prototype Integrated Feature Based Design and Expert Process Planning System for Rotational Parts", Proceedings of ASME International Computers in Conference, Boston, Vol. 1, pp. 33-51, 1990.

- [15] Lenau, T. and Mu, L., "Features in Integrated Modelling of Products and Their Production", Symposium on Feature Based Approaches to Design and Process Planning, Loughborough University, pp.3-9,1991.
- [16] Joshi S. and Chang, T.C., "Graph-Based Heuristics for Recognition of Machined Features From a 3D Solid Model", Computer Aided Design, Vol. 20, No. 2, 58-66, 1988.
- [17] Joshi, S. and Chang, T., "Feature Extraction and Feature Based Design Approaches in the Development of Design Interface for Process Planning", Journal of Intelligent Manufacturing, London, Vol. 1, No. 1, pp 1-15, 1990.
- [18] Chen, C.S., "Developing a Feature Based Knowledge System for CAD/CAM Integration", Computers Ind. Eng., Vol 15, Nos 1-4, pp.34-40,1988.
- [19] Joseph, A.T. and Davies, B.J., "Elicitation of Process Planning Knowledge in a Manufacturing Environment", The International Journal of Advanced Manufacturing Technology, Vol. 6, pp. 16-34, 1991.
- [20] Altung, L. Zhang, H., "Computer Aided Process Planning: the State-of-the-art Survey", International Journal of Production Research, Vol. 27, No.4, pp. 553-585, 1989.
- [21] Iwata, K., "Representation of Know-How and Its Application of Machining Reference Surface in Computer Planning", Annals of the CIRP, Vol. 35/ 1, pp. 321-324,1986.
- [22] Rustom, E.A. and Mileham, A.R., "The Development of a Generative Computer Aided Process Planning System for Prismatic Parts", Advances in Manufacturing Technology 4, Proceedings of the 5th National Conference on Production Research, Huddersfield Polytechnic, pp. 259- 263, 1989.
- [23] Groppetti, R. and Semeraro, Q., "CAPP-Computer Aided Process Planning Using Relational Data Base", ISATA'86- International Symposium on Automotive Technology and Automation, pp. 1-30, 1986.
- [24] Tonshoff, H.K., Beckendorff, U., Hellberg, K. and Anders, N., "The Future CAP Lies Within an Integrated Expert System", Second International Conference Computer Aided Production Engineering Edinburgh, pp.133- 140,1987.

- [25] Joseph, A.T. and Davies, B.J., " Elicitation of Process- Planning Knowledge in a Manufacturing Environment", Int. J. Adv. Manuf. Tech. ,Vol. 6, pp. 16-34, Springer-Verlag, London, 1991.
- [26] S. Bandyopadhyay, S.P. Dutta, D. Meloche and Rana, S.P., "Component Description for Knowledge Based Process Planning", the Int. J. of Adv. Manuf. Tech., Vol.1(3), pp. 55-74, IFS Publication Ltd. Kanada, 1986.
- [27] Srihari, K. and Greene, T.J., "Alternate Routings in CAPP Implementation in an FMS", Computers Ind. Engin., Vol. 15, No. 1-4, pp. 41-50, 1988.
- [28] Luong, L.H.S., "Process Planning Via Computer-Assisted Classification and Coding" Int. J. Adv. Manuf. Tech. ,Vol. 4, pp. 311-320, Springer-Verlag, London, 1989.
- [29] Kusiak, A., "The Generalized Group Technology Concept", Int. Journal of Production Research, Vol. 25, No.4, pp. 561-589,1987.
- [30] Lee, I.B.H., Lim, B.S. and Nee, A.Y.C., " Knowledge-Based Process Planning System for the Manufacture of Prograsive Dies ", Int. J. Prod. Res. Vol. 31ç. No.2, pp. 251-278, 1993.
- [31] Li, Y.N., Wang, X.L., Bian, W.Y. and Xia, Z.Q., "A New Method to Group Parts and Establish GT Cells - Production Flow Analysis and Classification Coding Cluster Analysis", ISITA'86 15th International Symposium on Automotive Tecknology & Automation, Switzerland, pp. 47-54, 1986.
- [32] Ali, D.L., Ali, K.S. and Ali, A.L., "Neural Networks: A Feasible Approach for Process Planning Automation", Proceedings of the European Simulation Multiconference, Nice, France, pp. 107-111,1988.
- [33] Joshi, S., Chang, T.C. and Liu, C.R., "Process Planning Formalisation in an All Framework", Artificial Intelligence, Vol. 1, No. 1, 1986.
- [34] Steudel, II.J., "Computer Aided Process Planning: Past, Persent and Future" CAPP-Computer Aided Process Planning , SME, Michigan 48121, pp.3-14,1984.
- [35] Iwata, K., "Knowledge Based Computer Aided Process Planning", Intelligent Manufacturing Systems II, pp. 3-25, 1988.
- [36] Kusiak, A.,"Knowledge Based Systems in Manufacturing", Taylor & Francis, London, pp.111-136,1989.

- [37] Freedman, R.S. and Sylvester, W.A., "The Evolution of An Expert System for Process Planning", 1st Int. Symposium MCLEAN VA IEEE, pp.328- 34, Oct 1985.
- [38] Domazet, D.S., "Concurrent Design and Process Planning of Rotational Parts", Annals of the CIRP, Vol. 41/1, pp. 181-184, 1992.
- [39] Kusiak, A., "Intelligent Manufacturing Systems", Prentice-Hall, Inc., New Jersey, 1990.
- [40] Isik, I. and Mileham, A. R., " The Design of a Generative Computer Aided Process Planning System-For Turned Components in a Conventional Machine Tool Environment", Advances in Manufacturing Technology 3- Proceedings of the 4th. National Conference on Production Research, Sheffield City Polytechnic, pp. 315- 320, 1988.
- [41] Opitz, H., "A Classification System to Describe Workpieces", Edited by R.A. Taylor and W. R. Mc Connell, Pergamon Press, 1970.
- [42] Groover, M.P. and Zimmers, E.W., "CAD/CAM Computer Aided Design and Manufacturing", Prentice Hall, India, 1985.
- [43] Desai, V.S. and Pande, S.S., "GFM-An Interactive Feature Modeller for CAPP or Rotational Components", Computer Aided Engineering Journal, pp. 17-221, 1991.
- [44] Husbands, P., Mill, F.G. and Warrington, S.W., "Representation of Components for Automated Process Planning", International Conference on Computer Aided Production Engineering, Edinburgh, pp. 359-360, 1987.
- [45] Kyprianou, L.K., "Shape Classification in Computer Aided Design", PhD Thesis, University of Cambridge UK, 1983.
- [46] Choi, B.K., Barash, M.M. and Anderson, D.C., "Automatic Recognition of Machined Surfaces From 3D Solid Model", Computer Aided Design, Vol.16, No. 2, pp.81-86, 1984.
- [47] Liu, C.R. and Srinivasan, R., "Generative Process Planning Using Syntactic Pattern Recognition", Computers in Mechanical Engineering, pp.63-66, March 1984.
- [48] Henderson, M.R., "Extraction of Feature Information From Three Dimensional CAD Data", PhD Thesis, Purdue University, Indiana, USA, 1984.
- [49] Houtzeel, A., "The MICLASS System", Proceedings of CAM-I's Executive Seminar - Coding, Classification and Group Technology for Automated Planning, CAM-I Inc., Arlington, Texas, USA, 1976.

- [50] Link, C.H., "CAPP, CAM-I Automated Process Planning System", Proceedings of the 13th NC Society Annual Meeting & Technical Conference, Cincinnati, Ohio, 1976.
- [51] Wysk, R.A., "An Automated Process Planning and Selection Program: APPAS" PhD Thesis, Purdue University, West Lafayette, Indiana, USA, 1977.
- [52] Eversheim, W., Fuchs, H. and Zons, K.H., "Automatic Process Planning With Regard to Production by Application of the System AUTAP for Control Problems", Computer Graphics in Manufacturing Systems, 12th CIRP International Seminar in Manufacturing Systems, pp. 85-87, 1980.
- [53] Eskicioglu, H. and Davies, B.J., "An Interactive Process Planning System for Prismatic Parts - ICAP", Annals of the CIRP, Vol. 32/ 1, 1983.
- [54] Sack, J.C.F., "CAM-I's Experimental Planning System XPS-1", Autofact 5 Conference Proceedings, Detroit, Michigan, USA, 14-17 November 1983.
- [55] Phillips, R.H., Arunthavanathan Zhou, X.D., "MICROPLAN - A
- [56] SCICON Ltd., "Process Planning A Route to Success", Machinery and Production Engineering, Vol. 2, December 1987
- [57] Zhao, L. and Kops, L., "An Integrated CAPP and Scheduling System", 15th NAMR Conference Proceedings, pp. 552-557, 1987.
- [58] Li, J., Han, C. and Ham, I., "CORE-CAPP - A Company Oriented Semi Generative Computer Automated Process Planning System", 19th CIRP International Seminar on Manufacturing Systems - Computer Aided Process Planning, Penn. State, USA, June 1987.
- [59] Tulkof J., "Process Planning: A Historical Review and Future Prospects", 19th. CIRP International Seminar on Manufacturing Systems- Computer Aided Process Planning, Pen. State, USA, June 1987.
- [60] Wang, H.P., and Wysk, R.A., "A Knowledge Based Approach for Automated Process Planning", International Journal of Production Research, Vol. 26, No. 6, pp. 999-1014, 1988.
- [61] Yeh, C. H. and Fischer, G.W., "A Structured Approach to the Planning of Machining Operation for Rotational Parts Based on Computer Integration of Standard Design and Process Data", The International Journal of Advanced Manufacturing Technology, Vol., pp.285-298. 1991.

- [62] Ngoi, B.K.A., "Computer-Aided Process Planning for Turned Components", *Journal of the Institution of Engineers*, Vol. 33, No.4, pp. 39-44, Singapore, 1993.
- [63] Descotte, Y. and Latombe, J.C., "GARI: A Problem Solver that Plans to
- [64] Matsuhima, K., Okada, N. and Sata, T., "The Integration of CAD and CAM by Application of Artificial Intelligence Techniques", *Annals of the CIRP*, Vol. 31/1, pp. 329-332, 1982.
- [65] Chang, T.C. and Wysk, R.A., "Integrating CAD and CAM Through Automated Process Planning", *International Journal of Production Research*, Vol.22, No.5, pp. 877-894, 1984.
- [66] Barkocy, B.N., Zdeblick, W.J., "A KBS for Mach. Operation Planning", *Tech. Paper MS84-716, Proc. of AUTOFACT 6 Conf.*, 1-4, Anaheim, Calif., 1984.
- [67] Zhang, K.F., Wright, A.J. and Davies, B.J., " A Feature-Recognition Knowledge Base for Process Planning of Rotational Mechanical Components", *Int. J. Adv. Manuf. Tech.* , Vol. 4, pp. 13-25, Springer-Verlag, London, 1989.
- [68] Nau,D.S. and Chang, T.C., "A Knowledge Based Approach to Process Planning", *Proceedings of the Symposium on Computer Aided Intelligent Process The Winter Annual Meeting of ASME*, Miami Beach, Florida, pp. 65-72, November 17-22 1985.
- [69] Iwata, K., Sugimura, N., "An integrated CAD/CAPP System with Know-Hows on Machining Accuracies of Parts", *Computer-Aided/Intelligent Procces*, 1985.
- [70] Mouleeswaran, C.B., and Fischer, H.G., " A Knowledge Based Environment For Process Planning", *Proceedings of 1. Conference on Applications of AI to Engineering Problems*, Southampton, England, pp. 1013-1027, 1986.
- [71] Allahverdi, N.M., Yaldız, S. ve Ünüvar, A. "Endüstride Uzman Sistem Uygulamaları", 2. Endüstriyel OTOMASYON'95 Sempozyumu, 30 Mart- 31 Mart 1995, İstanbul, Bildiriler Kitabı, sayfa. 75-86, 1995.
- [72] Berenji, H. R., Khosnevis B., "Use of Artificial Intelligence in Automated Procces Planning", *Computers in Mechanical Engineering*, September, pp. 47-55., 1986.
- [73] Lenau, T. and Alting, L., "XPLAN- An Expert Process Planning System", *Second Int. Expert Systems Conference*, London, 30 Sept., 2 October, 1986.
- [74] Kahorami, M.T., Mc.Millan, C.R. and Mowforth, P., "A Generative Integrated Process Planning System (GIPPS VI. I):Unexpected Material



- Benefits"; *Advances In Manufacturing Technology 3*, Proceedings of the Fourth National Conference on Production Research, Sheffield. City Polytechnic, pp. 163-67, Sept 1988.
- [75] Milacic, V.R. and Urosevic, M., "SAPT-Knowledge Based CAPP System", *Robotics & Computer Integrated Manufacturing*, Vol. 4, No. 1/2, pp. 69-76, 1988.
- [76] Lee, H. K., Lee, J.W., and Lee, J.M., "Pattern Recognition and Process Planning Prismatic Workpiece by Knowledge Based Approach", *Annals of the CIRP* Vol. 38/1, Seoul, 1989.
- [77] Saygin C., "A Rule Based Approach in Sequencing Machining Operations for Rotational Components" Yüksek Lisans Tezi, O.D.T.Ü. Ankara, 1992.
- [78] Gülesin M., "An Intelligence Knowledge Based Process Planning and Fixturing System Using The Step Standard" PhD Thesis, Coventry University, England, July, 1993.
- [79] Kar, A., "An Expert Consultant for Manufacturing Process Selection", *Proceedings of the 1986 IEEE International Conference on Systems, Man., Sysb.*, New York, USA, pp. 1164-8, 1986.
- [80] Jackson, P., "Introduction to Expert Systems", Second Edition, Addison-Wesley Publishing Company, UK, 1990.
- [81] Davies, B.J. and Darbyshire, I.L., "The Use of Expert Systems in Process Planning", *Annals of the CIRP*, Vol. 33/1, pp. 303-306, 1984.
- [82] Feigenbaum, E.A., "Knowledge Engineering in the 1980's", Dept. of Computer Science, Stanford University, Stanford, CA., 1982.
- [83] Riley, G., Giarratano, J., "Expert Systems", FWS-KENT Publishing Company, Boston, USA, 1989.
- [84] Milacic, V.R., "How to Build Expert Systems", *Annals of the CIRP*, Vol. 35/2, pp. 445-450, 1986.
- [85] Alder, G.M., McGeough, J.A., Hon, K.K.B. and Ismail, H.S., "Selection of Machining Processes by Intelligent Knowledge Based Systems", *International Conference on Computer Aided Production Engineering*, Edinburgh, pp. 61-66, 1986.
- [86] Bratko, "PROLOG Programming for Artificial Intelligence", Addison-Wesley Publishing Company, UK, 1987.
- [87] Chang, T.C. and Wysk, R.A., "An Introduction to Automated Process Planning Systems", Prentice Hall Inc., USA, pp. 90-109, 1985.
- [88] Alty, J.I. and Coombs, M.J., "Expert Systems Concept and Examples", NCC Publication, pp. 60-76, England, 1984.



- [89] Stillings, N. A., "Cognitive Science: An Introduction", The MIT Press, USA, 1987.
- [90] Quillian, M.R., "Semantic Memory", Semantic Information Processing, ed.by Marvin Minsky, The MIT Press, pp 227-270, 1068, 1986.
- [91] Winston, P.,H., " AI ", Second Edition, Addison-Wesley Inc. 1984.
- [92] Staugaard, A.C.Jr., "Robotics and AI: An Introduction to Applied Machine Intelligence", Prentice-Hall, 1987.
- [93] Shank, R., C., Abelson, R., P., "Scripts, Plans, Goals and Understanding", Lawrence Erlbaum, 1977.
- [94] Brachman, R.,J., "I lied about the trees", AI Magazine, Vol.6, (3), pp 80-93, 1985.
- [95] Durkin, J., Expert Systems- Design and Development, Macmillan Publishing Company, 866 Third Avenue, New York 10022, 1993.
- [96] Schildt, H., "Advanced Turbo Prolog: Version 1.1", McGraw Hill, USA,1987.
- [97] Rooney, J. and Steadman, P., "Principles of Computer Aided Design", Open University, Pitman Publishing, London, 1987.
- [98] Gornes, A.J.P. and Teixeira, J.C.G., "Form Feature Modelling in a Hybrid CSG/B-rep Scheme", Computers and Graphics, Vol. 15, No.2, 217-229, 1991.
- [99] Wilson, P.R., "A Short History of CAD Data Transfer Standards", IEEE Computer Graphics & Applications, Vol. 7, No. 6, pp. 64-67, 1987.
- [100] Perng, D., Chen, Z. and Li, R., "Automatic 3D Machining Feature Extraction from 3D CSG Solid Input", Computer Aided Design, Vol. 22, No. 5, pp. 285-295,1990.
- [101] Lee, Y.C. and Jea, K.F.J., "A New CSG Tree Construction Algorithm for Feature Representation", Proceedings of ASME, Computers in Engineering Conference, Vol.1, pp. 521-528, 1988.
- [102] Sakurai, II. and Gossard, D.C., "Recognizing Shape Features in Solid Models" IEEE Computer Graphics & Applications, pp. 22-32,1990.
- [103] Cutkosky, M.R., Tenenbaum, J.M. and Muller, D., "Features in Process-Based, Design", Proceedings of ASME, Computers in Engineering Conference, Vol. I, pp. 557-562, 1988.
- [104] Smith, B., Rinaudot, G.R., Reed, K.A., Wright, T., "The Initial Graphics Exchange Specification (IGES)", Version 4.0, US Department of Commerce, National Bureau of Standards, National Engineering Lab.NBSIR 88-3813, Gaithersburg, MD 20899, 1988.

- [105] Stauffer, R.N., "IGES and Beyond", CIM Technology Magazine, CASA/SME'S Magazine of Coimputers in Design and Manufacturing, pp. 20-21, 1985.
- [106] Wilson, P.R., "PDES STEPs Forward", IEEE Computer Graphics & Applications, Vol. 9, No. 2, pp. 79-80, 1989.
- [107] Eskicioglu, H. and Cebeci, H.M., "A Dialogue System for Process Planning of Prismatic Parts", Second International Conference on Computer Aided Production Engineering, Edinburgh, pp.291-295, April 1987.
- [108] Metal's Handbook, 9th Edition, Vol. 16, Machining, ASM International Handbook Committee, USA, pp. 1-630., 1989.
- [109] Lindberg, R.A., "Processes and Materials of Manufacture", Allyn and Bacon, USA,1990.
- [110] Ferreira, P.M. and Liu, C.R., "Generation of Workpiece Orientations for Machining Using Rule Based System", Robotics and Computer Integrated Manufacturing, Vol.4, No. 3/4, pp. 545-555, 1988.
- [111] Boothroyd, G. and Knight, W.A., "Fundamentals of Machining and Machine Tools", Marcel Dekker. , New York, 1989.
- [112] Keyser, L.A., Leach, J.L. and Schrader, G.F., "Manufacturing Processes and Matteredials for Engineers", Prentice Hall Inc., USA, 1985.
- [113] Wang, H.P. and Li, J.K., "Computer Aided Process Planning ", Elsevier, USA, 1991.
- [114] Trappey, J.C. and Liu, C. R., "A Literature Survey of Fixture Design Automation", The Int. Journal of Advanced Manufacturing Technology, pp. 240-255, 1990.
- [115] Pollack, H. W. , ".Tools Design", Prentice Hall, USA, 1988.
- [116] Dong, X., Devries, W.R. and Wozny, M.J., "Feature Based Reasoning in Fixture Design", Annals of the: CIRP, Vol. 40/1, pp. 111-114, 1991.

**EK-A**  
**EXPRESS DİLİNİN BAZI ELEMANLARI**  
**VE**  
**WIRTH SYNTAX NOTATION (WSN)**

EXPRESS, Ürün Verileri Değişirme Standardının (PDES/STEP-ISO-10303) bilgi ihtiyaçlarını tanımlamak için kullanılan resmi bilgi modelleme diline verilen addır. EXPRESS , bir programlama dili olmayıp açık ve kısaca ifade edilen verilerdeki kısıtlarla belirli obje tanımlamalarına izin veren dil elemanlarından meydana gelir. EXPRESS dili, girdi / çıktı (input/ output) ve bilgi işleme gibi dil elemanlarını içermez. Bu dil genel olarak aşağıdaki tasarım amaçlarını esas alır.

- STEP'in karmaşık ve geniş kapsamlı olması, çözümlemenin insandan ziyade, bilgisayarlarla yapılmasını gerektirir.

- Bu dil, STEP tarafından tanımlanan farklı objelerin bölümler halinde tanımlanmasına imkan verecek şekilde tasarlanmıştır. Burada hazırlanan şema bölümlenmeyi ve iç iletişimi esas alır.

- EXPRESS dili ilgilenilen varlıkların tanımlanmasını amaçlar. Varlıkların tanımlanması veriler ve davranış olarak ifade edilir. Veriler, kısıtlarla temsil edilen davranışları ve açıklanabilen bir varlıkla özellikleri temsil eder.

- Dil, otomatik ve belirli bir tarzda statik dosya değiştirme gibi, bazı imalat özelliklerini yerine getirir.

EXPRESS'te varlıklar, kullanım ve anlama açısından önemli kabul edilen nitelik veya karakteristiklerine göre tanımlanır. Bu nitelikler integer gibi basit bir veri tipi veya başka bir varlık tipi şeklinde temsil edilebilir. Örnek olarak geometrik bir nokta, üç koordinatla; isimler ise, bir varlığın tanımlanmasına yardımcı olan niteliklerle verilir. Tanımlanan varlık ve onu tanımlayan nitelikler arasında bir ilişki kurulur. Benzer bir durum niteliklerle bunların temsilleri arasında da mevcuttur.

Bölüm 5.4'te açıklandığı gibi, STEP metodolojisi **uygulama**, **mantıksal** ve **fiziksel katman** olmak üzere 3 bölümden meydana gelir. Veri modelleri mantıksal katmanda bulunur. EXPRESS dili Nitelikler, Sabitler, Enumartion tipindeki elemanlar, Varlıklar, Fonksiyonlar, Parametreler, Prosedürler, Kurallar, Şemalar, Tipler ve Değişkenler gibi... çeşitli veri modellerini açık veya kapalı bir şekilde deklere etmek için kullanılır. Daha sonra açıklanan veri modelleri fiziksel katmanda bulunan fiziksel dosyaya yazılır. Fiziksel dosyalar, bilgisayar tarafından işlenebilen Wirth Syntax Notation

(WSN)'e göre yapılır. Aşağıdaki bölümlerde EXPRESS dilinin bazı elemanları ve WSN açıklanmıştır.

## 1. EXPRESS DİLİNİN BAZI ELEMANLARI

**Karakter seti:** EXPRESS dili belli kurallara uyan karakterler dizisi olarak yazılır. Karakterler **Token**' ları şekillendirmek için bir araya getirilir. EXPRESS'te Tokenlar semboller, anahtar kelimeler veya belirleyicilerdir. EXPRESS'te kullanılan karakter seti, İngilizce alfabenin büyük ve küçük harflerinden ve arapca sayılardan oluşur. Bunlardan başka bazı özel karakterler de kullanılır.

**Semboller,** EXPRESS'te özel bir anlamı olan karakterler veya karakterler grubudur. EXPRESS'te kullanılan semboller ayırıcılar (separators) veya operatörler olarak kullanılır.

**Ayırıcı,** bitişik elemanları yorumlamaksızın birbirinden ayırır. Ayırıcılara ait bazı örnekler ".", ",", ";", ":" gibi verilebilir.

**Operatörler** ise, bir veya bir çift operandı icra eden çeşitli faaliyetleri gösterir. Çeşitli operatör örnekleri "\*", "+", "-", "=", "/", "<", ">", "\*\*", "<=", "<>", ">=" şeklinde verilebilir.

**Anahtar kelimeler,** EXPRESS'te okumayı ve anlamayı kolaylaştırmak bakımından dile semantik ilave etmek için kullanılır. EXPRESS'te kullanılan anahtar kelimelerden bazıları "end\_entity, end\_function, end\_if, end\_local, end\_model, end\_procedure, end\_repeat, end\_rule" şeklinde verilebilir.

**Belirleyiciler,** şemada deklere edilen elemanlara verilen isimlerdir. Belirleyicinin ilk karakteri daima bir harf olmalıdır. Diğer karakterler harfler, sayılar veya alt çizgi karakterlerinin kombinezonu olabilir. Belirleyiciler rezerv kelimelerden herhangi birisiyle aynı olamaz.

**Rezerv kelimeler,** EXPRESS'te rezerv kelimeler, anahtar kelimeler veya standart sabitler, fonksiyonlar veya prosedürleri yapılandıran isimlerdir. Rezerv kelimelerin hiç birisi belirleyici olarak kullanılamaz. Rezerv kelimeler büyük ve küçük harflerin kombinezonu, alt çizgi "\_" karakter ve soru işareti "?" kullanılarak yazılabilir. EXPRESS'in rezerv kelimeleri Tablo A.1'de gösterilmiştir.

Tablo A.1. Standart EXPRESS Kelimeleri

ABSTRACT	END_CONSTANT	IN	REMOVE
AGGREGATE	END_CONTEXT	INTEGER	REPEAT
ALIAS	END_ENTITY	INVERSE	RETURN
AND	END_FUNCTION	LENGTH	ROLESOF
ANDOR	END_IF	LIKE	RULE
ARRAY	END_LOCAL	LIST	SCHEMA
AS	END_MODEL	LOINDEX	SELECT
BAG	END_PROCEDURE	LOCAL	SELF
BEGIN	END_REPEAT	LOGICAL	SET
BINARY	END_RULE	MOD	SKIP
BLENGTH	END_SCHEMA	MODEL	STRING
BOOLEAN	END_TYPE	NOT	SUBTYPE
BY	ENTITY	NUMBER	SUPERTYPE
CASE	ENUMERATION	NVL	THEN
CONST_E	ESCAPE	OF	TO
CONSTANT	FIXED	ONEOF	TYPE
CONTEXT	FOR	OPTIONAL	UNIQUE
DERIVE	FORMAT	OR	UNTIL
DIV	FROM	OTHERWISE	USEDIN
ELSE	FUNCTION	PROCEDURE	USE
END	GENERIC	QUERY	VAR
END_ALIAS	HIINDEX	REAL	WHERE
END_CASE	IF	REFERENCE	WHILE
			XOR
<b>Fonksiyonları icra eden kelimeler</b>			
ABS	EXISTS	LOG10	SQRT
ACOS	EXP	LOG2	TAN
ASIN	HIBOUND	ODD	TYPEOF
ATAN	LOBOUND	SIN	VALUE
COS	LOG	SIZEOF	
<b>Prosedürleri icra eden kelimeler</b>			
INSERT	NULLIFY	VIOLATION	
<b>Sabitleri icra eden kelimeler</b>			
#	FALSE	PI	TRUE
			UNKNOWN

## 1.1. İcra Edilebilir Deyimler

İcra edilebilir deyimler, sadece fonksiyon, prosedür veya kurallar içinde bulunur ve fonksiyon, prosedür veya kural için lokal değişken rolü oynar. Bunlar kısıtların tanımlanmasını destekler ve gerekli mantık ve eylemleri tanımlamak için kullanılır. İcra edilebilir deyimlerden bazıları aşağıda kısaca açıklanmıştır.

**Null deyimi**, sadece bir noktalı virgül (;) içeren icra edilebilir deyim, boş deyim olarak adlandırılır.

```
IF A = 13 THEN
; ----- >Bu boş bir ifadedir
ELSE
  B: = 5;
END_IF;
```

**Alias deyimi**, bu deyim, varlık örneklerinin niteliklerini kapsayan ifadelerin yazımını basitleştiren değişken niteliğin kısa bir şeklini verir. Bir ifade **alias tanımlayıcı** tarafından nitelenen belireyiciler, tamamen nitelenmiş olarak kabul edilir. Örnek;

```
ENTITY line;
  start_point,
  end_point : point;
END_ENTITY;

LOCAL
  this_line : line;
END_LOCAL;

ALIAS start FOR this_line.start_point;
  distance := start.x**2 + start.y**2 + start. **2 ;
END_ALIAS;
```

**Assignment deyimi**, lokal değişkene veya parametreye bir değer atamak için kullanılır. Değişkene atanan değer ifadesinin veri tipi, değişken veya parametre ile uyumlu olmalıdır.

```

LOCAL
  A: INTEGER;
  B: INTEGER;
  C: INTEGER;
END_LOCAL;
  A = 1;
  B = 2;
  C = A+B;

```

Burada LOCAL, lokal deęişken tanımlamayı gösterir.

**Birleşik deyim**, BEGIN ve END ile sınırlandırılmış deyimlerin sıralamasıdır. Birleşik deyim, bir tek deyim gibi rol oynar. Örnek;

```

BEGIN
  A = A + 1
  IF A > 100 THEN
    A := 0;
  END_IF;
END;

```

**Escape deyimi**, icra edilebilir bir deyim işlem görürken **escape** deyimi prosedürü **repeat** deyimini izleyen satıra gönderir. Bu deyim kullanılması, prosedürde sonsuz tekrarı sona erdirebilecek yollardan birisidir. Örnek;

```

...
REPEAT UNTIL (A = 1);
...
IF (A < 0 ) THEN
  ESCAPE;-----> icra edildiğinde, kontrol END_REPEAT'den sonraki
                    ifadeye geçer
END_IF;
...
END_REPEAT;
...

```

**If-Then-Else deyimi**, mantıksal bir ifadeyi esas alan ifadelerin şartlı icrasına müsaade eden bir deyimdir. Mantıksal ifade doğru olarak değerlendirildiğinde, takip eden **then** ifadesi; mantıksal ifade yanlış veya bilinmeyen olarak değerlendirildiğinde ise, **else** deyimi icra edilir. İfade yanlış



veya bilinmeyen olarak değerlendirilirse ve **else** ihmal edilirse, kontrol takip eden ifadeye geçer. Örnek;

```

LOCAL
  A: INTEGER;
END_LOCAL;
  A = 5;
IF A < 10 THEN
  A = A + 1; ----- > Bu ifade icra edilecek
ELSE
  A = A - 1;
END_IF;

```

**While control**, kontrol ifadesi doğru olduğu sürece, tekrar ifadesi bloğunun icrasını başlatır ve sürdürür. Her iterasyondan önce ifade değerlendirilir. Şayet ifade de while control varsa ve ifade başlangıçta hatalı ise blok icra edilmez.

**Until control**, kontrol ifadesi doğru olarak değerlendirilene kadar, tekrar ifadesinin icrasını devam ettirir. İfade her iterasyondan sonra değerlendirilir. Until Kontrol, sadece bir durumu kontrol ederse, en azından bir iterasyon daha icra edilir. Until Kontrol iterasyon sonunda yeniden değerlendirilir.

**Return deyimi**, fonksiyon veya prosedürün icrasını sona erdirir. Yardım istenen noktaya bir fonksiyonun dönüşü, dönüşümün sonucu olan bir ifadeyi içermelidir. Bir prosedürün geri dönüşü, bir ifade kapsamalıdır. Örnek;

```

....
RETURN (50);          (* bir fonksiyondan *)
....
RETURN (* çalışma_noktası *)
....
RETURN;              (* bir prosedürden *)

```

**Skip deyimi**, prosedürü bloğun sonuna atlatarak işlemin devamını sağlar. Blok, bir tekrar bloğu ise, sınır şartları test edilir. Blok, repeat kontrollerinin değerlendirilmesine bağlı olarak tekrar edilir veya sona erdirilir.

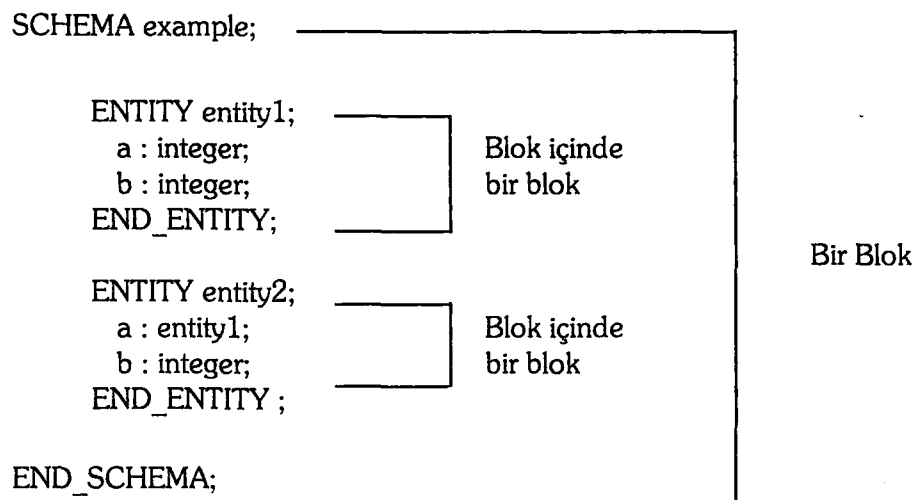
```

....
REPEAT UNTIL (A = 1);
...
IF (A < 0) THEN
SKIP;          (* icra edildiğinde, kontrol END_REPEAT ifadesine geçer
                burada son durum değerlendirilir *)
END_IF;
.....
END_REPEAT;
....

```

## 1.2. Dekleratif İfadeler ve Bloklar

**İfade**, EXPRESS gramerine uyan Token'lar sıralamasıdır. İfadeler bir noktalı virgül (;) karakteriyle sona erer. Kapalı olarak açıklanan objenin kesin formları, bu deklarasyonla ilgili **block** olarak adlandırılan yeni bir alanı tanımlar. Bloklar, ifadelerin belirli bir düzende sıralanmasından meydana gelir. Bir blokta, bloğun tipine bağlı olarak sadece belli tipte bir deklarasyon yapılabilir (Şekil A.1). Blok içerisinde aynı formata sahip birden fazla belirleyici olmamalıdır. En dıştaki blok, bir şema bloğu olmalıdır. Bir şema ise bir çok şemayı kapsayan bütün deklarasyonları içerir. Her deklarasyon formu için dil, deklarasyonun **scope**'u olarak adlandırılan metnin belirli bir bölümünü tanımlar. Bütün deklarasyonlar, tanımlanan objeyle bir belirleyiciyi ilişkilendirir.



Şekil A.1. Şema içerisindeki bloklar

Şemada belirli tip bir eleman, herhangi bir yerde bir belirleyicinin deklere edilmesiyle oluşturulur. Oluşturulan bu eleman tekrar kullanıldığında, deklere edildiği alan içerisinde gösterilmelidir. Varlıklar, Prosedürler ve Fonksiyonlar ")\*" ile başlar ve "(\*" sona erer. Tipik bir varlık örneği aşağıdaki şemada gösterilmiştir.

```
)*
ENTITY point;
  x: real;
  y: real;
  z: real;
END_ENTITY
(*
```

**Lokal kural**, bireysel niteliklerin değerini veya birleşik nitelikleri kısıtlayan şartları belirlemek için kullanılır. Bütün lokal kurallar "WHERE" anahtar kelimesini takip etmelidir. Örneğin birim vektörün uzunluğunun "1" olması gerekir. Bu kısıt şu şekilde açıklanabilir;

```
ENTITY unit_vector;
  a: real;
  b: real;
  c: real;
WHERE
  a** 2 + b** 2 + c** 2 = 1.0;
END_ENTITY
```

Lokal kurallar, bazen isteğe bağlı nitelikleri ele alır; niteliğin değeri olmadığı zaman, isteğe bağlı niteliği içeren kuralın bu parçası, o nitelik yokmuş gibi değerlendirilir. Yukardaki örnek değiştirilerek bu aşağıdaki gibi ifade edilir.

```
ENTITY unit_vector;
  a: real;
  b: real;
  c: optional_real;
WHERE
  a** 2 + b** 2 + c** 2 = 1.0;
END_ENTITY
```

**c** bir değere sahipse, kural ifadesi normal olarak değerlendirilir; **c** bir değere sahip değilse, alt ifade " $+c**2$ " değerlendirilmez.

İsteğe bağlı niteliklere sahip lokal kurallarla ilgili doğru bir yaklaşım, isteğe bağlı nitelik boş olduğunda, makül bir değer temin etmek için, standart bir fonksiyon kullanılır. İsteğe bağlı nitelik, bir değere sahip olduğu zaman, fonksiyon bu değere döner; aksi halde standart değere döner. Önceki örneğe daha doğru bir yaklaşım şudur:

```
ENTITY unit_vector;
  a, b : REAL;
  c : OPTIONAL REAL;
  WHERE
    length_1 : a**2 + b**2 + NVL(c, 0.0)**2 = 1.0;
END_ENTITY;
```

**Algoritmalar**, istenen son ifadeleri üreten ifadeler sıralamasıdır. Fonksiyonlar ve prosedürler olmak üzere iki çeşit algoritma belirlenebilir. Standart parametreler girdiyi bir algoritmaya tanıtır. Algoritma isimlendirildiği zaman, gerçek parametreler gerçek değerleri verir. Gerçek parametreler sayı, sıra ve tip olarak standart parametrelerle uyumlu olmalıdır.

**Prosedürler**, yardım istenen noktadan parametreleri elde eden bir algoritmadır. İstlenen son durumu meydana getirmek için herhangi bir tarzda söz konusu parametreler üzerinde faaliyet gösterir. Bu gerçek parametrelerin değeri, sadece standart parametreler değişken anahtar kelimelerle işlem gördüğü zaman değiştirilebilir. Örnek ;

```
PROCEDURE midpt(p1, p2 : point; VAR result : point) ;
```

**Fonksiyonlar**, parametreler üzerinde işlem yapan bir algoritmadır ve belirli bir tipin sonuç değerini oluşturur. Bir ifadedeki fonksiyon çağrısı, sonuç değerini değerlendirir. Bir fonksiyonu yardıma çağırma, yardıma çağırma noktasında üretilen sonuç değerinin oluşumuna eşdeğerdir. Örnek;

```
FUNCTION distance(p1, p2 : point) : REAL;
  (* İki nokta arasındaki en kısa mesafeyi hesaplar *)
END_FUNCTION;
```

```
FUNCTION normal(p1, p2, p3 : point) : vector;
```

```
(* Düzlemde üç nokta ile verilen bir düzlemin normalini hesaplar *)
END_FUNCTION;
```

```
ENTITY circle;
  centre : point;
  radius : REAL;
  axis   : vector;
DERIVE
  area : REAL := pi*radius**2;
END_ENTITY;
```

Bir fonksiyon, **return** deyiminin icra edilmesiyle sona erer. **return** deyimi tarafından ihtiyaç duyulan ifadenin değeri, çağrı fonksiyonu tarafından üretilen sonucu tanımlar.

### 1.3. EXPRESS 'te Bazı Varlık Örneklerinin Tanımlanması

Nokta, köşe ve kenar varlıklarının EXPRESS'te tanımlanması aşağıda verilen örneklerde olduğu gibi yapılır.

#### **Nokta tanımlama:**

```
*)
ENTITY point;
  SUPERTYPE OF (cartesian_point XOR
                point_on_curve XOR
                point_on_surface)
  SUBTYPE OF (geometry);
END_ENTITY;
(*
```

Yukarıdaki durumlar, herhangi bir yerde tanımlanan üç farklı nokta tipinin SUPERTYPE OF'ünü tanımlar. Nokta, kartezyen, eğri üzerinde veya bir yüzey üzerinde olabilir. Aynı zamanda bir nokta ait olduğu geometrinin bir alt tipi SUBTYPE'ıdır. Kartezyen noktanın tanımlanması aşağıda verilmiştir.

```
*)
ENTITY cartesian_point
  SUBTYPE OF (point);
  x_coordinate : REAL;
```

```

y_coordinate : REAL;
z_coordinate : OPTIONAL REAL;
DERIVE
dim:INTEGER: = count_dimensions ([X_coordinate,Y_coordinate,
                                Z_coordinate]);
END_ENTITY;
(*)

```

Bu blok bir kartezyen noktayı tanımlar. x-coordinate, y-coordinate, z-coordinate sırasıyla yerleştirilen noktanın X, Y ve Z koordinatlarıdır. İki boyutlu tanımlamalarda noktanın Z koordinatı olmayacaktır. Nokta varlığı üç boyutlu veya count\_dimensions fonksiyonuyla belirlendiği gibi, iki boyutlu uzayda tanımlanabilir. 3 boyutlu uzayda Z koordinatı ihmal edilirse, nokta tanımlama 2 koordinatlı olur.

### **Köşe tanımlama:**

```

*)
ENTITY vertex
SUBTYPE OF (topology);
vertex_point: OPTIONAL POINT;
WHERE
WR1: dimensionality(vertex) = 0;
WR2: qualitative_extent(vertex) = zero;
WR3: constraint_geometry_vertex(vertex);
END_ENTITY;
(*)

```

Yukardaki blok, köşe\_noktasının köşeyle ilgili isteğe bağlı geometrik bir nokta olduğunu tanımlar. WR1, köşenin boyutsal olarak " 0 " değerini; WR2, köşenin sıfır büyüklüğünü (genişlik); WR3 ise, yerine getirilmesi gereken köşenin sınırlarını ifade eder.

### **Kenar tanımlama:**

```

*)
ENTITY edge
SUBTYPE OF(topology);
edge_start : vertex;
edge_end   : vertex;

```

```

edge_curve : OPTIONAL curve or logical;
WHERE
WR1: dimensionality(edge) = 1;
WR2: qualitative_extent(edge) = finite_non_zero;
WR4: arcwise_connected(edge);
WR5: empty_set(geometric_intersection(edge_start_domain(edge)));
WR6: empty_set(geometric_intersection(edge_end_domain(edge)));
WR7: constraints_geometry_(edge);
END_ENTITY;
(*)

```

Burada **edge\_start**, kenarın başlangıç noktası (köşesi); **edge\_end** ise kenarın son noktası (köşesi) dir. **edge\_curve** kenar eğrinin isteğe bağlı eğri veya mantıksal varlık olduğunu ifade eder. Bu bir eğri veya eğri\_mantık\_yapısı olabilir. İlgili kuralların anlamları ise aşağıda açıklanmıştır.

WR1: Kenarın boyutu 1'dir.  
WR2: Kenarın uzantısı sonludur ve sıfır değildir.  
WR3: Kenar kendisiyle arakesit oluşturmaz.  
WR4: Kenar arcwise bağlantılıdır.  
WR5: Kenarın başlangıcı, kenarın bir parçası değildir.  
WR6: kenarın sonu, kenarın bir parçası değildir.  
WR7: Sınırlı\_geometrik\_kenar(kenar) fonksiyonu sağlanmalıdır.

Fonksiyon geometrik kısıtları değerlendirir ve onları tatmin edici bulursa, bu durumda TRUE'ya geri döner.



## 2. WIRTH SYNTAX NOTATION (WSN)

STEP deęiřtirme dosyasının syntaxı WSN'de tanımlanır. WSN ilk olarak Niklaus Wirth tarafından tanımlanmış ve Communications of the ACM, 20:11 (Nov 77), 822--823 tarafından yayınlanmıştır. Büyük harf formundaki dizi, dilin bir elemanıdır ve dizi elemanın ismidir.

Çift tırnak işareti ile sınırlandırılan her hangi bir dizin, çift tırnak işareti içerisinde bulunanın ne olduğunu tam olarak ifade eder. Eşit işareti "=" bir üretimi gösterir. Soldaki eleman, sağdaki elemanın kombinezonu olarak tanımlanır. Bir üretim, bir nokta "." ile sona erer. WSN notasyonu Tablo A.2'de gösterilmiştir.

Tabloda kullanılan sınır parantezlerinin anlamları aşağıda tanımlanmıştır.

BUKLELİ PARANTEZ	"{" sıfır veya daha fazla tekrarı gösterir.
KÖŞELİ PARANTEZ	"[]" isteęe baęlı parametreleri gösterir.
DÜŞEY ÇUBUK PARANTEZLER	" " mantıksal OR (veya)yı gösterir. "()" Özel olarak başka herhangi bir operasyonla birleşmede, seçilene kapsamak için öncelikli operasyonu gösterir. Örnek, $A(B C D) = AB AC AD$

Tablo A.2. WSN'nin Genel Yapısı

SYNTAX	= { PRODUCTION }.
PRODUCTION	= IDENTIFIER "=" EXPRESSION ".".
EXPRESSION	= TERM { " " TERM }.
TERM	= FACTOR { FACTOR }.
FACTOR	= IDENTIFIER   LITERAL   "[" EXPRESSION "]"   "(" EXPRESSION ")"   "{" EXPRESSION "}".
IDENTIFIER	= letter { letter }.
LITERAL	= "" character { character } "".

**Alfabe tanımlama:** WSN alfabesi, 32'den 127'ye kadar ASCII kodlarından çıkarılan ASCII karakterleri seti olarak tanımlanır. WSN alfabesi Tablo A.3'te verilmiştir.

**Düzenli ifadelerle tanımlanan token'lar:** Tablo A.4'teki düzenli ifadeler, Tablo A.3'te tanımlanan alfabe alt setlerini gösterir. Token'ları tanımlayan ifadelerin tümüyle tanınması gerekir. Tablo A.4'teki düzenli ifadeler, fiziksel dosyanın Token'larını tanımlar. Aşağıda verilen TOKEN tanımlaması okuyucuya kolaylık sağlaması bakımından kaynak teşkil eder. Bu, fiziksel dosyanın WSN tanımlaması tarafından referans alınmaz.

TOKEN = DELIMITER|KEYWORD | INTEGER | REAL |STRING |  
COMMENT|ENTITY| BINARY |ENTITY\_NAME  
| ENUMERATION .

Tablo A. 3. Alfabe Alt Setleri

SPACE	= " " .
DIGIT	= "0"   "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9" .
LOWER	= "a"   "b"   "c"   "d"   "e"   "f"   "g"   "h"   "i"   "j"   "k"   "l"   "m"   "n"   "o"   "p"   "q"   "r"   "s"   "t"   "u"   "v"   "w"   "x"   "y"   "z" .
UPPER	= "A"   "B"   "C"   "D"   "E"   "F"   "G"   "H"   "I"   "J"   "K"   "L"   "M"   "N"   "O"   "P"   "Q"   "R"   "S"   "T"   "U"   "V"   "W"   "X"   "Y"   "Z"   "_" .
SPECIAL	= "!"   "!"   "*"   "\$"   "%"   "&"   ""   "."   "#"   "+"   ","   "-"   "("   ")"   "?"   "/"   ":"   ";"   "<"   "="   ">"   "@"   "[   \"   ]"   "{"   " "   "}"   "~"   "^"   "`" .
BACKSLASH	= "\" .
QUOTE_CHAR	= "" .

Tablo A.4. Token Tanımlama

KEYWORD	= USER_DEFINED_KEYWORD   STANDARD_KEYWORD.
USER_DEFINED_KEYWORD	= !" UPPER { UPPER   DIGIT } .
STANDARD_KEYWORD	= UPPER { UPPER   DIGIT } .
SIGN	= "+"   "-".
INTEGER	= [ SIGN ] DIGIT { DIGIT } .
REAL	= [ SIGN ] DIGIT { DIGIT } "." { DIGIT } [ "E" [ SIGN ] DIGIT { DIGIT } ].
NON_Q_CHAR	= SPECIAL   DIGIT   SPACE   LOWER   UPPER .
STRING	= "" { NON_Q_CHAR   QUOTE_CHAR   BACKSLASH
BACKSLASH	[ CONTROL_ITEM } "" .
ENTITY_NAME	= "#" DIGIT [ DIGIT [ DIGIT [ DIGIT [ DIGIT [
DIGIT	[ DIGIT [ DIGIT [ DIGIT ] ] ] ] ] ] ] .
ENTITY_IDENTIFER	= "#" ENTITY_NAME.
ENTITY_REFERENCE	= "#" ENTITY_NAME.
ENUMERATION	= "." UPPER { UPPER   DIGIT } "." .
HEX	= "0"   "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "A"   "B"   "C"   "D"   "E"   "F" .
BINARY	= ( "0"   "1"   "2"   "3" )   "" HEX { HEX } "" .

Bir STEP dosyası, Token'lar sırasıyla temsil edilebilir. STEP dosya yapısında bir Token, bir sınırlayıcı, rezerv kelime, anahtar kelime veya basit veri tipi olarak tanımlanır.

**STEP dosya yapısının WSN'si:** Fiziksel dosyasının syntaxı Tablo A.5'te verilmiştir. Tüm fiziksel dosya yapıları bu syntaxla uyumlu olmalıdır. Tablo A.5, TabloA.4'te tanımlanan Token'ları referans alır. Bu WSN, belirli bir şemadan ziyade, mümkün olan tüm şemalar için fiziksel dosya yapısını temsil eder.

**Varlık tipi (Keyword):** Fiziksel dosyada belirli tip bir varlık örneğini gösteren özel karakterler dizisidir. Anahtar kelimeler, büyük harfler, sayılar, alt çizgi karakterler ve belki bir ünlem işaretinden oluşur. Burada ünlem işareti

anahtar kelimenin sadece ilk harfi olarak genelde bir defa yer alabilir. Ünlem işareti ile başlayan anahtar kelimeler, kullanıcı tanımlamalı varlıklar için kullanılır. Standart anahtar kelimeler, ünlem işareti ile başlamaz. Fiziksel dosyada temsil edilen bazı varlık tipi örnekleri aşağıdaki gibi verilebilir.

POINT, LINE, CIRCLE, EDGE\_LOOP, PLANE, AXIS, CLOSED\_SHALL

Tablo A.5. STEP Dosya Yapısının WSN'si

EXCHANGE_FILE	= "ISO-10303-21;"
HEADER_SECTION	DATA_SECTION
	"END-ISO-10303-21;" .
HEADER_SECTION	= "HEADER;" HEADER_ENTITY HEADER_ENTITY HEADER_ENTITY [HEADER_ENTITY_LIST] "ENDSEC;" .
HEADER_ENTITY_LIST	= HEADER_ENTITY { HEADER_ENTITY } .
HEADER_ENTITY	= KEYWORD "(" [ PARAMETER_LIST ] ")" ";" .
PARAMETER_LIST	= PARAMETER { "," PARAMETER } .
PARAMETER	= TYPED_PARAMETER   UNTYPED_PARAMETER   OMITTED_PARAMETER .
TYPED_PARAMETER	= KEYWORD "(" PARAMETER ")" .
UNTYPED_PARAMETER	= "\$"   INTEGER   REAL   STRING   ENTITY_NAME   ENUMERATION   BINARY   LIST .
OMITTED_PARAMETER	= "*" .
LIST	= "(" [ PARAMETER { "," PARAMETER } ] ")" .
DATA_SECTION	= "DATA;" ENTITY_INSTC_LIST "ENDSEC;" .
ENTITY_INSTC_LIST	= ENTITY_INSTANCE { ENTITY_INSTANCE } .
ENTITY_INSTANCE	= ENTITY_NAME "=" [ SCOPE ] SIMPLE_RECORD ";" 
ENTITY_NAME	"=" [ SCOPE ] SUBSUPER_RECORD ";" .
SCOPE	= "&SCOPE" ENTITY_INSTC_LIST "ENDSCOPE" [ EXPORT_LIST ] .
EXPORT_LIST	= "/" ENTITY_NAME { "," ENTITY_NAME } "/" .
SIMPLE_RECORD	= KEYWORD "(" [ PARAMETER_LIST ] ")" .
SUBSUPER_RECORD	= "(" SIMPLE_RECORD_LIST ")" .
SIMPLE_RECORD_LIST	= SIMPLE_RECORD { SIMPLE_RECORD } .

**Token ayırıcılar:** Token ayırıcı bir karakterdir veya herhangi iki token arasında görülebilen karakterler dizisidir. Token ayırıcılarının herhangi bir sayıdaki kombinezonu, herhangi iki token arasında birlikte görülebilir. Token ayırıcılar, tamsayılar, gerçek sayılar, listeler, varlık referansları, varlık tanımlayıcılar, rezerv kelimeler, anahtar kelimeler veya sınırlandırıcılar içinde görülmeyebilir. Boşluklar ve yorumlar token ayırıcılardır. Token ayırıcılar, boşluklar, yazılı talimatlar ve yorumlardır. Bir yorum "/\*" işareti ile başlar ve "\*/" işareti ile sona erer. Yorum içerisindeki her hangi bir karakterin fiziksel dosya yapısı için önemi yoktur.

**Sınırlayıcılar:** Sınırlayıcılar, özel karakterlerdir veya ayırma son verme, ya da verilerin anlamlı olarak toplanmasını göstermek için kullanılan özel karakterler kümesidir. Sınırlayıcılar Tablo A.6'da listelenmiştir.

Tablo A. 6. Sınırlayıcılar

KARAKTERLER	SINIRLAYICI	GÖREVİ
SCOPE	reserv kelime	Scope bölümünü başlatır
ENDSCOPE	reserv kelime	Scope bölümünü sona erdirir
,	virgül	Bölümleri ayırır
;	noktalı virgül	Oluşan varlığı sona erdirir
=	eşittir işareti	Belirleyiciyle varlık ismini ayırır
()	parantezler	Liste, set, dizin veya parametreleri kapsar
ENDSEC;	reserv kelime	Bir bölümü sona erdirir
ENDSTEP;	reserv kelime	Step dosyasını sona erdirir
/*	slaş-yıldız	Yorumları başlatır
*/	yıldız-slaş	Yorumları sona erdirir
/	slaş	Varlık export listesini içine alır

**Reserv kelimeler:** Rezerv kelimeler fonksiyonlar veya anlamlar ile ilgili belirli karakter dizisidir. Rezerv kelimeler büyük harfler, sayılar, alt çizgi karakterleri ve belki bir noktalı virgülden meydana gelir. Rezerv kelime veya anahtar bir kelime büyük harflerle ifade edilir ve bu kelimeler varlık tipi isimi olarak kullanılamaz. Bunlara ait örnekler aşağıda sıralanmıştır.

SCOPE  
 ENDScope  
 DATA;  
 ENDSEC;  
 STEP;  
 HEADER;  
 ENDSTEP;

**Basit veri tipleri:** STEP deęiřtirme format dosyalarında kullanılan altı basit veri tipi vardır. Bunlar; Integer, real, string, entity, identifier, entity reference ve enumeration'dir.

**Integer,** isteęe baęlı olarak "-" veya "+" iřaretiyle iřlem gören bir veya daha fazla sayıdan oluşur. Integer sayılar 10 tabanına göre ifade edilir.

**Real,** isteęe baęlı ondalıklı bir üs ile izlenen gerekli desimal mantisten meydana gelir. Mantis, isteęe baęlı "-" veya "+" iřareti, bir veya daha fazla sayı dizisi, ondalıklı bir nokta "." ve sıfır veya daha fazla sayılı dizinden meydana gelir. Ondalıklı üs, "E" karakteri ve önünde "+" veya "-" iřareti bulunan bir veya daha çok sayıdan oluşur. Fiziksel dosyadaki her **real** sayı, kendi tamlıęını ifade eder. Aynı zamanda bu deęer fiziksel dosyadaki veri temsilinin tamlıęına karřılık gelir. Örnek olarak iki real sayı řu řekilde verilebilir.

-32.178E+02 , 0.25E8

**String,** iki tek tırnak "'.....'" iřareti arasına yazılan boş veya bir çok karakterden oluşur. Boř dizin ' ' řeklinde, dolu dizin ise 'CAT' řeklinde gösterilir.

**Varlık tanımlayıcı:** Bir varlık tanımlayıcı bir sayı iřareti "#" ve bunu takip eden iřlem görmemiş bir sayıdan oluşur. Varlık tanımlayıcı, dosyada belirli bir düzen içerisinde olmayı gerektirmez. Örnek;

#12  
 #023

**Varlık referansı:** Varlık tanımlayıcılar, herhangi bir varlık nitelik listesinde görüldüęü zaman, bunlar, dięer varlıklara referans olarak kullanılır. Referans alınacak varlık tanımlayıcı sayı önceden tanımlanmalıdır. Örnek;

# 210 VERTEX(#12);

# 055 .....( #023);

**Enumeration:** Büyük harflerden veya büyük harflerle başlayan sayılardan oluşur. Büyük harfler koleksiyonu ve sayılar noktalarla sınırlandırılır. Verilen değer anlamı, enumeration tipi tanımlayan kavramsal şema ile tayin edilir. Örnek;

.FALSE.

.STEEL.

.TRUE.

.A123.

**Yapılanmış veri tipi:** STEP dosya yapısındaki yapılanmış veri tipi sadece listelerdir. Bir liste basit örneklerin veya yapılanmış veri tiplerinin koleksiyonudur. Liste bir sol parantez "(" ile başlar ve sağ parantez ")" ile sona erer. Listeler birbirlerinden virgüllerle ayrılır. Örnek;

List of List of Real ((0.0, 1.0, 2.0), (3.0, 4.0, 5.0))

List of List of Real ((0.0, 1.0, 2.0), ( ))

List of Integer (0, 1, 2, 3, 7, 2, 4)

List of String ('CAT', 'HELLO')



### 3. STEP DOSYA YAPISI

STEP deęiřtirme dosyası sıralı bir dosyadır. Dosyada ierilen bilgi serbest formattadır ve modüler bir tarzda hazırlanır. Dosya HEADER ve DATA bۆlümü olmak üzere, iki bۆlümde meydana gelir. STEP deęiřtirme dosyasının sonraki versiyonları ilave bۆlümleri ihtiva edebilir. Her bۆlüm, bۆlüm anahtar kelimesiyle bařlar ve bunu bir noktalı virgöl takip eder. STEP dosyası "STEP;" ile bařlar ve "ENDSTEP;" ile sona erer. HEADER bۆlümü dosya yapısının kendisiyle ilgili verileri, DATA bۆlümü ise, transfer edilen verileri saęlar.

**HEADER bۆlümü:** Dosyanın en bařında olmak üzere, sadece bir defa yazılan bu bۆlüm, "HEADER;" anahtar kelimesi ile bařlayıp "ENDSEC;" anahtar kelimesiyle sona erer. Bu bۆlüm her dosya yapısında mevcuttur.

HEADER bۆlümü, standart varlıkları ve varlık isimlerini tařımaması dıřında, DATA bۆlümü varlıkları gibi aynı genel syntaxı takip eder. Her dosya yapısının HEADER bۆlümünde belirli üç çeřit HEADER bۆlümü varlıęına ihtiya duyulur. Bunlar řu sırada yazılan FILE\_DESCRIPTION, FILE\_NAME ve FILE\_SCHEMA varlıklarıdır. İsteęe baęlı HEADER bۆlümü varlıkları, gerekli bulunursa, belirli bir sırada veya geliřigüzel yazılabilir. HEADER bۆlümü varlıkları ařaęıdaki genel formatı takip eder:

```
HEADER_ENTITY = KEYWORD "(" [ PARAMETER_LIST ] ")" ";"
```

Bu yazımda Keyword, varlıęın tipini karřılar. Parametre listesi, yapılanmıř veri tipleri veya basit bir örneęin listesidir. Parametreler, parantezler ierisine yazılarak virgüllerle birbirinden ayrılır. HEADER bۆlümü varlık örneęi noktalı virgöl ile sona erer.

HEADER bۆlümündeki kullanıcı tanımlamalı varlıklar, belirli kısıtlar altında, bu bۆlüm ierisinde yer almalıdır. Kullanıcı tanımlamalı varlıklar, ünlem iřareti "!" ile bařlaması dıřında, HEADER bۆlümündeki genel formatla aynı olmalıdır. Kullanıcı tanımlamalı varlıklar, "ISO~10303" te tanımlanan herhangi basit veri tiplerini veya yapılanmıř veri tiplerini kullanmaması gerekir. HEADER bۆlümü ile ilgili format ařaęıda verilmiřtir:

```
HEADER;
.
.
FILE_SCHEMA(("GEOMETRY"));
```

```
!A_SPECIAL_ENTITY ('ABC',123); /*Kullanıcı tanımlamalı varlık
nümunesi*/
.
.
ENDSEC;
```

**DATA bölümü:** DATA bölümü, EXPRESS dilini kullanarak tanımlanmış varlık örneklerinin bir koleksiyonudur. Bu bölüm "DATA;" anahtar kelimesiyle başlayıp transfer edilecek ürün verileri ile devam ederek "ENDSEC;" anahtar kelimesiyle sona erer. DATA bölümü, Fiziksel dosyada sadece bir defa yer alır. Bu bölümde referans alınacak varlıklar, referans alınmadan önce tanımlanmalıdır. DATA bölümündeki varlıklar aşağıdaki tarzda açıklanır.

```
ENTITY_IDENTIFIER = ENTITY_TYPE "(" [ PARAMETER_LIST ] ")"
```

Burada:

ENTITY\_IDENTIFIER, Fiziksel dosyada yer alan varlık tanımlayıcı olup "#" işaretiyle başlar ve bunu bir tam sayı izler.

ENTITY\_TYPE, varlığı tanımlayan bir anahtar kelimedir ve bu kelimedenden sonra bir sol parantez "(" açılır.

PARAMETER\_LIST, Parametre listesi ise, yapılanmış veri tipleri veya basit bir örneğin listesidir. Parametreler, parantezler içerisine yazılarak virgüllerle birbirinden ayrılır. Tam bir varlık örneği noktalı virgül ile sona erer.

Bir varlık örneğinde isteğe bağlı nitelik için değer verilmediği zaman nitelik değeri ", , " veya ",\$, " şeklinde gösterilir. Örnek;

```
#2 = NODE(1, $, 2,'ABC');
```

**SCOPE yapısı:** Varlık örneklerinde SCOPE yapısı, bir referans alanı sağlamak ve varlık sahipliğini belirlemede kullanılan bir mekanizmadır. SCOPE yapısı tüm varlık örnekleri için isteğe bağlıdır.

Bu yapı "SCOPE" rezerv kelimesiyle başlar, en azından bir varlık bunu takip eder; "ENDSCOPE" rezerv kelimesiyle sona erer. SCOPE yapısı içerisindeki varlıkların kendileri, bir SCOPE yapısına sahip olabilir. Bir SCOPE yapısı içindeki varlık örnekleri, aynı SCOPE içerisinde bulunan diğer varlık örnekleri tarafından, standart olarak referans alınabilir. ENDSCOPE' dan sonra isteğe bağlı nitelik listesi görülebilir. Buna ait bir örnek aşağıda sunulmuştur.

```
#1 = TRIANGLE  
SCOPE  
  #2 = POINT(10,5,0);  
  #3 = POINT(35,5,0);  
  #4 = POINT(17,13,0);  
ENDSCOPE  
(LINE(#2,#3), LINE(#3, #4), LINE(#4, #2));
```

Yukardaki örnekte ÜÇGEN varlığı, üç ÇİZGİ varlığı ile tanımlanmıştır. ÇİZGİ varlıkları, ÜÇGEN varlığının SCOPE'unda olduğundan, bunlar, ÜÇGEN varlığına bağlı olarak yerleştirilir. Çizgileri tanımlayan NOKTA varlıkları, ÜÇGEN'e bağlıdır. NOKTA varlıkları, aynı scope seviyesindeki ÇİZGİ varlıkları tarafından referans alınacak şekilde bulunur.



#### 4. EXPRESS DİLİNDEN FİZİKSEL DOSYA YAPISINA PLANLAMA

EXPRESS dili deklaratif ile icra edilebilir deyimler ve algoritmaları içerir. EXPRESS dilinde tanımlanan ürün veri varlıkları, daha sonra STEP standartının fiziksel dosya formatına göre yazılır. Ürün verileri değiştirme formatına sadece deklaratif deyimler yazılır. İcra edilebilir deyimler ve algoritmalar değiştirme formatına yazılmaz. EXPRESS varlıkları fiziksel dosyaya yazma işlemi aşağıda açıklanmıştır.

**Veri tiplerini fiziksel dosyaya yazma:** INTEGER, STRING ve REAL parametreleri, EXPRESS yapılarında tanımlandığı şekliyle, integer, string ve real olarak değiştirilmeden DATA bölümüne yazılır.

BOOLEAN parametresinin EXPRESS yapısı DATA bölümüne ".T." ve ".F." gibi enumeration veri tipi olarak yazılır. Benzer şekilde LOGICAL parametresinin EXPRESS yapısı da, DATA bölümüne ".T." , ".F." veya ".U." gibi enumeration veri tipi olarak yazılır. Bu değerler sırasıyla true, false ve unknown kelimelerine karşılık gelmektedir. Buna ait bir örnek aşağıda verilmiştir.

EXPRESS'te varlık tanımlama

ENTITY SAMLE

```

attribute1: INTEGER(2);  <----- A
attribute2: INTEGER;    <----- B
attribute3: STRING(3);  <----- C
attribute4: STRING      <----- D
attribute5: LOGICAL;    <----- E
attribute5 : BOOLEAN;   <----- F
attribute6: REAL;       <----- G
attribute7: REAL;       <----- H

```

END\_ENTITY;

Örnek varlığın DATA bölümünde oluşumu:

```
#2=WIDGET(99, 99999, 'ABC', 'ABCDEFG', .T., .F., 9.000, 1.2345);
```

```

      ^   ^   ^           ^   ^   ^   ^   ^
      |   |   |           |   |   |   |   |
      A   B   C           D   E   F   G   H

```

Bu varlığın oluşumunda niteliklerin aldığı değerler şunlardır:

- A nitelik 1, 99 değerindedir. Parantez içindeki 2 sayısı bu niteliğin kabul edilen ondalık sayı adedini gösterir.
- B nitelik2, 99999 değerindedir.
- C nitelik3, üç karakterli 'ABC' değerindedir.
- D nitelik4, 'ABCDEFGH' değerindedir.
- E nitelik5, TRUE değerindedir.
- F nitelik6, FALSE değerindedir.
- G nitelik7, 9 değerindedir.
- H nitelik8, 1.2345 değerindedir.

**Liste:** EXPRESS'te tanımlanan liste, DATA bölümüne bir liste veri tipi olarak yazılır. Liste boş ise DATA bölümünde boş parantez "()" olarak görülür. Buna ait bir örnek aşağıda verilmiştir.

EXPRESS'te varlık tanımlama:

```
ENTITY widget;
  attribute1: LIST [0 : #] OF INTEGER; -----> A
  attribute2: LIST [1 : #] OF INTEGER; -----> B
  attribute3: OPTIONAL LIST [1 : ?] OF INTEGER; --> C
  attribute4: REAL; -----> D
END_ENTITY;
```

Örnek varlığın DATA bölümünde oluşumu:

```
#4 = WIDGET( ^, (1,2,4), ^, 2.56);
              ^   ^   ^   ^
              |   |   |   |
              A   B   C   D
```

Bu varlığın oluşumunda niteliklerin aldığı değerler şunlardır:

- A nitelik1, boş bir listedir.
- B nitelik2, üç elemanı içermektedir.
- C nitelik3, değer almamıştır.
- D nitelik4, 2.56 değerindedir.

**Array:** EXPRESS'te konstrükte edilen ARRAY, bir liste veri tipi olarak DATA bölümüne yazılır. Örnek:

EXPRESS'te varlık tanımlama:

```
ENTITY widget;
  attribute1: ARRAY [-1 : 3] OF INTEGER;           <----- A
  attribute2: ARRAY [1 : 5] OF OPTIONAL INTEGER;  <----- B
  attribute3: ARRAY [1 : 2] OF ARRAY [1 : 3] OF INTEGER; <- ----- C
END_ENTITY;
```

DATA bölümünde örnek varlığın oluşumu:

```
#30 = WIDGET((1,2,3,4,5) , (1,2,3,$,5) , ((1,2,3),(4,5,6)));
           ^           ^           ^
           |           |           |
           A           B           C
```

A nitelik1, aşağıdaki değerlerden meydana gelir:

```
attribute1 [-1] = 1
attribute1 [0] = 2
attribute1 [1] = 3
attribute1 [2] = 4
attribute1 [3] = 5
```

B nitelik2, aşağıdaki değerlerden meydana gelir:

```
attribute2 [1] = 1
attribute2 [2] = 2
attribute2 [3] = 3
attribute2 [4] = MISSING
attribute2 [5] = 5
```

Eksik değer (missing) yokluğu veya anlamı, EXPRESS şemasında açıklanmalıdır.

C nitelik3, aşağıdaki değerlerden meydana gelir:

```
attribute3 [1,1] = 1
```

```

attribute3 [1,2] = 2
attribute3 [1,3] = 3
attribute3 [2,1] = 4
attribute3 [2,2] = 5
attribute3 [2,3] = 6

```

**Type:** EXPRES yapısına göre hazırlanan TİP deklarasyonu, DATA bölümüne yazılmaz. EXPRES'te tanımlanan herhangi bir varlığın bir niteliği olarak yer alır. Örneğin eğer bir tip REAL olarak tanımlanmışsa, real sayı DATA bölümüne yazılır. Tip planlama örneği aşağıda gösterilmiştir.

EXPRESS'te varlık tanımlama:

```

TYPE
  type1 = INTEGER;
END_TYPE;

TYPE
  type2 = LIST [1 : 2] of REAL;
END_TYPE;

ENTITY widget;
  attribute1: LOGICAL; -----> A
  attribute2: TYPE1; -----> B
  attribute3: TYPE2; -----> C
END_ENTITY;

```

DATA bölümünde örnek varlığın oluşumu:

```

#4 = WIDGET( .T., 256, (1.0,0.0));
          ^   ^   ^
          |   |   |
          A   B   C

```

Bu varlığın oluşumunda niteliklerin aldığı değerler şunlardır:

- A nitelik1, TRUE değerindedir.
- B Type1, bir integer olup 256 değerindedir.
- C Type2, array olup listede 2 REAL elemanı vardır.



**Enumeration:** EXPRESS'te konstrükte edilen ENUMERATION, büyük harflerden oluşan bir enumeration veri tipi olarak sağına ve soluna birer nokta konup DATA bölümüne yazılır.

EXPRESS'te varlık tanımlama:

TYPE

primary\_colour = ENUMERATION OF (red, green, blue);

END\_TYPE;

ENTITY widget;

colour: primary\_colour; -----> A

END\_ENTITY;

Örnek varlığın DATA bölümünde oluşumu:

#2 = WIDGET(.RED.);

^

|

A

A bu örnekte varlığın renk niteliği RED değerini almıştır.

**Entity:** EXPRESS'te hazırlanan ENTITY yapısı, bir varlık olarak DATA bölümüne yazılır. EXPRESS'te varlık tanımlamalarının karmaşıklığından dolayı, bir takım yazma kuralına ihtiyaç vardır. Yazma kuralları aşağıdaki varlık tanımlamalarında verilmiştir.

- Basit açık niteliklerle tanımlanan varlıklar,
- İsteğe bağlı niteliklerle tanımlanan varlıklar,
- Çıkarılan niteliklerle tanımlanan varlıklar,
- Bir varlığın nitelikleri olan diğer varlıklarla tanımlanan varlıklar,
- Diğer varlıkların alt veya üst tipi olarak tanımlanan varlıklar,
- WHERE kurallarıyla tanımlanan varlıklar,
- INVERSE nitelikleriyle tanımlanan varlıklar.

Bu bölümde basit ve belirli niteliklerle tanımlanan varlıklar açıklanarak birer örnek verilmiştir. Diğer varlık tanımlamalarıyla ilgili daha fazla bilgi "ISO

10303--21\ Implementation Methods: Clear text encoding of the exchange structure" de bulunabilir.

Belirli niteliklere sahip varlıklar, ürün verileri deęiřtirme dosyasının DATA bölümüne doğrudan yazılır. Varlık tipi ismi, küçük harfler büyük harflere dönüřtürülmüř olarak aynı isimle yazılır. EXPRESS'te varlık tanımlamalarının nitelik sırası, EXPRESS varlık ismini izleyen ilk varlığın nitelięi, birinci nitelik; ikinci varlık nitelięi de ikinci nitelik olarak dikkate alınır. Dięer nitelikler de aynı řekilde yerine getirilir.

EXPRESS varlık nitelikleri, yukarda DATA bölümü varlık örneklerinde açıklandığı sırada bire bir yazılır. DATA bölümündeki varlık örneklerinin nitelik sırası, varlık tipi anahtar kelimesini izleyen ilk datanın deęeri, ilk nitelięin deęeri olacak řekilde dikkate alınır. Benzer řekilde varlık tipi anahtar kelimesini izleyen ikinci deęer, ikinci nitelięin deęeri olarak dikkate alınır. Dięerleri de aynı tarzda yerine getirilir.

EXPRESS'te varlık tanımlama:

TYPE

```
primary_color_abbreviation = ENUMERATION OF (r, g, b);
END_TYPE;
```

```
ENTITY widget; -----> A
  attribute1: INTEGER; -----> B
  attribute2: STRING; -----> C
  attribute3: LOGICAL; -----> D
  attribute4: BOOLEAN; -----> E
  attribute5: REAL; -----> F
  attribute6: LIST [1 : 2] of LOGICAL; -----> G
  attribute7: ARRAY [-1:3] of INTEGER; -----> H
  attribute8: PRIMARY_COLOR_ABBREVIATION; ---> I
END_ENTITY;
```

Örnek varlığın DATA bölümünde oluşumu:

```
#1 = WIDGET( 1, 'A', .T., .F., 1.0, (.T.,.F.), (1,0,1,2,3), .R.);
      ^      ^      ^      ^      ^      ^      ^      ^      ^
      |      |      |      |      |      |      |      |      |
      A      B      C      D      E      F      G      H      I
```

A EXPRESS'te tanımlanan varlık ismi widget, varlık ismi anahtar kelimesi " WIDGET " olarak DATA bölümüne yazılır.

Bu varlığın oluşumunda niteliklerin aldığı değerler şunlardır:

- B nitelik1 1 değerine sahiptir.
- C nitelik2 'A' değerine sahiptir.
- D nitelik3 .T. değerine sahiptir.
- E nitelik4 .F. değerine sahiptir.
- F nitelik5 1.0 değerine sahiptir.
- G nitelik6 mantıksal bir listedir ve liste değeri aşağıda verilmiştir:

ATTRIBUTE6(1) = .T.

ATTRIBUTE6(2) = .F.

H nitelik7 tamsayılar dizinidir ve bunların değeri aşağıda sıralanmıştır:

ATTRIBUTE7(-1) = 1

ATTRIBUTE7( 0) = 0

ATTRIBUTE7( 1) = 1

ATTRIBUTE7( 2) = 2

ATTRIBUTE7( 3) = 3

I nitelik8 bir Enumeration olup .R. değerindedir.

#### 4.1. Diğer Varlıkların Alt veya Üst Tipleri Olarak Tanımlanan Varlıklar

EXPRESS varlıkları, karmaşık yapıları nedeniyle, diğer varlıkların alt veya üst tipleri olarak tanımlanır. Bu iki zorunlu tanımlama kuralı, **internal** veya **external tanımlama** olarak adlandırılır. Bir varlığa ABSTRACT üst tipi olmadığı sürece, sadece bir tanımlama kuralı uygulanır. ABSTRACT üst tipi varlıklar fiziksel dosyada ayrı tanınabilir varlık örneği olarak olmamalıdır.

İlgili varlık deklere edildiğinde, subtype/supertype grafiğinin karakteristikleri tarafından, tam olarak tayin edilir. Üst tipe sahip herhangi bir varlık tipi AND, ANDOR veya ONEOF gibi üç EXPRESS üst tip operasyonunun birisine dahil olur. ANDOR operasyonu, bir üst tip ve bunun alt tipi arasındaki ilişkiyi belirlemek için kullanılan açık bir üst tip cümlesi olmadığı zaman gereklidir.

EXPRESS'te varlık tanımlama:

```
ENTITY a;
END_ENTITY;
```

```
ENTITY b SUBTYPE OF (a);
END_ENTITY;
```

```
ENTITY c SUBTYPE OF (a);
END_ENTITY;
```

b ve c varlık tipleri ANDOR operasyonuna dahil edilir.

EXPRESS'te varlık tanımlama

```
ENTITY a SUPERTYPE OF (b AND c);
END_ENTITY ;
```

```
ENTITY b SUBTYPE OF (a);
END_ENTITY;
```

```
ENTITY c SUBTYPE OF (a);
END_ENTITY;
```

b ve c varlık tipleri AND operasyonuna dahil edilir. Hatta aşağıda verilen örnekte olduğu gibi, sadece bir alt tipe sahip bir varlık tipi için bu mümkündür.

```
ENTITY a SUPERTYPE OF (b);
END_ENTITY;
```

Bu duruma b varlık tipi ONEOF operasyonuna dahil edilmiş gibi dikkate alınır.

**Internal planlama:** Varlık tipinin kendisi ve onun tüm üst tipleri, sadece ONEOF operasyonuna dahil edilirse, bu durumda internal planlama uygulanır. Bir varlık tipi, Internal planlama ile kendisinin bütün üst tiplerinin niteliklerini taşır. Taşınan tüm nitelikler, herhangi bir varlığın niteliklerini sıralı olarak gösterecektir.

SUPERTYPE varlığın nitelikleri, sıralı olarak taşınır ve bu nitelikler kendi varlık üst tipinde görülür.

Aşağıda Internal planlamaya ait bir varlık tanımlaması verilmiştir.

```
ENTITY aa SUPERTYPE OF (ONEOF(bb,cc)); --> A
  attrib_a : STRING;
END_ENTITY;
```

```
ENTITY bb SUBTYPE OF (aa); -----> B
  attrib_b : INTEGER;
END_ENTITY;
```

```
ENTITY cc SUBTYPE OF (aa); -----> C
  attrib_c : REAL;
END_ENTITY;
```

```
ENTITY dd ; -----> D
  attrib_d : aa;
END_ENTITY;
```

Örnek varlığın DATA bölümünde oluşumu:

```
#1 = AA('SAMPLE STRING'); /*----- A */
#2 = BB('ABC', 666); /*----- B */
#3 = CC('XYZ', 99.99); /*----- C */
#4 = DD(#1); /*----- D */
#5 = DD(#2); /*----- D */
#6 = DD(#3); /*----- D */
```

Bu varlığın oluşumunda niteliklerin aldığı değerler şunlardır:

A bir üst tip varlık olan aa, fiziksel dosyaya AA varlık ismi ve ilgili parametresi ile yazılır.

B bb varlığı, aa varlığının bir alt tipidir. Bu nedenle aa ve bb'nin birlikte niteliklerini taşır.

C cc varlığı, aa varlığının bir alt tipidir. Bu nedenle aa ve cc'nin birlikte niteliklerini taşır.

D dd varlığı, nitelik olarak aa'yı referans alır. Böylece bir dd varlık örneği #1, #2 veya #3 herhangi birini referans alabilir.

**External planlama:** Varlık tipinin kendisi ve onun herhangi üst tipleri, sadece AND veya ANDOR operasyonuna dahil edilirse, bu durumda external planlama uygulanır. Aşağıda ANDOR ile ilgili bir alt tip planlama örneği verilmiştir.

```
ENTITY aa SUPERTYPE OF (bb ANDOR cc); --> A
  attrib_a : STRING;
END_ENTITY;
```

```
ENTITY bb SUBTYPE OF (aa); -----> B
  attrib_b : INTEGER;
END_ENTITY;
```

```
ENTITY cc SUBTYPE OF (aa); -----> C
  attrib_c : REAL;
END_ENTITY;
```

```
ENTITY dd ; -----> D
  attrib_d : aa;
END_ENTITY;
```

Örnek varlığın DATA bölümünde oluşumu:

```
#1 = (AA('SAMPLE STRING')BB(15)); /* ----- A */
#2 = (CC(3.0)AA('S')); /* ----- B */
#3 = (BB(17)AA('ASTRID')CC(4.0)); /* ----- C */
#4 = DD(#1); /* ----- D */
#5 = DD(#2); /* ----- D */
#6 = DD(#3); /* ----- D */
```

Bu varlığın oluşumunda niteliklerin aldığı değerler şunlardır:

A #1, aa ve bb varlıklarının bütünleşmiş bir örneğidir.

B #2, aa ve cc varlıklarının bütünleşmiş bir örneğidir.

C #3, aa, bb ve cc varlıklarının bütünleşmiş bir örneğidir.

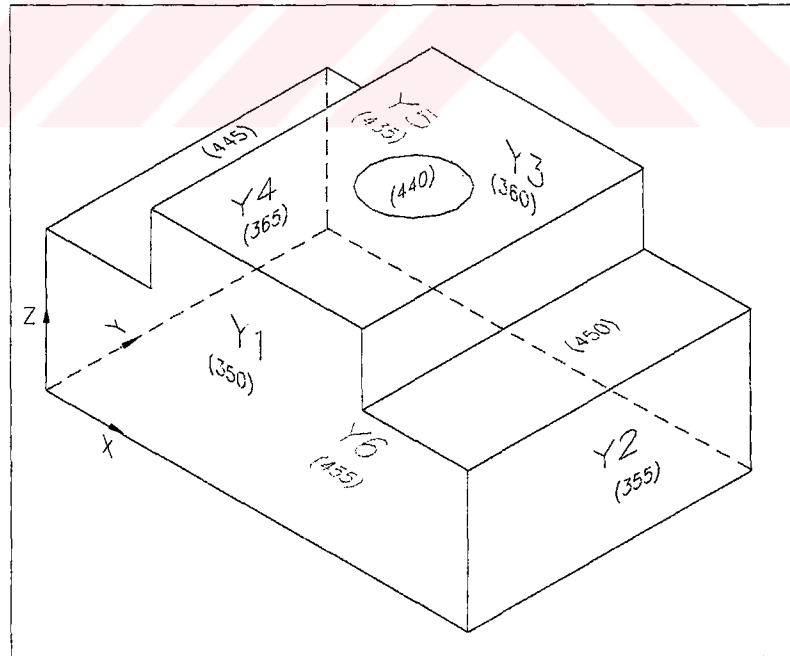
D dd varlığı, bir nitelik olarak aa varlığını referans alır. Böylece dd varlık örneği #1, #2 veya #3'ün herhangi birini referans alabilir.

## EK-B STEP'TE MEKANİK PARÇALARIN TEMSİLİ İÇİN VARLIK TANIMLAMALARI

### 1. STEP'TE MEKANİK PARÇALARI TEMSİL ETME

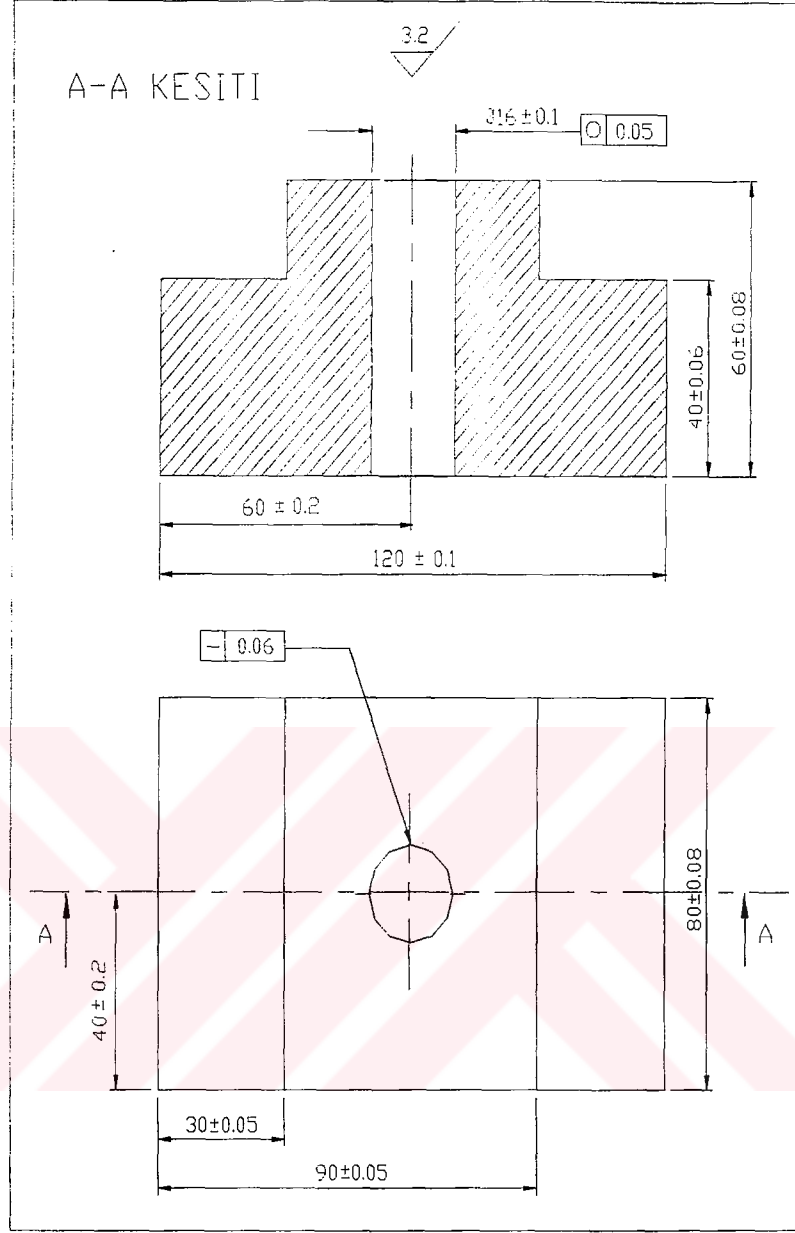
EK-A'da açıklandığı gibi, mekanik parça bilgisi, EXPRESS dili kullanmak suretiyle doğrudan fiziksel dosyada temsil edilir. Bir STEP dosyası, **STEP** ile başlar ve **ENDSTEP** ile sona erer. STEP dosyası, **HEADER** ve **DATA** kısmı olmak üzere iki bölümden meydana gelir. Ürün verileri DATA bölümünde temsil edilir. Bu bölümde Şekil B.1 ve B.2'de verilen örnek parçanın, STEP'te temsil edilmesi açıklanmıştır.

Bu çalışmada kullanılan örnek parçaların, ST modellemesi kullanan bir BDT sisteminde tasarlanmış olduğu kabul edilmiştir. Örnek olarak alınan parça, altı düzlem yüzeye ve yüzeylerden biri üzerine yerleştirilen bir deliğe ve iki kademe özelliğine sahiptir.



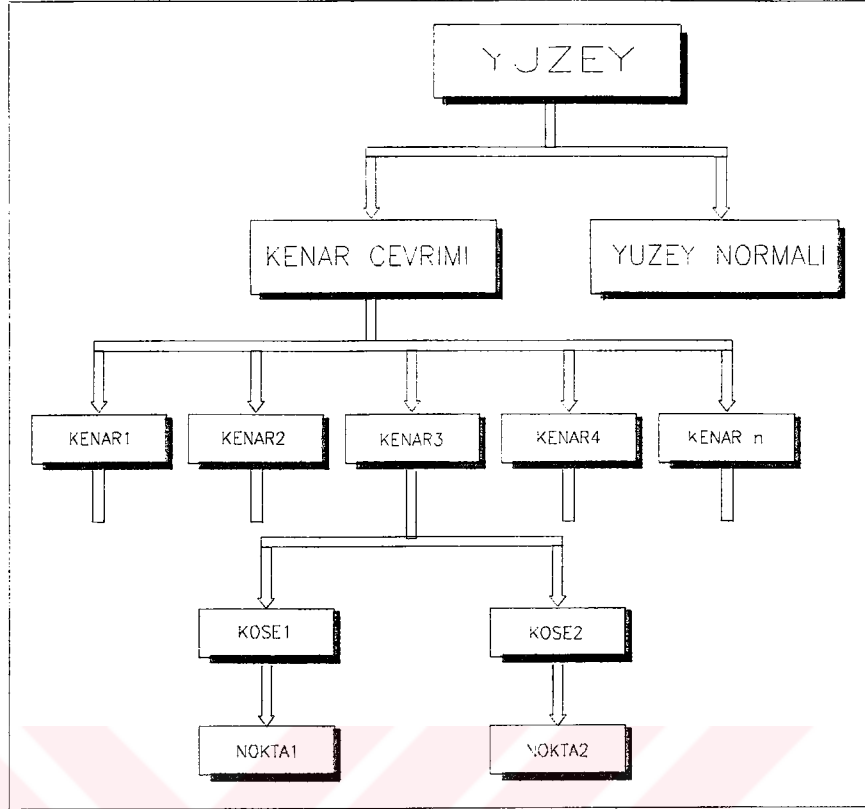
Şekil B.1. STEP formatı hazırlanan örnek parça





Şekil B.2. Örnek parçanın imalat resmi

Bir yüzey; kenarlar, köşeler (noktalar) ve bir yüzey normali ile ST'de temsil edilir (Şekil B.3). ST'de geometri ve topolojinin her ikisi de tanımlanmalıdır. Yüzeyin topolojisi, dört kenardan meydana gelen bir kenar çevrimi ile temsil edilir ve her kenar iki köşeden oluşur. Her köşe Şekil B.3'te gösterildiği gibi bir noktaya karşılık gelir.



Şekil B.3. Bir yüzeyin ST modelde topolojik temsili

**Nokta temsil etme:** STEP'te varlık tanımlaması "#" işareti ile başlayıp bunu varlığın özel numarası, ismi (varlık belirleyici) ve ilgili parametreleri takip eder. Nokta varlığı, üç boyutlu uzayda ise, üç kordinat değerine sahiptir ve EXPRESS dilinde aşağıdaki gibi temsil edilir.

\*)

ENTITY point

SUPERTYPE OF (ONEOF (cartesian\_point,point\_on\_curve,  
point\_on\_surface))

SUBTYPE OF (geometry);

END\_ENTITY;

(\*

(\*

ENTITY cartesian\_point;

SUBTYPE OF (point);

```

x coordinate : lenght_measure;
y coordinate : lenght_measure;
z coordinate : OPTIONAL lenght_measure;
END_ENTITY;
(*)

```

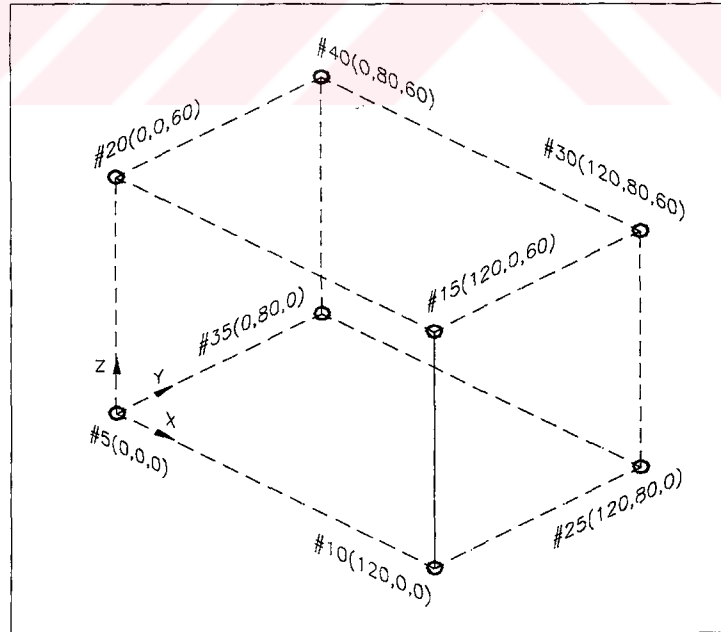
**x\_coordinate**, yerleştirilen noktanın X koordinatıdır.  
**y\_coordinate**, yerleştirilen noktanın Y koordinatıdır.  
**z\_coordinate**, yerleştirilen noktanın Z koordinatıdır.

Bu açıklamalara göre Şekil B.1'de verilen parçaya kademe özellikleri yerleşmeden önce, tüm yüzeylerinin köşelerine karşılık gelen noktalar Şekil B.4'te olduğu gibi gösterilebilir. Böylece Y1 yüzeyinin noktaları, fiziksel dosyada, aşağıdaki şekilde olacaktır. Burada noktaların sıralaması, daha ileride açıklanan yüzeylerin kenar çevrimi dikkate alınarak yapılmıştır.

```

#5 = POINT(0,0,0);
#10 = POINT(120,0,0);
#15 = POINT(120,0,60);
#20 = POINT(0,0,60);

```



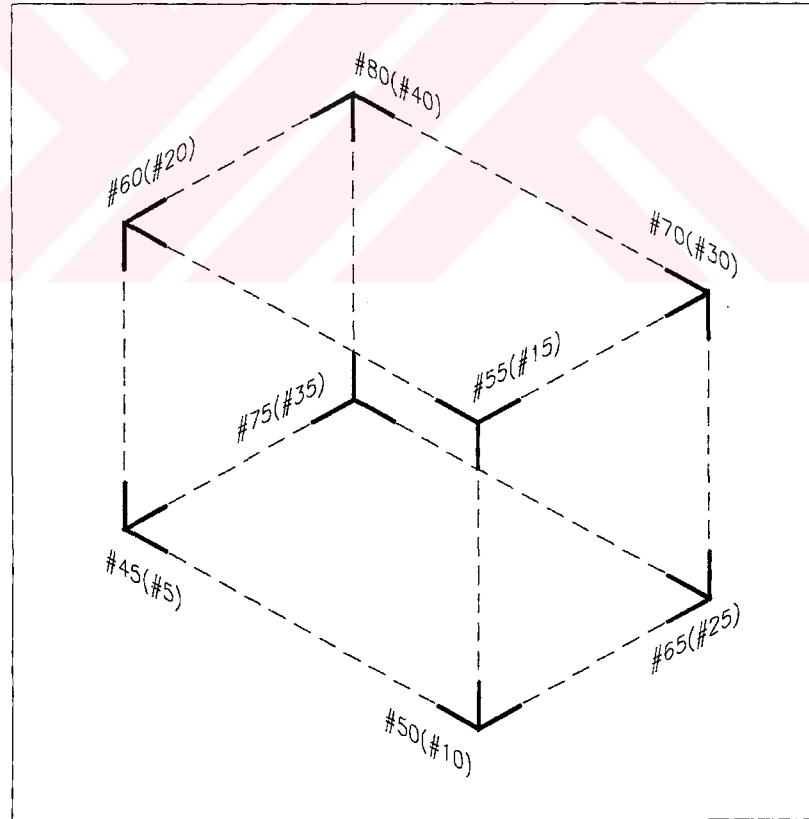
Şekil B.4. Nokta temsil etme

**Köşe temsil etme:** Noktalar belirlendikten sonra köşeler tanımlanabilir. Noktalar önce belirlendiği için, köşeler, noktaların özel varlık numaralarını, referans olarak kolayca tanımlanır. EXPRESS'te bir köşe şu şekilde tanımlanır:

```
*)  
ENTITY vertex  
SUBTYPE OF (topology);  
  vertex_point: OPTIONAL point;  
END_ENTITY;  
(*
```

**vertex\_point**, köşe ile ilgili geometrik bir noktadır.

Örnek parçanın köşeleri Şekil B.5'te gösterilmiştir.



Şekil B.5. STEP'te köşe tanımlama

Köşeler temsil edildikten sonra, Y6 yüzeyine ait köşeler fiziksel dosyada aşağıdaki şekilde olur.

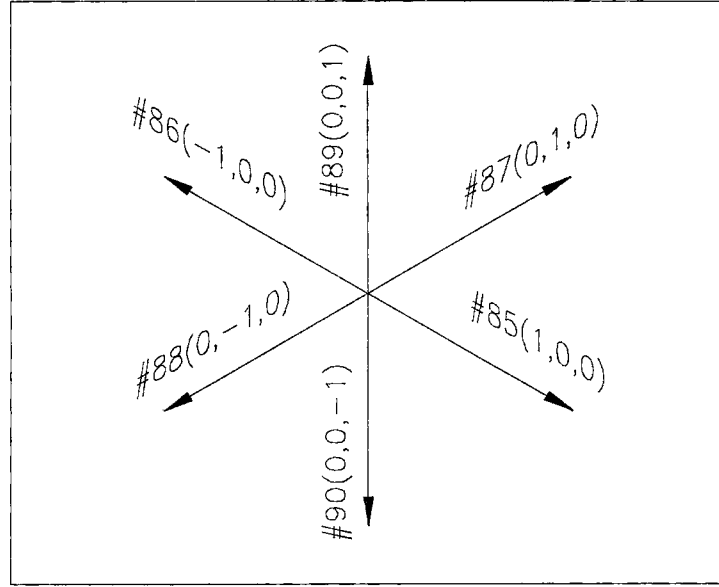
```
#45 = VERTEX(#5);
#50 = VERTEX(#10);
#55 = VERTEX(#15);
#60 = VERTEX(#20);
```

**Kenar tanımlama:** Bir kenarı tanımlamadan önce, kenarı yerleştirmek için, koordinat uzayında ilgili geometrik eğri tanımlanmalıdır. Bu açıklamada, üzerinde çalışılan parçaların kenarlarına ait eğriler, birer doğrudur. Bir doğru, bir nokta ve yönü ile tanımlanan sınırlandırılmamış bir eğridir. Doğrunun pozitif yönü, yön vektörünün yönündedir. Her doğru bir yön vektörüne sahip olduğundan, doğrudan önce bir yön vektörü tanımlanmalıdır. Örnek parçaya ait yön vektörleri Şekil B.6'da gösterilmiştir. Yön vektörü EXPRESS' te aşağıdaki şekilde tanımlanır.

```
*)
ENTITY vector
  SUPERTYPE OF (ONE OF (direction, vector_with_magnitude));
  SUBTYPE OF (geometry);
END_ENTITY;
(*
```

```
*)
ENTITY direction
  SUBTYPE OF (vector);
  X: REAL;
  Y: REAL;
  Z: OPTIONAL REAL;
END_ENTITY;
(*
```

X,Y ve Z sırasıyla X,Y ve Z eksenlerinin yönlerinin elemanlarıdır.



Şekil B.6. Yön vektörü tanımlama

Y1 yüzeyinin kenarlarının yön vektörleri, fiziksel dosyaya aşağıdaki gibi yazılır.

#85 = DIRECTION(1,0,0)	/* + X Yönü */
#89 = DIRECTION(,0,0,1)	/* +Z Yönü */
#86 = DIRECTION(-1,0,0)	/* - X Yönü */
#90 = DIRECTION(0,0,-1)	/* - Z Yönü */

Yön vektörü belirlenen bir doğru, EXPRESS'te aşağıdaki şekilde temsil edilir. Örnek parçaya ait doğrular Şekil B.7'de gösterilmiştir.

```
*)
ENTITY curve
  SUPERTYPE OF (ONE OF (line, conic, bounded_curve,
                        curve_on_surface, offset_curve));
```

```
  SUBTYPE OF (geometry);
END_ENTITY;
(*
```

```
*)
ENTITY line
  SUBTYPE OF (curve);
```

```

pnt: cartesian_point
dir: direction;
DERIVE
dim: INTEGER := coordinate_space(pnt);
WHERE
WR1: coordinate_space(pnt) = coordinate_space(dir);
WR2: (NOT EXIST (pnt.local_coordinate_system)) AND
      (NOT EXIST (dir.local_coordinate_system)) OR
      (pnt.coordinate_system = dir.local_coordinate_system);
END_ENTITY;
(*)

```

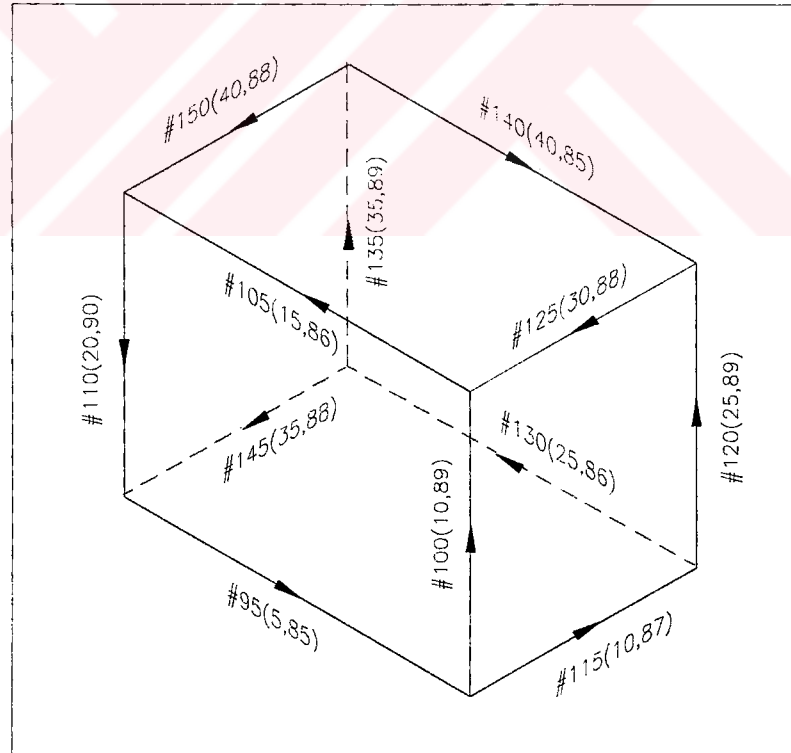
**pnt**, doğrunun konumudur.

**dir**, doğrunun yönüdür.

**dim**, doğrunun koordinat uzayının boyutsallığıdır.

**WR1**, **Pnt** ve **dir**, her ikisi de 2-D veya 3-D varlıklar olmalıdır.

**WR2**, **Pnt** ve **dir**, aynı koordinat sistemini kullanarak tanımlanmalıdır.



Şekil B.7. Doğru tanımlama

Y1 yüzeyinin doğruları, fiziksel dosyaya yazıldığında aşağıdaki şekilde olur.

```
#95 = LINE(#5, #85); /* Nokta ve +X yönünü referans alan doğru */
#100 = LINE(#10, #89); /* Nokta ve +Z yönünü referans alan doğru */
#105 = LINE(#15, #86); /* Nokta ve - X yönünü referans alan doğru */
#110 = LINE(#20, #90); /* Nokta ve - Z yönünü referans alan doğru */
```

Sonuç olarak kenarlar, ilgili köşeleri ve doğruları referans alarak tanımlanır. Örnek parçaya ait kenarlar Şekil B.8'de gösterilmiştir. EXPRESS dilinde kenar tanımı şu şekildedir:

```
*)
ENTITY edge
SUBTYPE OF (topology);
  edge_start : vertex;
  edge_end   : vertex;
  edge_curve : OPTIONAL curve_or_LOGICAL;
END_ENTITY;
(*
```

**edge\_start**, kenarın başlama noktası (köşe).

**edge\_end**, kenarın son noktası (köşe).

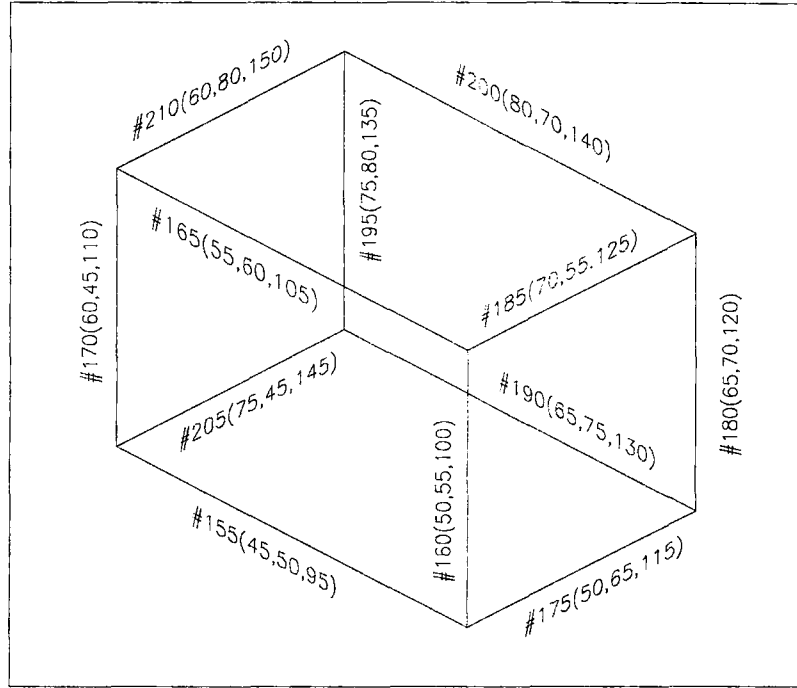
**edge\_curve**, isteğe bağlı eğri veya mantıksal varlık. Bu bir **eğri** veya **eğri\_mantıksal\_yapısı** olabilir.

Kenarlar fiziksel dosyaya aşağıdaki gibi yazılır:

```
#160 = EDGE(#50,#55,#100);
#155 = EDGE(#45,#50,#95);
#165 = EDGE(#50,#60,#105);
#170 = EDGE(#60,#45,#110);
```

**Kenar çevrimi:** Kenar çevrimi, **komşu kenarlar** veya **kenar\_mantık\_yapıları** (edge\_logical\_structures) listesiyle temsil edilir. Kenar çevrimi sınırsız bir yüzeyi sınırlandırmak için kullanılır.





Şekil B.8. Kenar tanımlama

Bir **kenar** listede verilirse, bu durum kenarın pozitif yönü ile kenar çevriminin pozitif yönünün aynı yönde olduğu anlamındadır. Eğer kenarın pozitif yönü, çevrimin pozitif yönüne karşı ise, o zaman bu FALSE mantık sembolü içeren **kenar\_mantık\_yapısında** ifade edilir. Şekil B.7'de iki komşu yüzey tarafından paylaşılan ve örnek parçayı oluşturan sekiz adet doğru, yönleriyle beraber tanımlanmıştır. Buna göre bir yüzeyin kenar çevrimi, ortak kenarların yönüyle ters olmadığı sürece, **kenar\_mantık\_yapısı** kullanılmaz. Örnek parçada Y1 yüzeyine ait kenarların yönleri, kenar çevrim yönü ile aynı olduğundan, burada **kenar\_mantık\_yapısı** kullanılmamıştır. Bu açıklamalara göre, tüm yüzeylere ait kenar\_mantık\_yapıları, Şekil B.9'da; kenar çevrimleri de, Şekil B.10'da verilmiştir. EXPRESS'te kenar çevrimi aşağıdaki şekilde yapılır.

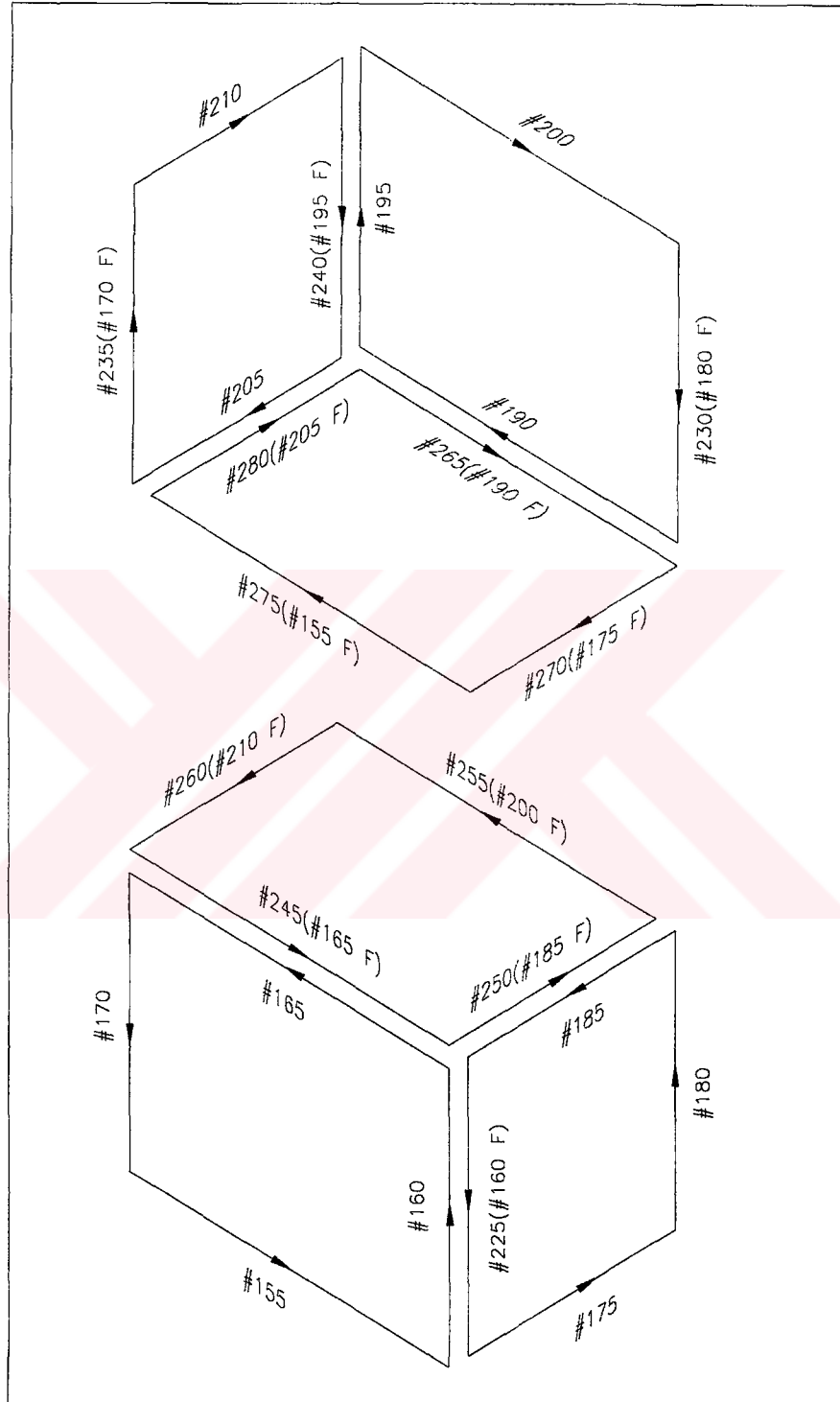
\*)

ENTITY edge\_loop

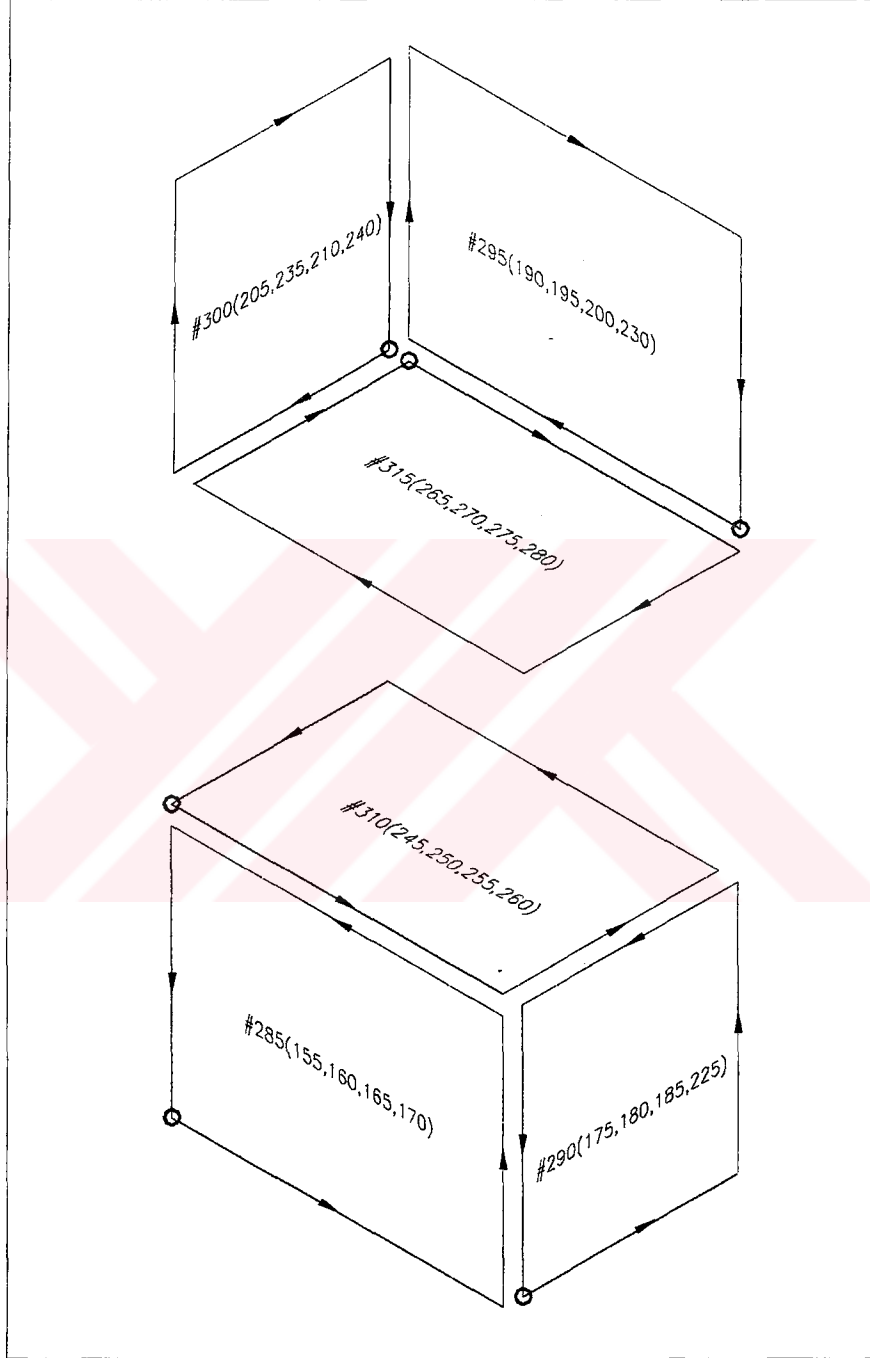
SUBTYPE OF (loop); loop\_edges: List [1: #] OF edge\_or\_logical;

END\_ENTITY;

(\*



Şekil B.9. Kenar mantık yapısı



Şekil B.10. Kenar çevrimi

**loop\_edges**, kenar veya mantıksal listedir. loop\_edges kenar veya kenar\_mantık\_yapısı listesi olarak yerine getirilir. Y1 yüzeyinin kenar çevrim varlığı, fiziksel dosyaya aşağıdaki şekilde yazılır.

```
#285 = EDGE_LOOP(#155,#160,#165,#170); /* kenarları referans alan
                                             kenar çevrimi*/
```

Örnek parçada Y6 yüzeyini meydana getiren tüm kenarlar, Şekil B.7'de tanımlanan yönlerle ters olduklarından, bunlar kenar\_mantık\_yapısında aşağıdaki şekilde temsil edilir.

```
#265 = EDGE_LOGICAL_STRUCTURE(#190,.F.);
#270 = EDGE_LOGICAL_STRUCTURE(#175,.F.);
#275 = EDGE_LOGICAL_STRUCTURE(#155,.F.);
#280 = EDGE_LOGICAL_STRUCTURE(#205,.F.);
```

Şekilde görüldüğü gibi her yüzeyin kenar çevrimi saat ibresinin tersine hareket ederek yüzeyin kenar çevrimini kapatmaktadır.

**Sınırsız düzlem yüzey ve yüzey temsil etme:** Geometrik bir yüzey, birden fazla yüzey tarafından kullanılabilir veya başka bir yüzey ile birleşik olabilir. **Sınırsız düzlem yüzey**, sınırsız yüzey üzerindeki bir nokta ve kendi normal ile tanımlanır. Bir düzlem yüzey, bir **nokta** ve bir **eksen** (axis2\_placement) varlığı ile temsil edilir.

```
*)
ENTITY plane
  SUBTYPE OF (elementary_surface);
  position: axis2_placement;
END_ENTITY;
(*)
```

**position**, yüzeyin uyumu ve konumudur.

```
*)
ENTITY axis2_placement
  SUBTYPE OF (axis_placement);
  location      : cartesian_point;
  axis          : OPTIONAL direction;
  ref_direction: OPTIONAL direction;
```

END\_ENTITY;  
 (\*

**location**, ilgili lokal kordinat sisteminin orijini ve referans noktasıdır.

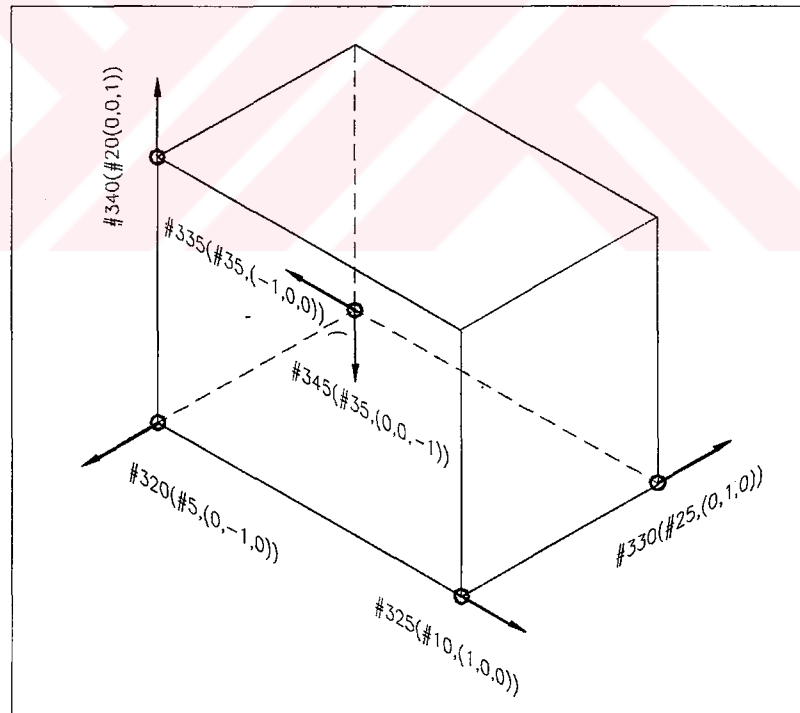
**axis**, lokal Z ekseninin yönüdür.

**ref\_direction**, lokal X ekseninin yönüdür.

Y1 yüzeyi için fiziksel dosyada düzlem yüzey tanımlama aşağıdaki şekilde yapılır.

```
#320 = PLANE(AXIS2_PLACEMENT(#5,(0,-1,0), ));
```

Burada Y1 yüzeyi -Y yönündedir. Şayet Y1 veya öteki yüzeylerden herhangi birisine bir özellik yerleşirse, **location**, **axis** ve **ref\_direction** parametreleri de kullanılır. Örnek parçanın yüzeylerine ait düzlemler Şekil B.11'de verilmiştir.



Şekil B.11. Düzlem tanımlama

**Bir düzlem yüzey**, kendi sınırlı çevrimi ile temsil edilir. Yüzeyin kenar çevrimi, daha önce açıklandığı gibi, yüzeyin çevresinde saat ibresinin tersine hareket eder. Her yüzey dış sınır, sınırlar, sınırlı ve sınırsız yüzeyler takımı ile tanımlanır. Dış sınır, yüzeyin dışındadır. EXPRESS'te yüzey varlığı aşağıdaki şekilde tanımlanır:

```

*)
ENTITY face
SUBTYPE OF (topology);
  outer_bound : OPTIONAL loop_or_logical;
  bounds      : SET [1:#] OF loop_or_logical;
  face_surface : OPTIONAL surface_or_logical;
END_ENTITY;
(*)

```

**outer\_bound**, yüzeyin dış sınırındadır.

**bounds**, yüzeyin tüm sınırlarının listesidir.

**face\_surface**, yüzeyin altında uzanan geometrik yüzeydir.

Y1 yüzeyi fiziksel dosyaya yazıldıktan sonra aşağıdaki şekilde görülür.

```

#350 = FACE (#285,(#285),#320); /*Üç nitelikli yüzey tanımlama:
      dış sınır, sınırlar listesi ve yüzey altındaki düzlem yüzey*/

```

Y5 yüzeyinin tanımı, üzerine yerleşen özellikleriyle birlikte aşağıdaki şekildedir.

```

#435 = FACE
SCOPE
  #440 = FORM_FEATURE('HOLE',(#375), , );
  #445 = FORM_FEATURE('STEP',(#400), , );
  #450 = FORM_FEATURE('STEP',(#425), , );
ENDSCOPE
      (#310,(#310),#340);

```

### 1.1. Şekil Özelliği Tanımlama

Dört niteliğe sahip bir şekil özelliği, EXPRESS dilinde aşağıdaki şekilde tanımlanır.

```

*)
ENTITY form_feature

```

```

feature_type : STRING;
implicit_reps : SET [0:#] OF implicit_form_feature;
pattern_rep : OPTIONAL implicit_form_feature_pattern;
replicate_rep : OPTIONAL replicate_form feature;
END_ENTITY;
(*)

```

**feature\_type**, özelliğin (havşa, delik, kanal, kademe, cep vs) tipinin spesifikasyonudur.

**implicit\_reps**, özelliğe ait kapalı temsilin koleksiyonudur.

**pattern\_rep**, özelliğin, kapalı şekil özellik model temsili gösterir.

**replicate\_rep**, özelliğin replicate temsilidir.

Delikler, kademeler, kanallar ve oyuklar gibi çeşitli özellikler STEP'te **Özellik hacmi** temsili kullanılarak tanımlanır. İlave edilecek malzemenin artışı veya eksilişi olarak modellenen şekil özelliklerinin kapalı tanımlarını belirlemek için, ilk şekle eklenen veya çıkarılan bir hacim kullanılır. Özellik hacmi, aşağıda gösterildiği gibi, özellik süpürme ve özellik yönlendirmeden oluşur.

```

*)
ENTITY feature_volume
  SUPERTYPE OF (feature_sweepXOR feature_ruling);
END ENTITY;
(*)

```

**feature\_ruling**, iki eğrinin bir rulingidir. Bir şekil özelliğinin kapalı bir temsilidir. Feature ruling kapalı bir pasaj, çıkıntı veya girintiyi belirlemek için kullanılır.

**feature\_sweep**, 2 1/2 boyutlu bir şekil, bir düzlem profil ile bu profilin süpüreceği uzunlamasına bir yol ve bir veya iki uçlu şekil tanımlama içerirse, özellik süpürme bir prosedüral tanımdır.

Profil ile ilgili yolun uyumu aşağıdaki şekilde yerine getirilir.

1. Özellik yolunun, Lokal Koordinat Sistemiyle (LKS) ilgili uyumu: Doğrusal süpürme yolu, LKS'nin orijininde başlar ve pozitif Z eksenine boyunca uzanır.

2. Profiller, AB uzayı olarak adlandırılan iki boyutlu uzayda tanımlanır. Her profil tipi, kendi AB uzayında önceden tanımlanan konuma ve uyuma sahiptir. Örneğin bir dikdörtgen profil, A eksenine paralel bir çift kenar belirlemeyle, AB orijininde kendi merkezine sahiptir.

3. Profilin A ve B eksenlerinin özelliğın LKS'sinde yerleřtirilmesi řöyle olur. Doğrusal bir yol için profilin A ve B eksenleri, sırasıyla, LKS'nin X ve Z eksenlerine yerleřtirilir.

**Süpürme**, süpürülen girintiyi ve kapalı pasajlar için çıkarılan veya kapalı çıkıntılar için eklenen hacmi tanımlar. EXPRESS'te özellik süpürme ařağıdaki gibi ifade edilir.

```

*)
ENTITY feature_sweep
  SUPERTYPE OF (along_feature_sweep XOR
                 axisymmetric_feature_sweep XOR
                 in_out_feature_sweep)
  SUBTYPE OF (feature_volume);
  location : axis2_placement;
END_ENTITY;
(*

```

**location**, özellik süpürme yerini belirlemek için eksen yerleřtirir.

### **Delik Özelliğı Tanımlama**

Bir delik özelliğı **simetri\_eksenli\_özellik\_süpürme**de tanımlanır. Simetri eksenli özellik süpürme, düzleme dik eksen etrafında, düzlem eğriyi 360° döndürmeyle meydana getirilir. Süpürme eğrisi, kapalı veya açık tanımlanabilir. Açık tanımlama, başka simetri eksenli özellik süpürme yolu ile elde edilebilir. Kapalı eğriler **sabit çap**, **simetri eksenli özellik süpürme** ve **konikleřtirilmiş simetri eksenli özellik süpürme** ile belirlenir. Simetri eksenli özellik süpürmenin kısıtları řunlardır:

1. Süpürme eksenini, özelliğın LKS'nin Z eksenini ile çakıřır.
2. Eğri herhangi bir uzunlukta olabilir.
3. Simetri eksenli süpürmeler için tüm profiller, lokal AB uzayında tanımlanır. Profilin lokal A ve B eksenleri, sırasıyla özelliğın LKS'nin X ve Z eksenlerine yerleřtirilir.
4. Süpürme eğrisi uçlarının, simetri eksenli özellik süpürme ucu belirlenmedikçe, düzlem olacağı kabul edilir.



Eğer herhangi bir simetri eksenli özellik süpürme ucu verilirse, bu  $Z=0$ 'dan  $Z$  yönünde uzatılır. Simetri eksenli özellik süpürme aşağıdaki gibi tanımlanır.

```

*)
ENTITY axisymmetric_feature_sweep
SUPERTYPEOF(constant_diameter_axisymmetric_feature_sweep XOR
            tapered_axisymmetric_feature_sweep XOR
            other_axisymmetric_feature_sweep)
SUBTYPE OF (feature_sweep);
sweep_length : size_paremeter;
sweep_end    : OPTIONAL_axisymmetric_feature_sweep_end;
END_ENTITY;
(*

```

**sweep\_length**, süpürülen profilin uzunluk ölçüsüdür.

**sweep\_end**, istenen uç şeklinin tanımlanmasıdır.

**Sabit çaplı simetri eksenli özellik süpürme:** Bu profil, B eksenine paralel bir doğrudan oluşan, simetri eksenli özellik süpürme profilidir. Lokal B eksenine ile özelliğin Z eksenine çakıştıktan sonra, doğru, bir silindiri oluşturmak üzere Z eksenine etrafında döndürülür (Şekil B.12).

Sabit çaplı simetri eksenli özellik süpürme, EXPRESS'te aşağıdaki gibi tanımlanır.

```

*)
ENTITY(constant_diameter_axisymmetric_feature_sweep
SUPERTYPE OF (axisymmetric_feature_sweep);
sweep_size : size_paremeter;
END_ENTITY;
(*

```

**sweep\_size**, süpürülen özelliğin çapıdır.

Simetri eksenli özellik süpürme ile tanımlanan kapalı şekil özelliğinin (örneğin bir delik gibi) uç şekli, **simetri eksenli özellik süpürme** ucu ile belirlenir. Bu, düz, konik ve küresel uçlu bir delik için ortak uç şekillerini modellemek üzere kullanılabilir. Uç,  $Z=0$ 'da özelliğinin süpürülen gövdesine birleşir ve -  $Z$  yönünde uzatılır.

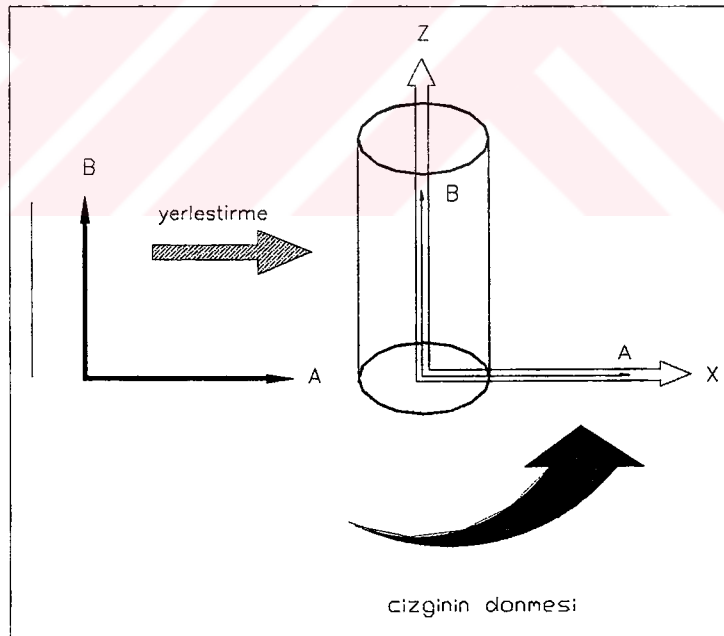
Simetri eksenli özellik süpürme ucu verilmediği zaman, özelliğin düz uclu simetri eksenli özellik süpürme olduğu kabul edilir. Düz bir uç, bir kenar birleşme yerine sahip ise, bu açık olarak modellenmelidir. Uç şekli aşağıdaki şekilde belirlenir.

```

*)
ENTITY axisymmetric_feature_sweep_end
SUPERTYPE OF (axisymmetric_feature_sweep_flat_end XOR);
                axisymmetric_feature_sweep_spherical_end XOR
                axisymmetric_feature_sweep_conical_end);
blend : OPTIONAL implicit_edge_blend;
END_ENTITY;
(*

```

**Simetri eksenli düz uçlu özellik süpürme:** Simetri eksenli özellik süpürmenin uç şekli düz olup, bu, aşağıdaki şekilde temsil edilir. Simetri eksenli düz uçlu özellik süpürme düzleminde lokal,  $Z=0$ 'dır.



Şekil B.12. Bir çizginin dönmesiyle silindir veya deliğin elde edilmesi

```

*)
ENTITY axisymmetric_feature_sweep_flat_end
  SUBTYPE OF (axisymmetric_feature_sweep_end);
END_ENTITY;
(*

```

**Simetri eksenli konik uçlu özellik süpürme:** Bir konik ucu modelleyen tanımlama aşağıdaki gibi temsil edilir.

```

*)
ENTITY axisymmetric_feature_sweep_conical_end
  SUBTYPE OF (axisymmetric_feature_sweep_end);
  end_angle : angle_paremeter;
  tip_blend : OPTIONAL implicit_edge_round;
WHERE
  WR1 : 0 < end_angle_dimension < 180;
(*

```

**end\_angle**, kapalı koninin derece olarak tam veya yarım açısıdır.

**tip\_blend**, isteğe bağlı koninin uç tipidir.

**WR1**, koni açısının  $0^0$  ile  $180^0$  arasında olması gerektiğini ifade eder.

Varlıklar bir deliğin modellenmesi için EXPRESS'te tanımlandıktan sonra, fiziksel dosyaya yazılır. Örneğin Şekil B.13'teki deliğin fiziksel dosyada temsil edilmesi aşağıdaki şekilde yapılır.

```

#370= AXIS2_PLACEMENT ((60,40,0),(0,0,1),(1,0,0)); /* Deliği
                                     yerleştirmek için Z ve X eksen yönleri*/
#375= FEATURE_SWEEP
SCOPE
  #382= AXISYMMETRIC_FEATURE_SWEEP
SCOPE
#384=CONSTANT_DIAMETER_AXISYMMETRIC_FEATURE_SWEEP(20);
ENDSCOPE
(60,); /* 60 süpürme uzunluğudur. Simetri eksenli özelliğin ucu
tanımlanmamıştır*/
ENDSCOPE
(#370);

```

Şekil özellik varlığı, **özellik\_süpürme** ve ilgili varlıkları referans aldığından, bu varlıklar tanımlandıktan sonra, şekil özellik varlığı tanımlanır.

FORM\_FEATURE varlığı fiziksel dosyaya şu şekilde yazılır.

```
#440= FORM FEATURE('HOLE',(#375),) /* İsteğe bağlı şekil özellik
modeli ve şekil özelliğinin
replicati tanımlanmamıştır*/
```

Şekil B.13'te gösterilen delik boydan boyadır. Delik kör ve konik uçlu ise tanımlanması aşağıdaki gibi olur.

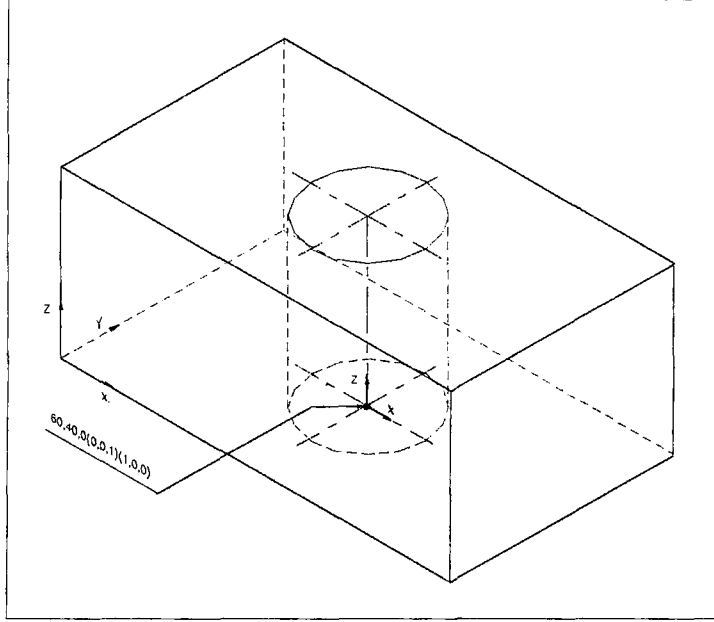
```
#350= AXIS2_PLACEMENT ((30,25,30),(0,0,1),(1,0,0));
#355= AXISYMMETRIC_FEATURE_SWEEP_END
SCOPE
#360= AXISYMMETRIC_FEATURE_SWEEP_CONICAL_END(118,);
ENDSCOPE
();
```

```
#365= FEATURE_SWEEP
SCOPE
#372= AXISYMMETRIC_FEATURE_SWEEP
SCOPE
```

```
#374=CONSTANT_DIAMETER_AXISYMMETRIC_FEATURE_SWEEP(18);
ENDSCOPE
(35,#355);
ENDSCOPE
(#350);
```

### **Kademe Özelliği Tanımlama**

Ayrıca kademe (L) özellikleri, **özellik\_süpürme** varlığını kullanarak **özellik\_hacmi** bölümünde de tanımlanır. Boydan Boya Özellik Süpürme, özellik\_süpürmenin bir alt tipi olup ilk şeklin sınırları içerisinde bulunan (hava/malzeme arayüzünde) bir yoldaki özellik süpürmedir. Kademe, kanal gibi bazı ortak özellikler bu şekilde tanımlanır.



Şekil B.13. Örnek parçaya deliğin yerleştirilmesi

Boydan boya özellik süpürme varlığının EXPRESS'te tanımlanması aşağıdaki gibidir.

\*)

```
ENTITY along_feature_sweep
SUBTYPE OF (feature_sweep);
  sweep_path : feature_sweep_path;
  sweep_profile : open_feature_sweep_profile;
  sweep_ends : SET [0:2] OF along_feature_sweep_end;
END_ENTITY;
(*
```

**sweep\_path**, süpürmeyi belirleyen yoldur.

**sweep\_profile**, yol boyunca süpürülecek profilin özelliğidir.

**sweep\_ends**, süpürülen şeklin uç tipini açıklar.

**Boydan boya özellik süpürme ucu:** Kapalı şekil özelliğinin uç şekli, boydan boya özellik süpürme ucu olarak adlandırılan varlıkla tanımlanmıştır. Bu, sadece süpürme ucu olduğu zaman kullanılabilir. Eğer özellik süpürme işlemi parçanın bir ucundan girip öteki ucundan çıkıyorsa, herhangi bir özellik süpürme ucu söz konusu değildir. Bir örnek olarak bu varlık, frezelenmiş kanalın

yuvarlatılmış ucunu modellemek için kullanılabilir. Özellik süpürme işlemi hava/malzeme sınırında son bulmadığı ve boydan boya bir özellik süpürme işlemi olmadığı zaman, özelliğin düz bir uca sahip olduğu kabul edilir. Boydan boya özellik süpürme ucu aşağıdaki şekilde tanımlanır.

```
*)
ENTITY along_feature_sweep_end
  SUPERTYPE OF (along_feature_sweep_flat_end XOR);
  along_feature_sweep_radiused_end);
  sweep_end : feature_end_types;
END_ENTITY;
(*)
```

**sweep\_end**, başlangıç veya sonda bir uç olup olmadığını gösterir.

**Boydan boya özellik süpürme (düz uçlu):** Boydan boya özellik süpürme ucunun düzlem olduğunu göstermek için kullanılır.

```
*)
ENTITY along_feature_sweep_flat_end
  SUBTYPE OF (along_feature_sweep_end);
  end_blend: OPTIONAL implicit_edge_blend
END_ENTITY;
(*)
```

**end\_blend**, istenen uç birleşme yerinin tanımıdır.

**Boydan boya özellik süpürme (radyüs uçlu):** Bir radyüs ile tanımlanır. Ucu şekli kullanılan açık özellik süpürme profilinin tipine bağlıdır.

```
*)
ENTITY along_feature_sweep_radiused_end
  SUBTYPE OF (along_feature_sweep_end);
END_ENTITY;
(*)
```

**Özellik süpürme yolu:** Açık form özelliğinin şeklini tanımlamak için boydan boya süpürülen bir özellik süpürme profili eğrisidir.

```

*)
ENTITY feature_sweep_path
  SUPERTYPE OF (circular_feature_sweep_path XOR
                 spiral_feature_sweep_path XOR
                 surface_conforming_feature_sweep_path_XOR
                 other_feature_sweep_path_XOR
                 linear_feature_sweep_path);
END_ENTITY;
(*)

```

**Doğrusal özellik süpürme yolu:** Doğrusal bir çizgi olarak tanımlanır. Özelliğin LKS'nin +Z eksenini boyunca,  $Z=0$ 'dan  $Z=L$ 'ye kadar uzanır. Burada  $L$ = yol uzunluğudur. Doğrusal süpürmede kullanılan profillerin A ve B eksenleri sırasıyla özelliğin LKS'nin X ve Y eksenlerine yerleştirilir. Yol boydan boya ise özellik süpürme, tanımlanan boydan boya özellik süpürme uçlarına sahiptir. Bu durumda uçlar  $Z=0$ 'dan  $-Z$  yönünde veya  $Z=L$ 'den  $+Z$  yönünde uzanır.

```

*)
ENTITY linear_feature_sweep_path
  SUBTYPE OF (feature_sweep_path);
  path_lenght : size_parameter;
END_ENTITY;
(*)

```

**path\_lenght**, süpürme yolunun uzunluk boyutudur.

**Özellik süpürme profili:** Bir şekil özelliğini tanımlamak için, bir özellik süpürme yolunun, alan boyunca süpürülmesiyle oluşan bir düzlem eğri veya birleşik eğriler setidir. Profiller, yolun tipine göre özelliğin LKS'ne yerleştirilen lokal AB uzayında tanımlanır. AB uzayında profilin uyumu, profilin tipine göre belirlenir.

```

*)
ENTITY feature_sweep_profile
  SUPERTYPE OF (closed_feature_sweep XOR
                 open_feature_sweep_profile);
END_ENTITY;
(*)

```

**Açık özellik süpürme profili:** Kapalı olmayan eğri veya bileşik eğriden meydana gelen özellik süpürme profilidir. Profilin açık uçlarının malzeme/hava arayüzüne kadar uzatıldığı kabul edilir. Örneğin "U" şeklindeki bir profil için, sadece genişlik ve kenar birleşme yerleri tanımlanır. Düşey kolların uzunluğu profilin yerleştirilmesiyle belirlenir.

\*)

```
ENTITY open_feature_sweep_profile
  SUPERTYPE OF (circular_arc_feature_sweep_profile XOR
    rounded_u_feature_sweep_profile XOR
    vee_feature_sweep_profile XOR
    square_u_feature_sweep_profile XOR
    tee_feature_sweep_profile XOR
    ell_feature_sweep_profile XOR
    line_feature_sweep_profile XOR
    half_obround_feature_sweep_profile XOR
    other_open_feature_sweep_profile);
  SUBTYPE OF (feature_sweep_profile);
END_ENTITY;
```

(\*

**L (Kademe) özellik süpürme profili:** L şeklindeki özellik süpürme profilidir. Bu bir dikey **endbar** ve **stem**'i içerir. Stem, endbarla uçları tespit edilmiş iki paralel, yarı sonsuz doğru tarafından temsil edilir. Endbar bir dikdörtgendir. Endbar, stem'e zıt kenarlı pozitif B yarım düzleminde ve A eksenini üzerindedir. L profil niteliği, öncelikle endbar'ın pozitif A veya negatif A yarım düzleminde olup olmadığını gösterir (Şekil B.14).

Endbar'ın yüksekliği ve stem'in genişliği sıfıra eşit olduğunda, L özelliğini yerleştirme Şekil B.15'te görüldüğü gibi yapılır.

L özellik süpürme profili EXPRESS'te şu şekilde yazılır.

\*)

```
ENTITY ell_feature_sweep_profile
  SUPERTYPE OF open_feature_sweep_profile);
  stem_width      : size_parameter;
  endbar_width    : size_parameter;
  endbar_height   : size_parameter;
  ell_orientation : ell_orientation_types;
  stem_endbar_blend : OPTIONAL implicit_edge_blend;
```



```

endbar_blend1 : OPTIONAL implicit_edge_blend;
endbar_blend2 : OPTIONAL implicit_edge_blend;
endbar_blend3 : OPTIONAL implicit_edge_blend;
END_ENTITY;
(*)

```

**stem\_width**, stem'in genişliğidir(X boyutu).

**endbar\_width**, endbarın genişliğidir(X boyutu).

**endbar\_height**, endbarın yüksekliğidir(Y boyutu).

**ell\_orientation**, endbarın ucunun pozitif veya negatif X yarım düzleminde olup olmadığını gösterir.

**stem\_endbar\_blend**, stem ve endbar arasında kalan konkav köşenin kavisidir. endbar\_blend1, endbar\_blend2, endbar\_blend3 L'nin köşelerinin birleşim yeridir.

#### L- uyum tipleri aşağıdaki şekilde tanımlanır:

```

*)
TYPE ell_orientation_types = ENUMERATION OF
    (ell_positive,
    ell_negative);
END_ENTITY;
(*)

```

Kademe özelliği için varlık bilgisi, fiziksel dosyaya yazılmadan önce, özellik Şekil B.16'da görüldüğü gibi, LKS'ye planlanır.

Fiziksel dosyada kademe özelliği için şekil özelliği tanımlama aşağıdaki gibidir.

```

#385= AXIS2_PLACEMENT ((30,0,40),(0,1,0),(-1,0,0));
#390= LINEAR_FEATURE_SWEEP_PATH (80);
#395= ELL_FEATURE_SWEEP_PROFILE(0,30,0, ELL_POSITIVE., , , );
/*isteğe bağlı 5., 6., 7., 8. özellik süpürme profili nitelikleri
tanımlanmamıştır */

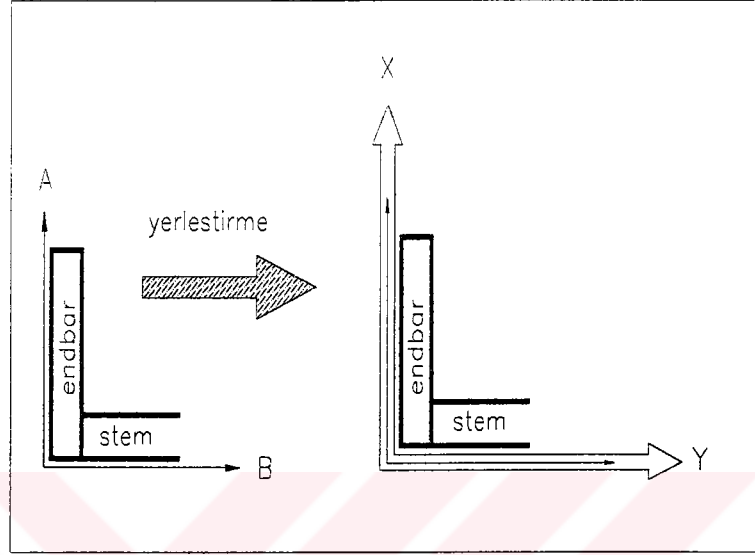
#400= FEATURE_SWEEP
SCOPE
    #405= ALONG_FEATURE_SWEEP(#390,#395,()); /* uzunlamasına
özelliğ süpürme ucu seti boş */
ENDSCOPE

```

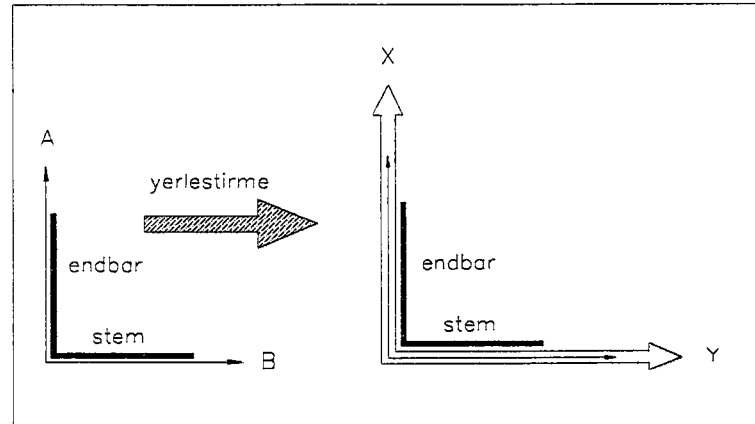
(#385);

.....

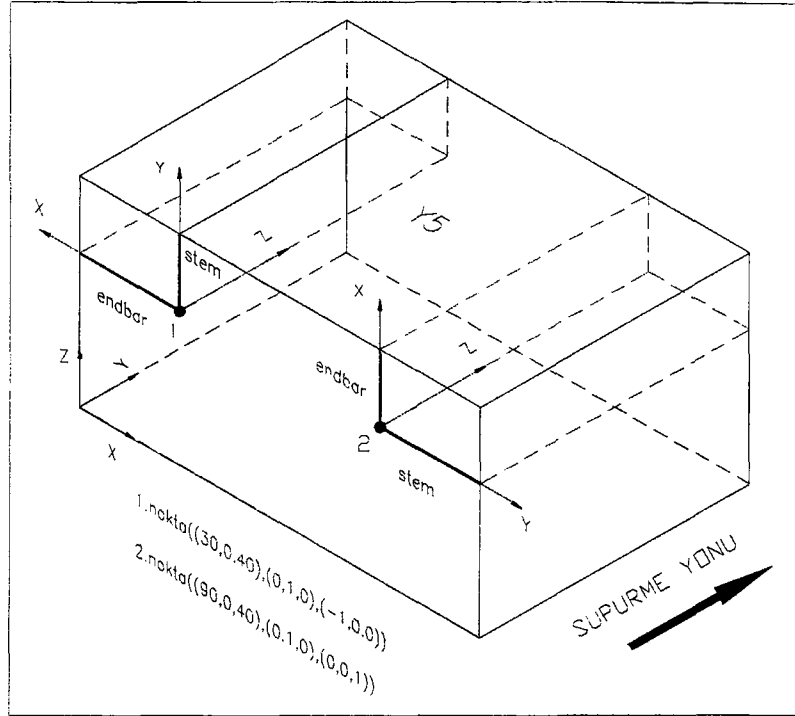
#445 = FORM\_FEATURE('STEP',(#400),,);



Şekil B.14. Kademe özelliğinin A-B ekseninden X-Y eksenine yerleştirilmesi



Şekil B.15. Endbar'ın yüksekliği ve stem'in genişliği sıfır olduğunda kademe özelliğinin yerleştirilmesi



Şekil B.16. Örnek parçada L kademe özelliği süpürme

### Kanal Özelliği Tanımlama

Kanal özelliği tanımlama, süpürme profili dışında, kademe özelliğine benzer şekilde yapılır. Kanal özelliği; `feature_sweep`, `along_feature_sweep`, `linear_feature_sweep_path` ve `axis2_placement` varlık tanımlamaları ile temsil edilir. Bu nedenle sadece süpürme profilinin temsil edilmesi açıklanmıştır. Kademe özelliğinin süpürme profili varlığı **`square_u_feature_sweep_profile`** veya **`rounded_u_feature_sweep_profile`** adıyla belirlenir.

**Kare U özellik süpürme profili:** İki paralel yarı sonsuz doğrunun sabit uçları arası, bu paralel doğrulara dik bir doğru ile bağlanmış olan açık bir özellik süpürme profilidir. Bu, A eksenı üzerindeki kendi doğru parçası ile pozitif B yarım düzleminde uzanır. Şekil B.18'de görülen Y5 yüzeyine orta noktası orijinde olan bir U kanalı yerleştirmek için, STEP varlık tanımlamaları kullanılır. Önce süpürme özelliği A-B eksenine yerleştirilir ve sonra bu özellik LKS'ye aktarılır (Şekil B.17). Varlık tanımı aşağıdaki şekilde yapılır.

```

*)
ENTITY square_u_feature_sweep_profile
  SUPERTYPE OF open_feature_sweep_profile);
  sweep_width      : size_parameter;
  blend1           : OPTIONAL implicit_edge_blend;
  blend2           : OPTIONAL implicit_edge_blend;
END_ENTITY;
(*

```

**sweep\_width**, doğru parçasının uzunluğu.

**blend1**, tabanda sol yan köşedeki birleşme yeri.

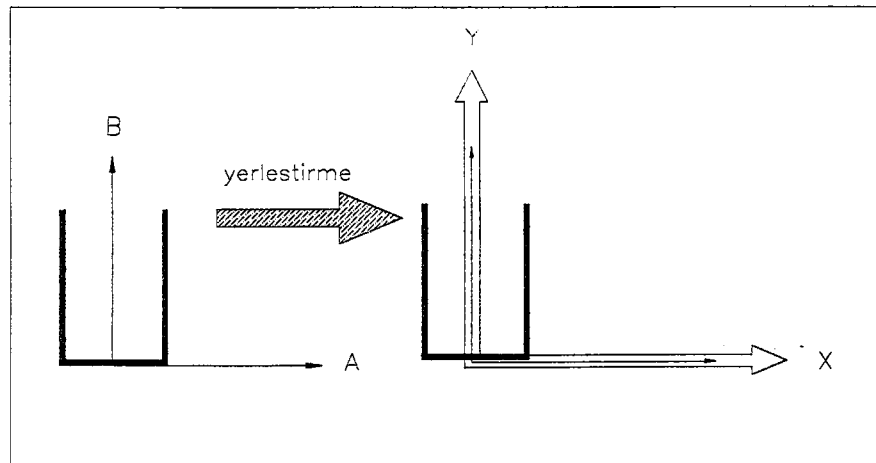
**blend2**, tabanda sağ yan köşedeki birleşme yeri.

Fiziksel dosyada bir kanal özelliğinin temsil edilmesi aşağıdaki gibi yerine getirilir.

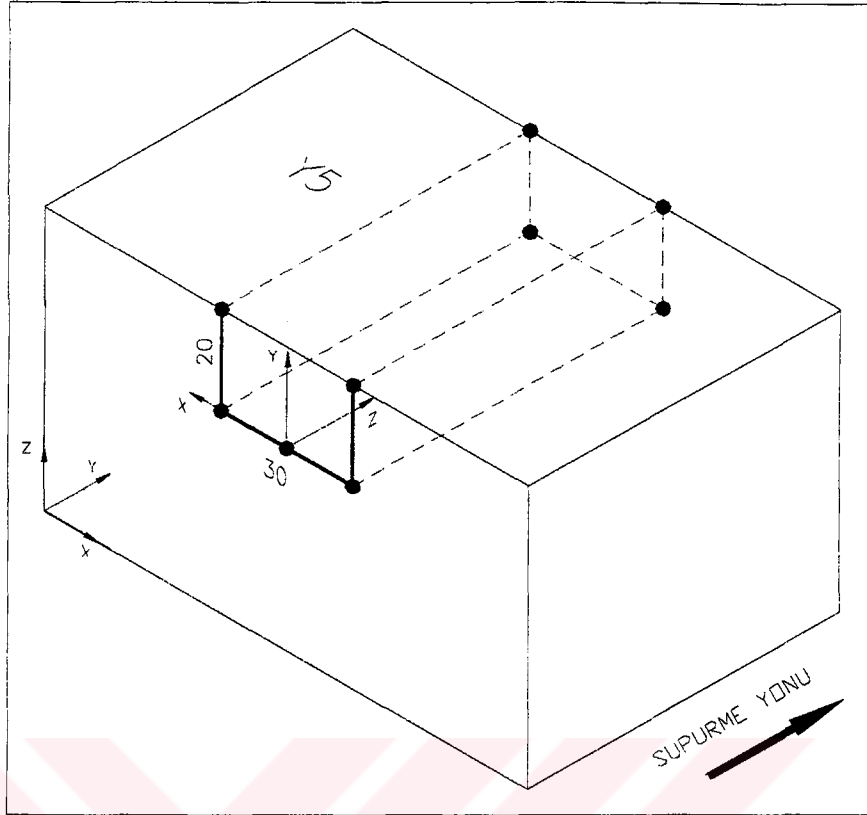
```

#480= AXIS2_PLACEMENT ((75,0,40),(0,1,0),(-1,0,0));
#485= LINEAR_FEATURE_SWEEP_PATH (70);
#490= SQUARE_FEATURE_SWEEP_PROFILE (30, , );
#495= FEATURE_SWEEP
  SCOPE
    #500= ALONG_FEATURE_SWEEP (#485,#490,());
  ENDScope
  (#480);
.....
#510 = FORM_FEATURE('SLOT',(#495), , );

```



Şekil B.17. Bir U kanalı şekillendirmek için yerleştirme



Şekil B.18. A-B ekseninden X-Y eksenine kanal özelliği profili yerleştirme

## 1.2. Tolerans ve Boyutlar

### 1.2.1. Tolerans

Tolerans mutlak bir değerden sapmaları gösterir. Boyut değeri geometrik modelde açık bir şekilde verilir. Her tolerans, üzerinde boyutları bulunan bir model içinde belirtilir. Tolerans değerleri şekil/ölçü elemanlarından çıkarılır veya kapalı özelliklerin kısmı olarak açıkça ifade edilir.

**Boyut ölçme:** Boyutlar bazen bir özelliğin tam ölçüsünü (örneğin bir silindirin çapı gibi) ve bazen de yarı ölçüsünü ifade eder (örneğin koni yarı açısı gibi). Boyut özelliği aşağıda verilen **tip** tanımı ile yerine getirilir. Bu tip, boyutun tam veya yarı olup olmadığını ifade eder.

```

*)
TYPE dimension_measurment = ENUMERATION OF
    (full; half);
END_TYPE;
(*)

```

Malzeme durum düzenleyici (tol\_mlsn), toleranslandırılan özelliğin belirlenen ölçüsünü esas alarak tolerans bölgesinin ölçüsünü etkileyen bir işarettir.

```

*)
TYPE tol_mlsn = ENUMERATION OF
    maxme,
    leastme,
    regardless,
    none,
END_TYPE;
(*)

```

**maxme**, tolerans sınırları içerisinde olmak üzere delik en küçük ve mil en büyük ölçüsünde olduğu zaman maksimum malzeme durumu (en küçük boşluk) dur.

**leastme**, delik en büyük ve mil en küçük ölçüsünde olduğu zaman minimum malzeme durumu (en büyük boşluk) dur.

**regardless**, özellik ölçüsü kayıt edilmemiştir.

**none**, uygulanamaz olduğunu ifade eder.

**Tolerans aralığı**, müsaade edilebilen üst ve alt sınır sapması toplamı, tolerans aralığını belirler.

```

*)
ENTITY tolerance_rang
    SUPERTYPE OF (coordinate tolerance_range XOR
        NULL);
    plus_tol    : REAL;
    minus_tol   : REAL;
WHERE
    WR1 : plus_tol >= 0
    WR2 : minus_tol =< 0
    WR3 : plus_tol + minus_tol > 0;
END_ENTITY;
(*)

```

**plus\_tol**, müsaade edilen maksimum ölçüyü oluşturmak için, nominal değere ilave edilen toleransın mutlak değeridir.

**minus\_tol**, müsaade edilen minimum ölçüyü elde etmek için, nominal değerden çıkarılan toleransın mutlak değeridir.

**Geometrik tolerans** aşağıdaki gibi tanımlanır:

\*)

```
ENTITY geometric_tolerance
  SUPERTYPA OF (angularity XOR
                circular_runout XOR
                circularity XOR
                concentricity XOR
                cylindricity XOR
                flatness XOR
                parallelism XOR
                perpendicularity XOR
                position XOR
                profile_line XOR
                profile_surface XOR
                straightness XOR
                total_runout);
  SUBTYPE OF (shape_tolerance);
END_ENTITY;
(*
```

**Dairesellik toleransı:** Dairesellik, dönel bir yüzeyin durumudur. Bir düzlemlle kesilen dairesel yüzeyin bütün noktaları, dönel yüzeyin merkezine eşit mesafededir. Dairesellik toleransı, toleranslandırılan özelliğin koordinatları içerisindeki iki konsantrik (eksantrik olmayan) daire arasındaki mesafe olarak tanımlanır.

\*)

```
ENTITY circularity
  SUBTYPE OF (geometric_tolerance)
    toleranced_ents : SET [1:#] OF area_or_seam_or_fos;
    magnitude       : tolerance_magnitude
  WHERE
    WR1 : is_circular(toleranced_ents);
END_ENTITY;
(*
```

**toleranced\_ents**, alan ve ek yeri şekil elamanı ile özellik ölçüsünden oluşan tolerans uygulanan varlık elamanları setidir. Alan, ek yeri ve köşe terimleri uygun şekil elemanlarını göstermek için kullanılır ve kabaca yüzey, kenar ve köşenin ST notasyonuna karşılık gelir.

**magnitute**, toleransın büyüklüğüdür.

**Doğrusallık toleransı:** Doğrusallık toleransı, sınırsız bir yüzey özelliğine ait doğru parçalarının veya bir doğrunun müsaade edilen sapmalarını tanımlar. Sınırsız yüzey özellikleri için doğrusallık toleransı, iki paralel doğru arasındaki mesafeyi belirler.

\*)

ENTITY straightness

SUBTYPE OF (geometric\_tolerance)

toleranced\_ents : SET [1:#] OF area\_or\_seam\_or\_fos;

magnitute : tolerance\_magnitute;

aplication\_direction : OPTIONAL direction;

material\_condition : tol\_msln;

cylindrical\_zone : LOGICAL;

per\_unit\_length : REAL;

END\_ENTITY;

(\*

**toleranced\_ents**, tolerans uygulanan bir varlıklar kümesidir.

**magnitute**, toleransın büyüklüğüdür.

**aplication\_direction**, doğrusal yüzey elemanı için doğrusallık toleransını belirleyen yöndür.

**material\_condition**, toleransı uygulamada maksimum, minimum, kayıt edilmemiş ve uygulanamaz malzeme durumlarını gösteren sayısal bir listedir.

**cylindrical\_zone**, bir Boolean (true/false) sembolü bir eksen veya çizgi içindeki silindirik bölgenin tolerans değerini gösterir. Eğer false ise, bölge, iki paralel doğru veya düzlem arasındaki boşluktur.

**per\_unit\_length**, tolerans değerinin uygulandığı doğrusal mesafeyi belirler.

### 1.2.2. Boyutlar

Boyutlar, teorik olarak modellenen cisimlerin gerçek şeklini tanımlar. Koordinat boyutu, imalat resimlerinde bulunan boyutlara karşılık gelir. Bu, açı ve ölçü boyutunun bir sınıflandırmasıdır.



```

*)
ENTITY coordinate_dimension
  SUPERTYPE OF (angle_dimension XOR
                location_dimension XOR
                size_dimension);
END_ENTITY;
(*)

```

**Yerleştirme boyutu:** Yerleştirme boyutu, diğer özelliklerle ilgili bir özelliğin yerleştirilmesinin ölçümüdür. Yerleştirme boyutunun değeri, iki paralel şekil temsil elemanı arasındaki mesafedir veya şekil temsil elemanlarından (düzlem olanların herhangi bir bileşimi, doğrusal birleşme yeri veya köşe veya sabit mesafe ile ayrılmış iki eleman) çıkarılan değerlerdir. Yerleştirme boyutundaki her özellik iki karakteristikten oluşur. Bunlar:

- 1- Boyutun orijini olan şey.
- 2- Kontrol edilen cismin veya yüzeyin boyutu üzerindeki temsil elemanın gerçek şekli (yüzey/alan, eğri/birleşme yeri veya nokta/köşe). Boyutun değeri doğrusal bir yol boyunca ölçülür.

```

*)
ENTITY location_dimension
  SUBTYPE OF (coordinate_dimension);
  range      : OPTIONAL coordinate_tolerance_range;
  target     : dt_feature;
  origin     : dt_feature;
  directed   : LOGICAL;
  linear     : LOGICAL;
  measure_path : OPTIONAL geometric_divation;
END_ENTITY;
(*)

```

**range**, tolerans aralığının isteğe bağlı değeridir.

**target**, boyut özelliği için gereken bir özelliktir (boyut/tolerans).

**origin**, boyut özelliğinden elde edilen bir özelliktir.

**directed**, ölçüm, orijinden hedefe doğru ise TRUE; orijin ve hedef arasında bir öncelik yoksa, FALSE' dir.

**linear**, ölçme yolu doğrusal ise TRUE'dır.

**measure\_path**, ölçümün yapıldığı yolun boyudur (örneğin non-lineer eğri).

**Ölçü boyutu:** Ölçü boyutu, bir cismin şeklinin sadece bir özelliğinin ölçüsüdür. Tolerans, sayısal bir aralık olup boyutun değerini kısıtlar. Bir ölçü boyutunun 3 karakteristiği vardır.

- 1- Boyutsal değer (temsili geometri içerisinde kapalı bir şekilde tanımlanır veya açıkça ifade edilir),
- 2- İlgili özelliği tanımlayan temsil elemanlarının şekli,
- 3- Özelliğin elemanlarına ait simetri merkezi.

Bir ölçü boyutunun değeri, bir büyüklük olarak ifade edilir ve özelliğin yerleşiminden bağımsızdır. Ölçü boyutunun değeri, simetri merkezinden geçen özellik üzerindeki bir noktadan, bu noktanın karşısında ve özellik üzerinde bulunan başka bir noktaya, doğrusal olarak ölçmek suretiyle belirlenir. Bir çok durumda bir deliğin çapı veya bir kanalın genişliğinin ölçü boyutunun belirlenmesi önemsizdir.

```

*)
ENTITY size_dimension
  SUPERTYPE OF (size_charecteristik_dimension XOR
                size_paremeter_dimension);
  SUBTYPE OF (coordinate_dimension);
  size_ent      : dt_feature;
  range         : OPTIONAL coordinate_tolerance_range;
  measurement   : dimension_measurement;
END_ENTITY;
(*)

```

**size\_ent**, ölçü boyutunun uygulandığı özelliktir.

**range**, tolerans aralığının isteğe bağlı değerleridir.

**measurement**, boyutun özelliğinin, özelliğin bir kenarından simetri merkezinin öteki tarafındaki kenarına kadar veya bir kenar ve simetri merkezi arasında yarım ölçü olup olmadığını gösterir.

### 1.3. Diğer Varlık Temsilleri

**Parça modeli:** Bu varlık, bir parçasının tam şeklini tanımlayan bilgileri bulundurur.

```

*)
ENTITY part_model

```

```

SUBTYPE OF (design_model);
  nominal_shape : shape_model;
  model_units   : units;
  part_feature  : OPTIONAL LIST [0:#] OF formfeature;
  part_tolerance : OPTIONAL LIST [0:#] OF shape_tolerance;
  equivalentents : OPTIONAL LIST [0:#] OF part_model_structure;
END_ENTITY;
(*)

```

**nominal\_shape**, parçanın anılan şeklidir.

**model\_units**, modeli tanımlayan uzunluk birimleridir.

**part\_feature**, nominal parça şekli ile ilgili parçanın özelliğidir.

**part\_tolerance**, nominal parça şekli ile ilgili toleranslardır.

**equivalentents**, parçanın eşdeğer temsilidir.

**Parça model yapısı:** Parça modeli ile LKS'nin ilişkisini kurar.

```

*)
ENTITY part_model_structure
  model_element : part_model
  location      : axis2_placement;
END_ENTITY;
(*)

```

**model\_element**, bir parça modelidir.

**location**, parça modelinin yerleşimidir.

**Şekil modeli:** Bir parçanın nominal şeklinin temsilidir.

```

*)
ENTITY shape_model
  SUPERTYPE OF (solid_model XOR
                surface_modelXOR
                wire_frame_modelXOR
                geometric_set);
END_ENTITY;
(*)

```

**Katı model:** Ürünün nominal şeklinin komple temsilidir. Katı model temsilinin farklı birkaç tipi vardır.

```

*)
ENTITY solid_model
  SUPERTYPE OF (boolean_expression XOR
                CSG_primitive XOR
                faceted_brep XOR
                half_space XOR
                manifold_brep XOR
                solid_instance XOR
                swept_area_solid);
  SUBTYPE OF (shape_model);
END_ENTITY;
(*

```

**Manifold solid B\_rep**, aşağıdaki şekilde tanımlanır.

```

*)
ENTITY manifold_solid_brep
  SUBTYPE OF (solid_model);
  outhr      : shell_or_logical;
  voids      : SET [0:#] OF shell_or_logical_structure;
END_ENTITY;
(*

```

**outer**, katı modelin dış sınırını tanımlayan kapalı bir kabuktur.

**voids**, katı model boşluklarını tanımlayan kapalı kabuklar kümesidir. Küme boş da olabilir.

**Kapalı bir kabuk:** Yüzeyler koleksiyonu ile temsil edilen bir kabuktur.

```

*)
ENTITY closed_shell
  SUBTYPE OF (shell);
  cshell_boundary : SET [1:#] OF face_or_logical;
END_ENTITY;
(*

```

**cshell\_boundary**, kapalı kabuktan meydana gelen yüzey mantık yapısı veya yüzey kümesidir.

#### 1.4. Fiziksel Dosyada Komple Bir Parçanın Temsil Edilmesi

Bu bölümde açıklanan varlık tanımlamalarını kullanarak Şekil B.1'de verilen örnek parçanın, fiziksel dosyada STEP standarda göre temsil edilmesi ve Prolog formatı aşağıda verilmiştir. Bu çalışmadaki program, Türkçe ifadelerden oluştuğu için, STEP standardındaki varlıkların isimleri de Türkçe karşılıkları kullanılarak hazırlanmıştır.

```

/* STEP standartı */

PARÇA_ADI(PARCA2) .
STEP .
  HEADER(
    DOSYA_TANIMLAMA('PARCA1','10,01,1995','SÜLEYMAN YALDIZ',
                    'SELCUK UNİVERSİTESİ',01,'_') ,
    DOSYA_AÇIKLAMASI('Parça2-iki kademe ve bir delik'),
    UYG_SEV('1. SEVİYE'),
    SINIFI(_),
  ENDSEC;

DATA
  #5 = NOKTA(0,0,0);      /* nokta tanımlama */
  #10 = NOKTA(120,0,0);
  #15 = NOKTA(120,0,60);
  #20 = NOKTA(0,0,60);
  #25 = NOKTA(120,80,0);
  #30 = NOKTA(120,80,60);
  #35 = NOKTA(0,80,0);
  #40 = NOKTA(0,80,60);

  #45 = KÖSE(#5);        /* köşe tanımlama */
  #50 = KÖSE(#10);
  #55 = KÖSE(#15);
  #60 = KÖSE(#20);
  #65 = KÖSE(#25);
  #70 = KÖSE(#30);
  #75 = KÖSE(#35);
  #80 = KÖSE(#40);

```

```

#85 = YÖN(1,0,0);      /* + X yönlü vektör */
#86 = YÖN(-1,0,0);    /* - X yönlü vektör */
#87 = YÖN(0,1,0);    /* + Y yönlü vektör */
#88 = YÖN(0,-1,0);   /* - Y yönlü vektör */
#89 = YÖN(0,0,1);    /* + Z yönlü vektör */
#90 = YÖN(0,0,-1);   /* - Z yönlü vektör */

```

```

#95 = ÇİZGİ(#5,#85);  /* çizgi tanımlama */
#100 = ÇİZGİ(#10,#89);
#105 = ÇİZGİ(#15,#86);
#110 = ÇİZGİ(#20,#90);
#115 = ÇİZGİ(#10,#87);
#120 = ÇİZGİ(#25,#89);
#125 = ÇİZGİ(#30,#88);
#130 = ÇİZGİ(#25,#86);
#135 = ÇİZGİ(#35,#89);
#140 = ÇİZGİ(#40,#85);
#145 = ÇİZGİ(#35,#88);
#150 = ÇİZGİ(#40,#88);

```

```

#155 = KENAR(#45,#50,#95); /* kenar tanımlama */
#160 = KENAR(#50,#55,#100);
#165 = KENAR(#55,#60,#105);
#170 = KENAR(#60,#45,#110);
#175 = KENAR(#50,#65,#115);
#180 = KENAR(#65,#70,#120);
#185 = KENAR(#70,#55,#125);
#190 = KENAR(#65,#75,#130);
#195 = KENAR(#75,#80,#135);
#200 = KENAR(#80,#70,#140);
#205 = KENAR(#75,#45,#145);
#210 = KENAR(#60,#80,#150);

```

```

#225 = KENAR_MANTIK_YAPISI(#160,.F.);
#230 = KENAR_MANTIK_YAPISI(#180,.F.);
#235 = KENAR_MANTIK_YAPISI(#170,.F.);
#240 = KENAR_MANTIK_YAPISI(#195,.F.);
#245 = KENAR_MANTIK_YAPISI(#165,.F.);
#250 = KENAR_MANTIK_YAPISI(#185,.F.);
#255 = KENAR_MANTIK_YAPISI(#200,.F.);
#260 = KENAR_MANTIK_YAPISI(#210,.F.);
#265 = KENAR_MANTIK_YAPISI(#190,.F.);
#270 = KENAR_MANTIK_YAPISI(#175,.F.);
#275 = KENAR_MANTIK_YAPISI(#155,.F.);
#280 = KENAR_MANTIK_YAPISI(#205,.F.);

```

```

#285 = KENAR_ÇEVİRİMİ(#155,#160,#165,#170);
#290 = KENAR_ÇEVİRİMİ(#175,#180,#185,#225);
#295 = KENAR_ÇEVİRİMİ(#190,#195,#200,#230);
#300 = KENAR_ÇEVİRİMİ(#205,#235,#210,#240);
#310 = KENAR_ÇEVİRİMİ(#245,#250,#255,#260);
#315 = KENAR_ÇEVİRİMİ(#265,#270,#275,#280);

#320 = DÜZLEM(EKSEN2_YERLEŞTİRME(#5,(0,-1,0), ));
#325 = DÜZLEM(EKSEN2_YERLEŞTİRME(#10,(1,0,0), ));
#330 = DÜZLEM(EKSEN2_YERLEŞTİRME(#25,(0,1,0), ));
#335 = DÜZLEM(EKSEN2_YERLEŞTİRME(#35,(-1,0,0), ));
#340 = DÜZLEM(EKSEN2_YERLEŞTİRME(#20,(0,0,1), ));
#345 = DÜZLEM(EKSEN2_YERLEŞTİRME(#35,(0,0,-1), ));

#350 = YÜZEY(#285,(#285),#320);      /* yüzey tanımlama */
#355 = YÜZEY(#290,(#290),#325);
#360 = YÜZEY(#295,(#295),#330);
#365 = YÜZEY(#300,(#300),#335);

#370=EKSEN2_YERLEŞTİRME((60,40,0),(0,0,1),(1,0,0));
#375= ÖZELLİK_SÜPÜRME
SCOPE
#382= SİMETRİEKSENLİ_ÖZELLİK_SÜPÜRME
SCOPE

#384=SABİT_ÇAPLI_SİMETRİEKSENLİ_ÖZELLİK_SÜPÜRME(20);
ENDSCOPE
(60,);
ENDSCOPE
(#370);

#385=EKSEN2_YERLEŞTİRME((30,0,40),(0,1,0),(-1,0,0));
#390= DOĞRUSAL_ÖZELLİK_SÜPÜRME_YOLU(80);
#395=L_ÖZELLİK_SÜPÜRME_PROFİLİ(0,30,0, . L_POZİTİF., , , );
#400= ÖZELLİK_SÜPÜRME
SCOPE
#405= BOYDANBOYA_ÖZELLİK_SÜPÜRME(#390,#395,());
ENDSCOPE
(#385);

#410=EKSEN2_YERLEŞTİRME((90,0,40),(0,1,0),(0,0,1));
#415= DOĞRUSAL_ÖZELLİK_SÜPÜRME_YOLU(80);
#420=L_ÖZELLİK_SÜPÜRME_PROFİLİ(0,20,0, . L_POZİTİF., , , );
#425= ÖZELLİK_SÜPÜRME
SCOPE
#430= BOYDANBOYA_ÖZELLİK_SÜPÜRME(#415,#420,());

```

ENDSCOPE  
(#410);

#435 = YÜZEY  
SCOPE

#440 = ŞEKİL\_ÖZELLİĞİ('DELİK',(#375), , );  
#445 = ŞEKİL\_ÖZELLİĞİ('KADEME',(#400), , );  
#450 = ŞEKİL\_ÖZELLİĞİ('KADEME',(#425), , );

ENDSCOPE  
(#310,(#310),#340);

#455 = YÜZEY(#315,(#315),#345);

#460 = TOLERANS\_ARALIĞI(0.01,0.01);  
#462 = TOLERANS\_ARALIĞI(0.02,0.02);  
#464 = TOLERANS\_ARALIĞI(0.05,0.05);  
#466 = TOLERANS\_ARALIĞI(0.2,0.2);  
#468 = TOLERANS\_ARALIĞI(0.1,0.1);  
#470 = TOLERANS\_ARALIĞI(0.09,0.09);  
#472 = TOLERANS\_ARALIĞI(0.1,0.1);  
#474 = TOLERANS\_ARALIĞI(0.6,0.6);

#480 = YERLEŞTİRME\_BOYUTU(#472,#355,#365,.T.,.T., );  
#485 = YERLEŞTİRME\_BOYUTU(#468,#360,#350,.T.,.T., );  
#490 = YERLEŞTİRME\_BOYUTU(#468,#455,#435,.T.,.T., );  
#495 = YERLEŞTİRME\_BOYUTU(#466,#440,#365,.T.,.T., );  
#500 = YERLEŞTİRME\_BOYUTU(#466,#440,#350,.T.,.T., );  
#505 = YERLEŞTİRME\_BOYUTU(#464,#450,#455,.T.,.T., );  
#510 = YERLEŞTİRME\_BOYUTU(#464,#450,#365,.T.,.T., );  
#515 = YERLEŞTİRME\_BOYUTU(#464,#445,#365,.T.,.T., );  
#520 = YERLEŞTİRME\_BOYUTU(#466,#445,#455,.T.,.T., );

#525 = ÖLÇÜ\_BOYUTU(#440,#468,.TAM.);  
#530 = DAİRESELLİK(#440),0.05);  
#535 = DOĞRUSALLIK(#440),0.6, ,.REGARDLESS.,.T.,60);  
#540 = KAPALI\_KABUK(#350,#355,#360,#365,#435,#455);  
#545 = ÇEŞİTLİ\_KATI\_TEMSİLİ(#540,());  
#550 = PARÇA\_MODELİ(#545,'mm',(#440,#445,#450),(),());  
#555 = MALZEME\_ÖZELLİĞİ('1040', );  
#560 = EKSEN2\_YERLEŞTİRME((0,0,0),(0,0,1),(1,0,0));  
#565 = PARÇA\_MODEL\_YAPISI(#550,#560);

ENDSEC;  
ENDSTEP;



**/\* Prolog formatı \*/**

varlik(5 , nokta(0,0,0)).  
 varlik(10 , nokta(120,0,0)).  
 varlik(15 , nokta(120,0,60)).  
 varlik(20 , nokta(0,0,60)).  
 varlik(25 , nokta(120,80,0)).  
 varlik(30 , nokta(120,80,60)).  
 varlik(35 , nokta(0,80,0)).  
 varlik(40 , nokta(0,80,60)).

varlik(45 , kose(varlikn(5))).  
 varlik(50 , kose(varlikn(10))).  
 varlik(55 , kose(varlikn(15))).  
 varlik(60 , kose(varlikn(20))).  
 varlik(65 , kose(varlikn(25))).  
 varlik(70 , kose(varlikn(30))).  
 varlik(75 , kose(varlikn(35))).  
 varlik(80 , kose(varlikn(40))).

varlik(85 , yon(1,0,0)).  
 varlik(86 , yon(-1,0,0)).  
 varlik(87 , yon(0,1,0)).  
 varlik(88 , yon(0,-1,0)).  
 varlik(89 , yon(0,0,1)).  
 varlik(90 , yon(0,0,-1)).

varlik(95 , cizgi(varlikn(5),varlikn(85))).  
 varlik(100 , cizgi(varlikn(10),varlikn(89))).  
 varlik(105 , cizgi(varlikn(15),varlikn(86))).  
 varlik(110 , cizgi(varlikn(20),varlikn(90))).  
 varlik(115 , cizgi(varlikn(10),varlikn(87))).  
 varlik(120 , cizgi(varlikn(25),varlikn(89))).  
 varlik(125 , cizgi(varlikn(30),varlikn(88))).  
 varlik(130 , cizgi(varlikn(25),varlikn(86))).  
 varlik(135 , cizgi(varlikn(35),varlikn(89))).  
 varlik(140 , cizgi(varlikn(40),varlikn(85))).  
 varlik(145 , cizgi(varlikn(35),varlikn(88))).  
 varlik(150 , cizgi(varlikn(40),varlikn(88))).

varlik(155 , kenar(varlikn(45),varlikn(50),varlikn(95))).  
 varlik(160 , kenar(varlikn(50),varlikn(55),varlikn(100))).  
 varlik(165 , kenar(varlikn(55),varlikn(60),varlikn(105))).  
 varlik(170 , kenar(varlikn(60),varlikn(45),varlikn(110))).  
 varlik(175 , kenar(varlikn(50),varlikn(65),varlikn(115))).  
 varlik(180 , kenar(varlikn(65),varlikn(70),varlikn(120))).

varlik(185 , kenar(varlikn(70),varlikn(55),varlikn(125))).  
 varlik(190 , kenar(varlikn(65),varlikn(75),varlikn(130))).  
 varlik(195 , kenar(varlikn(75),varlikn(80),varlikn(135))).  
 varlik(200 , kenar(varlikn(80),varlikn(70),varlikn(140))).  
 varlik(205 , kenar(varlikn(75),varlikn(45),varlikn(145))).  
 varlik(210 , kenar(varlikn(60),varlikn(80),varlikn(150))).

varlik(225 , kenar\_mantik\_yapisi(varlikn(160),'f')).  
 varlik(230 , kenar\_mantik\_yapisi(varlikn(180),'f')).  
 varlik(235 , kenar\_mantik\_yapisi(varlikn(170),'f')).  
 varlik(240 , kenar\_mantik\_yapisi(varlikn(195),'f')).  
 varlik(245 , kenar\_mantik\_yapisi(varlikn(165),'f')).  
 varlik(250 , kenar\_mantik\_yapisi(varlikn(185),'f')).  
 varlik(255 , kenar\_mantik\_yapisi(varlikn(200),'f')).  
 varlik(260 , kenar\_mantik\_yapisi(varlikn(210),'f')).  
 varlik(265 , kenar\_mantik\_yapisi(varlikn(190),'f')).  
 varlik(270 , kenar\_mantik\_yapisi(varlikn(175),'f')).  
 varlik(275 , kenar\_mantik\_yapisi(varlikn(155),'f')).  
 varlik(280 , kenar\_mantik\_yapisi(varlikn(205),'f')).

varlik(285,kenar\_cevrime([varlikn(155),varlikn(160),varlikn(165),varlikn(170)])).  
 varlik(290,kenar\_cevrime([varlikn(175),varlikn(180),varlikn(185),varlikn(225)])).  
 varlik(295,kenar\_cevrime([varlikn(190),varlikn(195),varlikn(200),varlikn(230)])).  
 varlik(300,kenar\_cevrime([varlikn(205),varlikn(235),varlikn(210),varlikn(240)])).  
 varlik(310,kenar\_cevrime([varlikn(245),varlikn(250),varlikn(255),varlikn(260)])).  
 varlik(315,kenar\_cevrime([varlikn(265),varlikn(270),varlikn(275),varlikn(280)])).

varlik(320 , duzlem(eksen\_yerlestirme(varlikn(5),0,-1,0,0)).  
 varlik(325 , duzlem(eksen\_yerlestirme(varlikn(10),1,0,0,0)).  
 varlik(330 , duzlem(eksen\_yerlestirme(varlikn(25),0,1,0,0)).  
 varlik(335 , duzlem(eksen\_yerlestirme(varlikn(35),-1,0,0,0)).  
 varlik(340 , duzlem(eksen\_yerlestirme(varlikn(20),0,0,1,0)).  
 varlik(345 , duzlem(eksen\_yerlestirme(varlikn(35),0,0,-1,0)).

varlik(350 , yuzey(varlikn(285),[varlikn(285)],varlikn(320)).  
 varlik(355 , yuzey(varlikn(290),[varlikn(290)],varlikn(325)).  
 varlik(360 , yuzey(varlikn(295),[varlikn(295)],varlikn(330)).  
 varlik(365 , yuzey(varlikn(300),[varlikn(300)],varlikn(335)).

varlik(370,eksen2\_yerlestirme(koordinat(60,40,0),vektor(0,0,1),vektor(1,0,0)).  
 varlik(375,ozellik\_supurme(  
 [  
 varlik3(382,simetriksenli\_ozellik\_supurme(  
 [  
 varlik4(384,sabit\_capli\_simetriksenli\_ozellik\_supurme(16))  
 ],  
 ],

```
60,varlikn(0)))
],
varlikn(370))).
```

```
varlik(385,eksen2_yerlestirme(koordinat(30,0,40),vektor(0,1,0),vektor(-1,0,0))).
varlik(390,dogrusal_ozellik_supurme_yolu(80)).
varlik(395,l_ozellik_supurme_profili(0,30,0,l_pozitif)).
varlik(400,ozellik_supurme1(
[
varlik5(405,boydanboya_ozellik_supurme(varlikn(390),varlikn(395)))
],
varlikn(385))).
```

```
varlik(410,eksen2_yerlestirme(koordinat(90,0,40),vektor(0,1,0),vektor(0,0,1))).
varlik(415,dogrusal_ozellik_supurme_yolu(80)).
varlik(420,l_ozellik_supurme_profili(0,20,0,l_pozitif)).
varlik(425,ozellik_supurme1(
[
varlik5(430,boydanboya_ozellik_supurme(varlikn(415),varlikn(420)))
],
varlikn(410))).
```

```
varlik(435,yuzey1(
[
varlik2(440,sekil_ozelligi("delik",[varlikn(375)])),
varlik2(445,sekil_ozelligi("kademe",[varlikn(400)])),
varlik2(450,sekil_ozelligi("kademe",[varlikn(425)]))
],
varlikn(310),[varlikn(310)],varlikn(340))).
```

```
varlik(455,yuzey(varlikn(315),[varlikn(315)],varlikn(345))).
```

```
varlik(460,tolerans_araligi(0.01,0.01)).
varlik(462,tolerans_araligi(0.02,0.02)).
varlik(464,tolerans_araligi(0.05,0.05)).
varlik(466,tolerans_araligi(0.2,0.2)).
varlik(468,tolerans_araligi(0.1,0.1)).
varlik(470,tolerans_araligi(0.09,0.09)).
varlik(472,tolerans_araligi(0.1,0.1)).
varlik(474,tolerans_araligi(0.6,0.6)).
```

```
varlik(480,yerlestirme_boyutu(varlikn(472),varlikn(355),varlikn(365),'t','t','a')).
varlik(485,yerlestirme_boyutu(varlikn(468),varlikn(360),varlikn(350),'t','t','a')).
varlik(490,yerlestirme_boyutu(varlikn(468),varlikn(455),varlikn(435),'t','t','a')).
varlik(495,yerlestirme_boyutu(varlikn(466),varlikn(440),varlikn(365),'t','t','a')).
varlik(500,yerlestirme_boyutu(varlikn(466),varlikn(440),varlikn(350),'t','t','a')).
```

varlik(505 , yerlestirme\_boyutu(varlikn(464),varlikn(450),varlikn(455),'t','t','a')).  
 varlik(510 , yerlestirme\_boyutu(varlikn(464),varlikn(450),varlikn(365),'t','t','a')).  
 varlik(515 , yerlestirme\_boyutu(varlikn(464),varlikn(445),varlikn(365),'t','t','a')).  
 varlik(520 , yerlestirme\_boyutu(varlikn(466),varlikn(445),varlikn(455),'t','t','a')).

varlik(525 , olcu\_boyutu(varlikn(440),varlikn(468),"tam")).  
 varlik(530 , dairesellik([ varlikn(440)] ,0.05)).  
 varlik(535 , dogrusallik([ varlikn(440)],0.6, regardless,'t',50)).

varlik(540,kapali\_kabuk([varlikn(350),varlikn(355),varlikn(360),varlikn(365),  
 varlikn(435),varlikn(455) ])).  
 varlik(545 , cesitli\_kati\_temsili(varlikn(540),[])).  
 varlik(550,parca\_modeli(varlikn(545),"mm",  
 [varlikn(440),varlikn(445),varlikn(450)],[])).  
 varlik(555 , malzeme\_ozelligi("1040",2)).  
 varlik(560 , eksen2\_yerlestirme(koordinat(0,0,0),vektor(0,0,1),vektor(1,0,0))).  
 varlik(565 ,parca\_model\_yapisi(varlikn(550),varlikn(560))).



## EK-C PARÇALARIN MODELLENMESİNDE KULLANILAN FORMÜLLER

### 1. Bir Doğru (Kenar) Üzerindeki Noktanın Belirlenmesi

Doğrular, üç boyutlu uzayda aşağıdaki üç doğru denklemiyle ifade edilir.

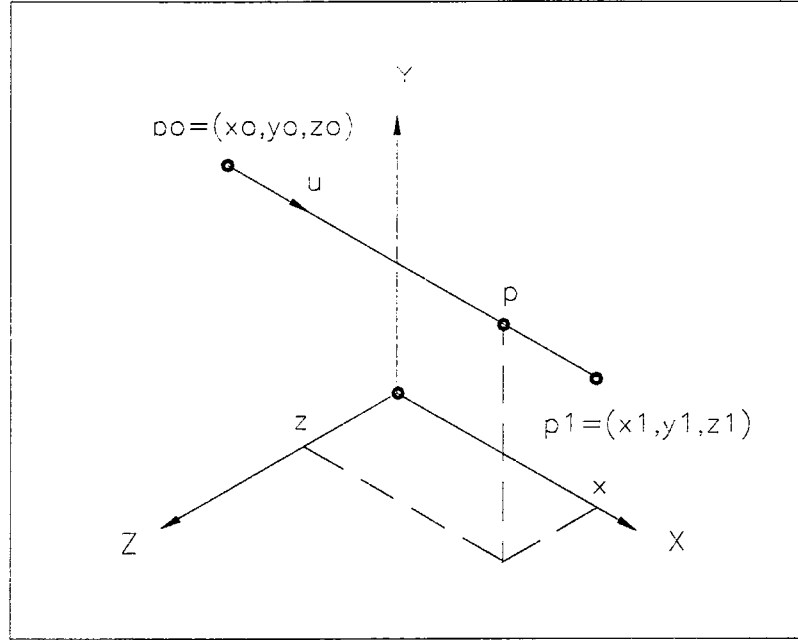
$$\begin{aligned} x &= (x_1 - x_0) u + x_0 \\ y &= (y_1 - y_0) u + y_0 \\ z &= (z_1 - z_0) u + z_0 \end{aligned} \quad u \in [0,1] \quad (C. 1)$$

Bu ifadede  $x, y, z$  bağlı değişkenler olup bir noktanın koordinatlarıdır.

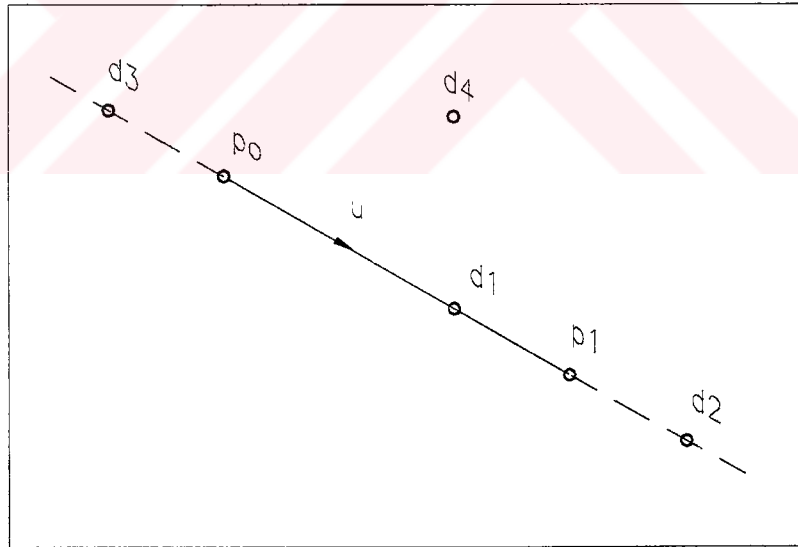
$u$  parametrik değişkeni, **birim boyu** gösterir. Bu denklemler,  $u$  0'dan 1'e kadar değerler alırsa, sınırlı bir doğru parçası meydana getirir. Bu denklemleri uygulamak suretiyle  $P_0$  ve  $P_1$  uç noktaları arasında çeşitli noktalar oluşturulabilir (Şekil C.1).

Test noktası olarak verilen bir  $d$  noktası doğrunun ya üzerinde, ya da dışındadır (Şekil C.2). Eğer test noktası doğrunun üzerinde ise, bu nokta sınır noktaları arasında bir  $d_1$  noktası; doğrunun ileriye uzantısı üzerinde bir  $d_2$  noktası veya geriye uzantısı üzerinde bir  $d_3$  noktası olabilir. Aksi halde doğrunun uzantısında olmayan  $d_4$  gibi bir noktadır. Verilen C.1 denklemine göre  $u$  aşağıdaki şekilde elde edilir.

$$\begin{aligned} U_x &= \frac{X - X_0}{X_1 - X_0} \\ U_y &= \frac{Y - Y_0}{Y_1 - Y_0} \\ U_z &= \frac{Z - Z_0}{Z_1 - Z_0} \end{aligned} \quad (C. 2)$$



Şekil C.1. Üç boyutlu uzayda doğru parçası



Şekil C.2. Doğru ve nokta ilişkisi

Eğer  $u_x = u_y = u_z$  ise, koordinatları  $\mathbf{d}=(\mathbf{x}, \mathbf{y}, \mathbf{z})$  şeklinde verilen her hangi bir  $\mathbf{d}$  noktası, doğru üzerindedir. Aksi halde doğru üzerinde değildir. Bu yüzden aşağıdaki (C.3, C.4, C.5) denklemleri doğru üzerindeki noktayı sağlamalıdır.

$$u_x = u_y$$

$$\frac{x - x_0}{x_1 - x_0} = \frac{y - y_0}{y_1 - y_0}$$

$$((x - x_0)(y_1 - y_0) - (x_1 - x_0)(y - y_0)) = 0 \quad (\text{C. 3})$$

$$u_y = u_z$$

$$\frac{y - y_0}{y_1 - y_0} = \frac{z - z_0}{z_1 - z_0}$$

$$((y - y_0)(z_1 - z_0) - (y_1 - y_0)(z - z_0)) = 0 \quad (\text{C. 4})$$

$$u_x = u_z$$

$$\frac{x - x_0}{x_1 - x_0} = \frac{z - z_0}{z_1 - z_0}$$

$$(x - x_0)(z_1 - z_0) - (x_1 - x_0)(z - z_0) = 0 \quad (\text{C. 5})$$

Diğer bir ifadeyle, eğer doğru (bir kenar gibi) sınırlandırılmış ve bir **noktanın** bir **kenar** üzerinde olup olmadığını kontrol etmek gerekirse, aşağıdaki karşılaştırmalar sağlanmalıdır.

$$X_0 \leq X < X_1$$

$$Y_0 \leq Y < Y_1$$

$$Z_0 \leq Z < Z_1$$

(C. 6)

## 2. İki Doğrunun Arakesit Noktasının Bulunması

Arakesit oluşturan iki doğru bir **arakesit noktasına** sahiptir. Şekil C.3'te görülen  $L_1$  ve  $L_2$  doğruları şöyle bir arakesit noktasına sahip olsunlar:

$$\begin{aligned}x &= x_{L_1} = x_{L_2} \\y &= y_{L_1} = y_{L_2} \\z &= z_{L_1} = z_{L_2}\end{aligned} \quad (\text{C. 7})$$

Denklem C. 1' den aşağıdakiler yazılır.

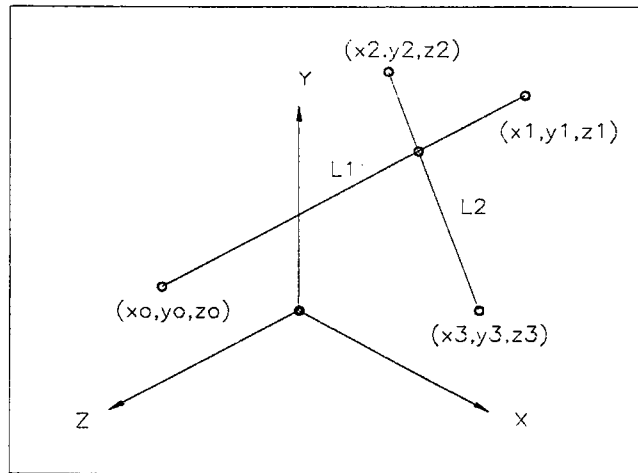
$$((x_1 - x_0)u_1 + x_0 = (x_3 - x_2)u_2 + x_2 \quad (\text{C. 8})$$

$$((y_1 - x_0)u_1 + y_0 = (y_3 - y_2)u_2 + y_2 \quad (\text{C. 9})$$

$$((z_1 - z_0)u_1 + z_0 = (z_3 - z_2)u_2 + z_2 \quad (\text{C. 10})$$

Böylece  $u_1$  ve  $u_2$  gibi iki bilinmeyenli üç denklem elde edilir. Bunlardan birisi çekilerek diğerlerinden birinde yerine yazılırsa:

$$u_1 = \frac{(x_3 - x_2)u_2 + x_2 - x_0}{(x_1 - x_0)} \quad (u_1 \text{'in değeri C. 9'da yerine yazılır})$$



Şekil C.3. İki doğrunun arakesiti



$$(y_1 - y_0) \left( \frac{(x_3 - x_2)u_2 + x_2 - x_0}{(x_1 - x_0)} \right) + y_0 = (y_3 - y_2)$$

$$u_2 + y_2$$

$$\frac{(y_1 - y_0)(x_3 - x_2)u_2 + x_2(y_1 - y_0) - x_0(y_1 - y_0)}{(x_1 - x_0)} = (y_3 - y_0)$$

$$u_2 + y_2 - y_0$$

$$(y_1 - y_0)(x_3 - x_2)u_2 + x_2(y_1 - y_0) - x_0(y_1 - y_0) = (x_1 - x_0)(y_3 - y_2)u_2 + y_2(x_1 - x_0) - y_0(x_1 - x_0)$$

$$(y_1 - y_0)(x_3 - x_2)u_2 - (x_1 - x_0)(y_3 - y_2)u_2 = (x_1 - x_0)(y_2 - y_0) - x_2(y_1 - y_0) + x_0(y_1 - y_0)$$

$$u_2((y_1 - y_0)(x_3 - x_2) - (x_1 - x_0)(y_3 - y_2)) = (y_2 - y_0)(x_1 - x_0) + (x_0 - x_2)(y_1 - y_0)$$

$$u_2 = \frac{(y_2 - y_0)(x_1 - x_0) + (x_0 - x_2)(y_1 - y_0)}{(y_1 - y_0)(x_3 - x_2) - (x_1 - x_0)(y_3 - y_2)} \quad \text{elde edilir.}$$

$u_1$  ve  $u_2$ ' yi yerine koymayla C.10 denklemi sağlanır. Eğer denklem sağlanırsa bu iki doğru kesişir.

Arakesit noktasının koordinatlarını hesaplamak gerekirse, aşağıdaki eşitlikler kullanılır.

$$x = (x_1 - x_0)u_1 + x_0$$

$$y = (y_1 - y_0)u_1 + y_0$$

$$z = (z_1 - z_0)u_1 + z_0$$

veya

$$x = (x_3 - x_2)u_2 + x_2$$

$$y = (y_3 - y_2)u_2 + y_2$$

$$z = (z_3 - z_2)u_2 + z_2$$

### 3. Yüzeyler Arasındaki Açıların Hesaplanması

İki yüzey arasındaki açının hesaplanması, yüzeylerin yön vektörleri arasındaki açıya bağlıdır. STEP dosyasında her yüzey, birim vektör olan bir **yön vektörü** ile yönlendirilmiştir (Şekil C.4). Her yüzeyin birim vektörü ( $U$ ), üç elemana sahiptir.  $U=[U_x, U_y, U_z]$ . İki birim vektör ( $U_1$  ve  $U_2$ ) arasındaki  $\theta$  açısı aşağıdaki denklemle sağlanır.

$$u_1 \cdot u_2 = |u_1| |u_2| \cos \theta \quad (C.11)$$

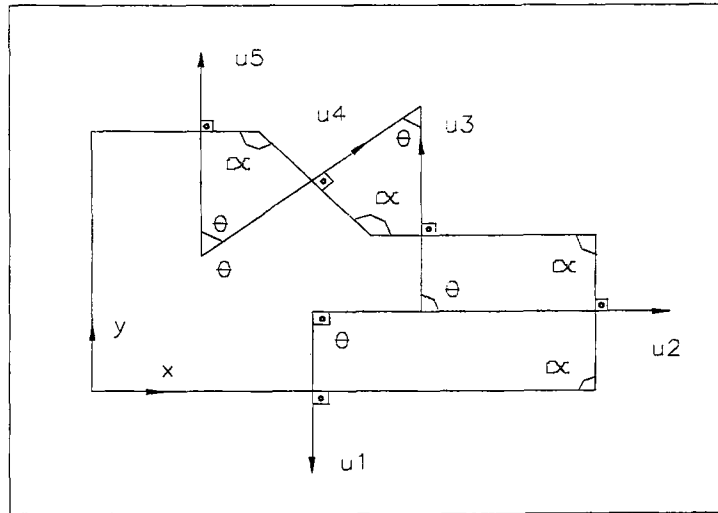
Burada  $|u_1|$  ve  $|u_2|$   $u_1$  ve  $u_2$  birim vektörlerinin büyüklüğüdür. Denklem C.11'i çözerek  $\theta$  açısı elde edilir.

$$\theta = \cos^{-1} \frac{u_1 \cdot u_2}{|u_1| |u_2|} \quad (C.12)$$

İki yüzey arasındaki  $\alpha$  açısı aşağıdaki eşitlik ile hesaplanır.

$$\alpha + \theta = 180^\circ \quad (C.13)$$

$$\alpha = 180 - \theta \quad (C.14)$$



Şekil C.4. Yüzeyler arasındaki açıların hesaplanması

Örnek olarak  $u_4$  ve  $u_5$  birim vektörleri aşağıdaki şekilde verilmiş olsun.

$$u_4 = [0.866 \ 0.5 \ 0]$$

$$u_5 = [0 \ 1 \ 0]$$

İki vektörün skaler çarpımıyla:

$$u_4 \cdot u_5 = [0.866 \ 0.5 \ 0] \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$u_4 \cdot u_5 = 0.5 \quad \text{elde edilir.}$$

Elde edilen sonucun C.12'ye uygulanmasıyla  $\theta$  açısı bulunur.

$$\cos\theta = \frac{U_1 \cdot U_2}{|U_1| |U_2|} = \frac{0.5}{\sqrt{0.866^2 + 0.5^2 + 0^2} \cdot \sqrt{0^2 + 1^2 + 0^2}}$$

$$\cos\theta = \frac{0.5}{0.9999} = 0.5 \quad \theta = 60^\circ$$

Bulunan değeri C.14'te yerine yazarak iki yüzey arasındaki  $\alpha$  açısı elde edilir.

$$\alpha = 180 - 60$$

$$\alpha = 120^\circ$$

#### 4. Bir Doğrunun Vektörel Denklemi

$\mathbf{d}$  noktasından geçen ve  $\mathbf{e}$  vektörüne paralel olan bir doğrunun vektörel denklemi aşağıdaki denklem ile tanımlanır.

$$\mathbf{p} = \mathbf{d} + t\mathbf{e} \quad (\text{C.15})$$

Burada :

$t$ ,  $e$  ile çarpım halindeki skalar bir değişkendir.  $t$ , ard arda sayısal değerler aldığıında, doğrusal bir çizgiyi oluşturur. Alınan son değerler  $p$  ile verilen noktanın koordinatlarına karşılık gelir.  $d$  ve  $e$  sabitken,  $t$ 'nin her hangi bir reel değeri, aynı doğrusal çizgi üzerinde bir nokta meydana getirir (Şekil C.5).

C.15 denklemi açılmış vektör formunda yazılarak aşağıdaki eşitlik elde edilir.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + t \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}$$

Eşitlik cebrik formda yazılırsa:

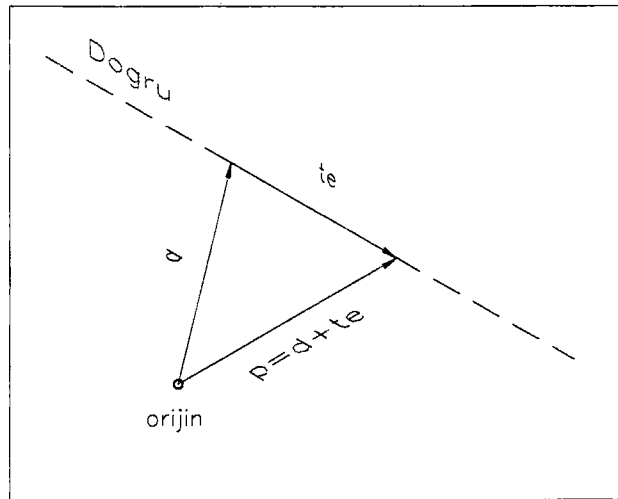
$$x = x_0 + te_x$$

$$y = y_0 + te_y$$

$$z = z_0 + te_z \quad \text{elde edilir.}$$

(C.17)

Burada  $t$ , bağımsız değişken;  $x$ ,  $y$  ve  $z$  bağılı değişken,  $x_0$ ,  $y_0$ ,  $z_0$ ,  $e_x$ ,  $e_y$  ve  $e_z$  sabitlerdir.



Şekil C.5. Doğrunun vektörel denklemi

## 5. Düzlemin Vektörel Denklemi

$\mathbf{a}$ 'dan geçen,  $\mathbf{b}$  ve  $\mathbf{c}$  gibi iki bağımsız vektöre paralel bir düzlemin vektörel denklemi, şu şekilde tanımlanır.

$$\mathbf{p} = \mathbf{a} + u\mathbf{b} + v\mathbf{c} \quad (\text{C. 18})$$

Burada :

$\mathbf{u}$  ve  $\mathbf{v}$ , skalar parametrik değişkenlerdir.  $\mathbf{u}$  ve  $\mathbf{v}$  parametreleri bağımsız olarak değişirken,  $\mathbf{p}$  vektörü düzlem üzerinde noktalar meydana getirir (Şekil C.6). C.18 denklemi açılmış vektör formunda yazılarak aşağıdaki eşitlik elde edilir.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + u \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} + v \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} \quad (\text{C. 19})$$

Bu eşitlik cebrik formda yazılırsa:

$$\begin{aligned} x &= x_0 + ub_x + vc_x \\ y &= y_0 + ub_y + vc_y \\ z &= z_0 + ub_z + vc_z \end{aligned} \quad (\text{C. 20})$$

elde edilir.

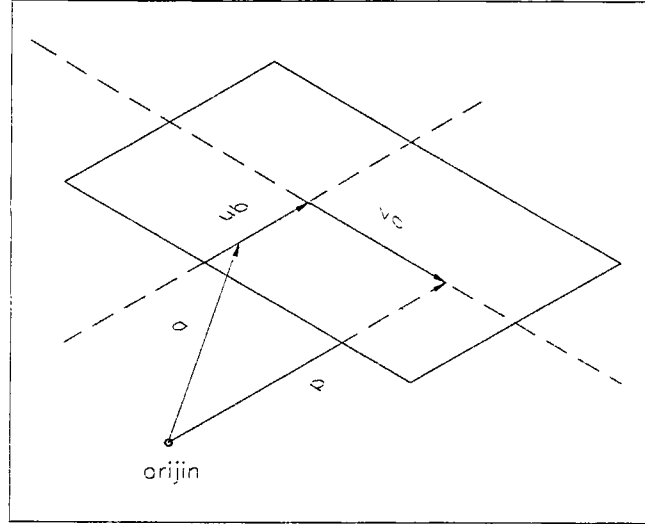
## 6. Bir Düzlem ve Bir Doğru Parçasının Arakesiti

$\mathbf{b}$  ve  $\mathbf{c}$  gibi iki vektörün vektörel çarpımının sonucu, daima bu iki vektöre dik bir vektördür.  $\mathbf{b}$  ve  $\mathbf{c}$  gibi iki vektörün vektörel çarpımlarıyla bulunan vektörün,  $\mathbf{b}$  ve  $\mathbf{c}$  den her hangi birisi ile skalar çarpımının sonucu sifira eşittir.

$$(\mathbf{b} \times \mathbf{c}) \cdot \mathbf{b} = 0 \quad \text{ve} \quad (\mathbf{b} \times \mathbf{c}) \cdot \mathbf{c} = 0$$

Düzlem ile doğru arasındaki arakesit aşağıdaki denklemle belirlenir.

$$\mathbf{a} + u\mathbf{b} + v\mathbf{c} = \mathbf{d} + t\mathbf{e} \quad (\text{C. 21})$$



Şekil C.6. Bir düzlemin vektörel denklemini

Bu eşitlik  $t$ 'yi yalnız bırakarak çözüm yapmak için kullanılır. Eşitliğin her iki tarafı  $(b \times c)$  ile çarpılırsa:

$$(b \times c) \cdot (a + ub + vc) = (b \times c) \cdot (d + te)$$

$(b \times c)$ 'nin sonucu  $b$  ve  $c$  vektörlerine diktir.

$$(b \times c) \cdot a = (b \times c) \cdot d + t(b \times c) \cdot e \quad \text{buradan } t \text{ çekilirse:}$$

$$t = \frac{(b \times c) \cdot a - (b \times c) \cdot d}{(b \times c) \cdot e} \quad \text{elde edilir.} \quad (C.22)$$

## 7. Bir Düzlem Üzerindeki Noktanın Belirlenmesi

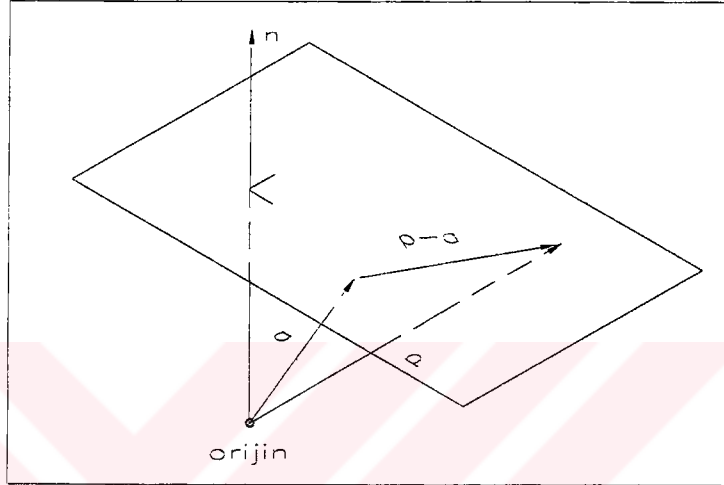
Bir düzlem, düzlemden geçen bir  $a$  noktası ve düzlemin birim normali  $n$  ile tanımlanır (Şekil C.7).  $a$  vektörü, böyle bir noktayı temsil eder. Eğer  $p-a$  vektörü  $n$ 'e dikse veya

$$(p - a) \cdot n = 0 \quad \text{ise} \quad (C.23)$$

o halde  $\mathbf{p}$  vektörü ile temsil edilen bir nokta, düzlem üzerindedir. Vektör bileşen formunda şu şekilde yazılır.

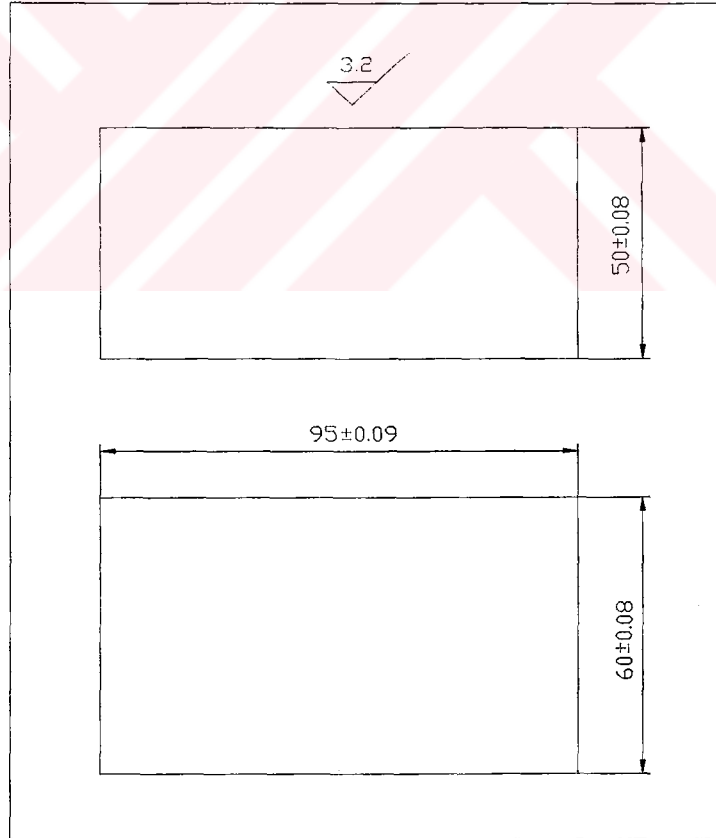
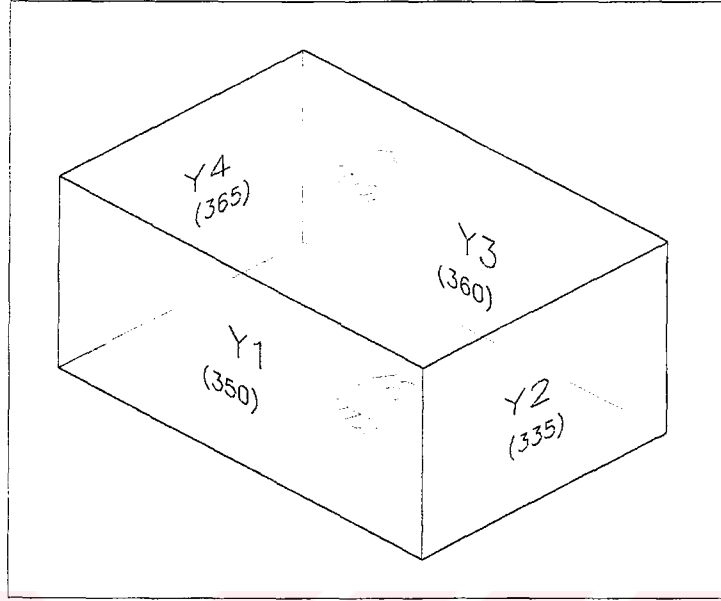
$$(x - x_0)n_x + (y - y_0)n_y + (z - z_0)n_z = 0 \quad (\text{C. 24})$$

Burada  $\mathbf{n}_x$ ,  $\mathbf{n}_y$  ve  $\mathbf{n}_z$ ,  $\mathbf{n}$ 'nin bileşenleridir.



Şekil C.7. Nokta ve yüzey normali ile düzlem tanımlama

**EK-D**  
**ÖRNEK PARÇALAR VE İŞLEM PLAN KARTLARI**



Şekil D.1. Örnek parça-1



## ISLEM PLANI

PARCA ADI : parca1  
 TASLAK OLCUSU : 100\*65\*55  
 MALZEME : C 1040  
 SERTLIK : 22 HRC

OP NO	VAR NO	OZELLIK	OPERASYON	TAKIM	ILERL. SAYISI	DEVIR SAYISI	TAKIM OMRU	TEZGAH
10	375	duzlem_yuzey	yuzey_frezeleme	alin_freze1	196	314	480	freze1
20	370	duzlem_yuzey	yuzey_frezeleme	alin_freze1	196	314	480	freze1
30	350	duzlem_yuzey	yuzey_frezeleme	alin_freze1	196	314	480	freze1
40	365	duzlem_yuzey	yuzey_frezeleme	alin_freze1	196	314	480	freze1
50	360	duzlem_yuzey	yuzey_frezeleme	alin_freze1	196	314	480	freze1
60	355	duzlem_yuzey	yuzey_frezeleme	alin_freze1	196	314	480	freze1

## SECILEN TAKIMLAR

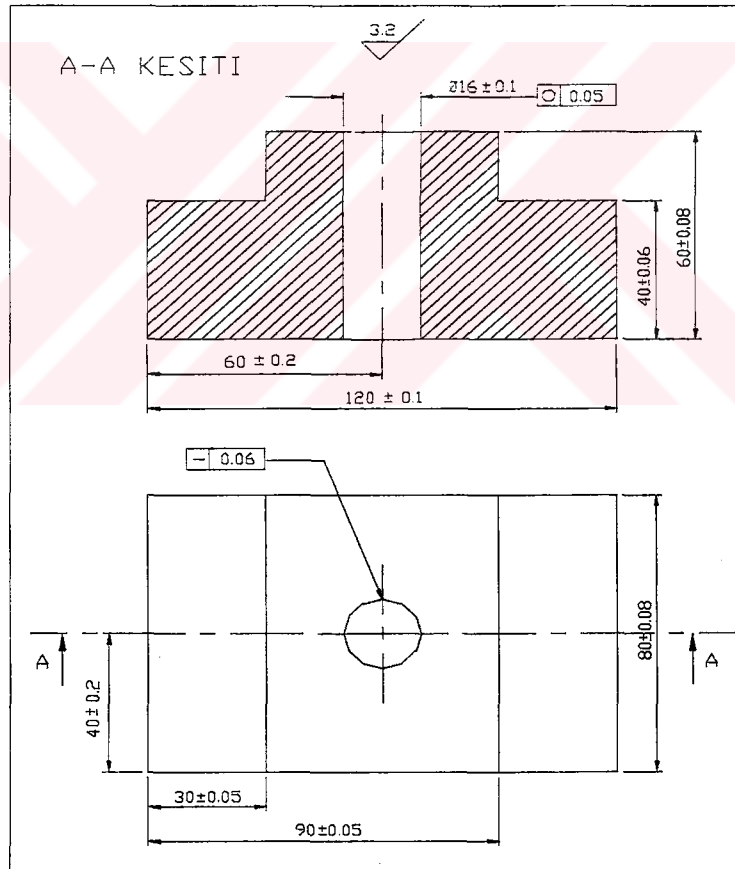
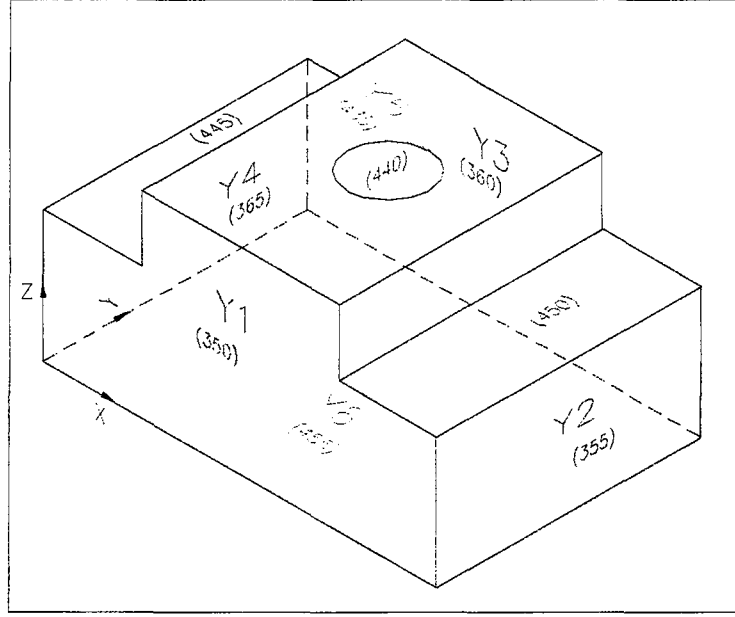
PARCA ADI : parca1

OPNO	TAKIM	EKSENEL ACI	RADYAL ACI	KLAVUZ ACISI	CAP	DIS SAYISI	TAKIM MALZ.	TAKIM KODU
375	alin_freze1	p	p	45	104	8	karbid	p_20
370	alin_freze1	p	p	45	104	8	karbid	p_20
350	alin_freze1	p	p	45	104	8	karbid	p_20
365	alin_freze1	p	p	45	104	8	karbid	p_20
360	alin_freze1	p	p	45	104	8	karbid	p_20
355	alin_freze1	p	p	45	104	8	karbid	p_20

## BAGLAMA

PARCA ADI : parca1

OZELLIK NO	OPERASYON	ISLEME KONUMU	DESTEK YUZEYI NO	Ns	1.YER YUZEYI NO	Ns	2.YER. YUZEYI NO	Ns	1.BAG. YUZEYI NO	Ns	2.BAG YUZEYI NO	Ns
375	yuzey_frezeleme	dusey	370	3	350	2	355	1	360	2	365	1
370	yuzey_frezeleme	dusey	375	3	360	2	355	1	350	2	365	1
350	yuzey_frezeleme	dusey	360	3	375	2	365	1	370	2	355	1
365	yuzey_frezeleme	dusey	355	3	375	2	360	1	370	2	350	1
360	yuzey_frezeleme	dusey	350	3	375	2	355	1	370	2	365	1
355	yuzey_frezeleme	dusey	365	3	375	2	350	1	370	2	360	1



Şekil D.2. Örnek parça-2

---

 ISLEM PLANI
 

---

PARCA ADI : parca2  
 TASLAK OLCUSU : 125\*85\*65  
 MALZEME : C 1040  
 SERTLIK : 22 HRC

OP VAR NO	OZELLIK	OPERASYON	TAKIM	ILERL.	DEVIR SAYISI	TAKIM OMRU	TEZGAH
10 455	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
20 435	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
30 365	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
40 360	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
50 355	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
60 350	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
70 450	kademe	kad_frez.	hss_alin_frz1	1061	1698	480	islm_merk1
80 445	kademe	kad_frez.	hss_alin_frz1	1061	1698	480	islm_merk1
90 440	delik	delme	helisel_matk1	1989	40	480	matkap1

---

 SECILEN TAKIMLAR
 

---

PARCA ADI : parca2

OPNO	TAKIM	EKSENEL ACI	RADYAL ACI	KLAVUZ ACISI	CAP	DIS SAYISI	TAKIM MALZ.	TAKIM KODU
455	alin_freze1	p	p	45	136	10	karbid	p_20
435	alin_freze1	p	p	45	136	10	karbid	p_20
365	alin_freze1	p	p	45	136	10	karbid	p_20
360	alin_freze1	p	p	45	136	10	karbid	p_20
355	alin_freze1	p	p	45	136	10	karbid	p_20
350	alin_freze1	p	p	45	136	10	karbid	p_20
450	hss_alin_freze1	p	p	0	30	6	hss	ts303_11
445	hss_alin_freze1	p	p	0	30	6	hss	ts303_11
440	helisel_matkap1	p	p	0	16	2	hss	ts62_4

---

 BAGLAMA
 

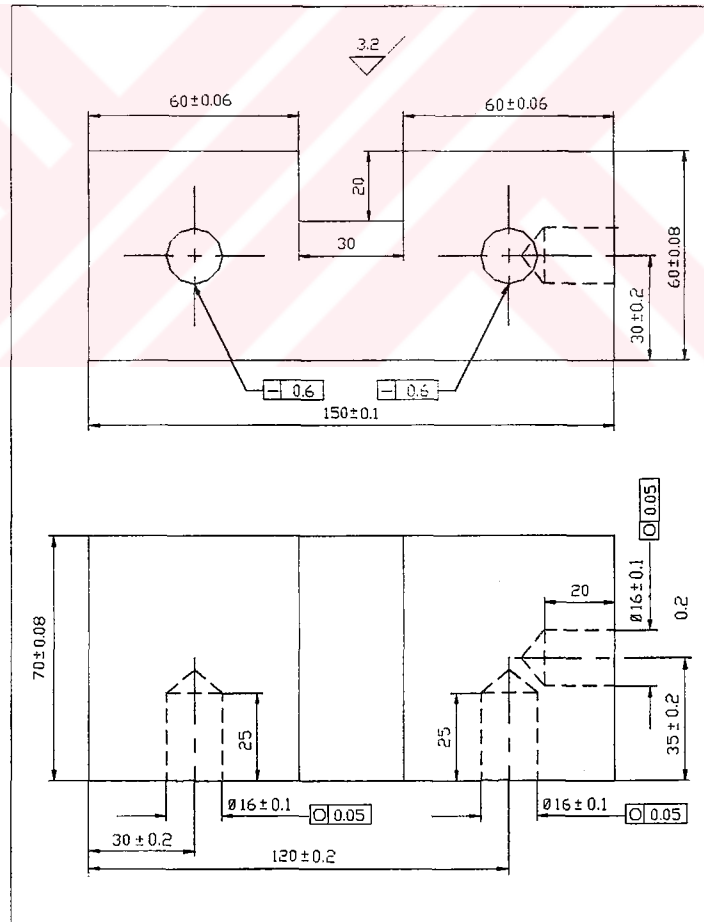
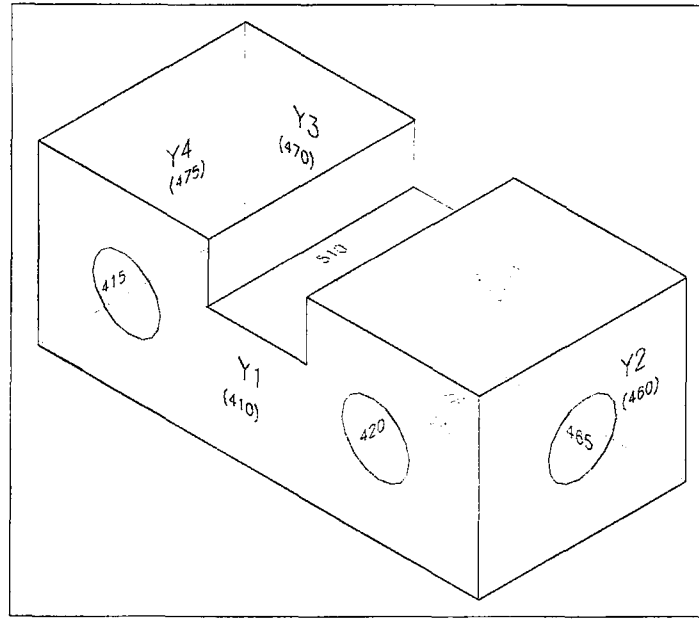
---

 PARCA ADI : parca2
 

---

OZELLIK NO	OPERASYON	ISLEME KONUMU	DESTEK.		1.YER		2.YER.		1.BAG.		2.BAG	
			YUZEYI NO	Ns	YUZEYI NO	Ns	YUZEYI NO	Ns	YUZEYI NO	Ns	YUZEYI NO	Ns
455	yuzey_frezeleme	dusey	435	3	350	2	355	1	360	2	365	1
435	yuzey_frezeleme	dusey	455	3	360	2	355	1	350	2	365	1
365	yuzey_frezeleme	dusey	355	3	455	2	360	1	435	2	350	1
360	yuzey_frezeleme	dusey	350	3	455	2	355	1	435	2	365	1
355	yuzey_frezeleme	dusey	365	3	455	2	350	1	435	2	360	1
350	yuzey_frezeleme	dusey	360	3	455	2	365	1	435	2	355	1
450	kademe_frezeleme	dusey	455	3	360	2	355	1	435	2	365	1
445	kademe_frezeleme	dusey	455	3	360	2	355	1	435	2	365	1
440	delme	dusey	455	3	360	2	355	1	435	2	365	1

---



Şekil D.3. Örnek parça-3

---

 ISLEM PLANI
 

---

PARCA ADI : parca3  
 TASLAK OLCUSU : 155\*75\*65  
 MALZEME : C 1040  
 SERTLIK : 22 HRC

---

OP NO	VAR NO	OZELLIK	OPERASYON	TAKIM	ILERL	DEVIR SAYISI	TAKIM OMRU	TEZGAH
10	515	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
20	505	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
30	475	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
40	470	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
50	460	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
60	410	duzlem_yuzey	yuzey_frez.	alin_freze1	196	314	480	freze1
70	510	kanal	kanal_frez.	parmak_frz1	1061	1698	480	isl_merk1
80	465	delik	delme	helisel_matk1	1989	40	480	matkap1
90	420	delik	delme	helisel_matk1	1989	40	480	matkap1
100	415	delik	delme	helisel_matk1	1989	40	480	matkap1

---



---

 SECILEN TAKIMLAR
 

---

PARCA ADI : parca3

---

OPNO	TAKIM	EKSENEL ACI	RADYAL ACI	KILAVUZ ACISI	CAP	DIS SAYISI	TAKIM MALZ.	TAKIM KODU
515	alin_freze1	p	p	45	120	8	karbid	p_20
505	alin_freze1	p	p	45	120	8	karbid	p_20
475	alin_freze1	p	p	45	120	8	karbid	p_20
470	alin_freze1	p	p	45	120	8	karbid	p_20
460	alin_freze1	p	p	45	120	8	karbid	p_20
410	alin_freze1	p	p	45	120	8	karbid	p_20
510	parmak_freze1	p	p	0	30	4	hss	ts303_16
465	helisel_matkap1	p	p	0	16	2	hss	ts62_4
420	helisel_matkap1	p	p	0	16	2	hss	ts62_4
415	hellisel_matkap1	p	p	0	16	2	hss	ts62_4

---

---

BAGLAMA

---

PARCA ADI : parca3

---

OZELLIK NO	OPERASYON	ISLEME KONUMU	DESTEK.		1.YER		2.YER		1.BAG.		2.BAG	
			YUZEYI NO	Ns	YUZEYI NO	Ns	YUZEYI NO	Ns	YUZEYI NO	Ns	YUZEYI NO	Ns
515	yuzey_frezeleme	dusey	505	3	470	2	475	1	410	2	460	1
505	yuzey_frezeleme	dusey	515	3	470	2	475	1	410	2	460	1
475	yuzey_frezeleme	dusey	460	3	515	2	470	1	505	2	410	1
470	yuzey_frezeleme	dusey	410	3	515	2	475	1	505	2	460	1
460	yuzey_frezeleme	dusey	475	3	515	2	470	1	505	2	410	1
410	yuzey_frezeleme	dusey	470	3	515	2	475	1	505	2	460	1
510	kanal_frezeleme	dusey	515	3	470	2	475	1	505	2	460	1
465	delme	dusey	475	3	515	2	470	1	460	2	410	1
420	delme	dusey	470	3	515	2	475	1	410	2	460	1
415	delme	dusey	470	3	515	2	475	1	410	2	460	1

---