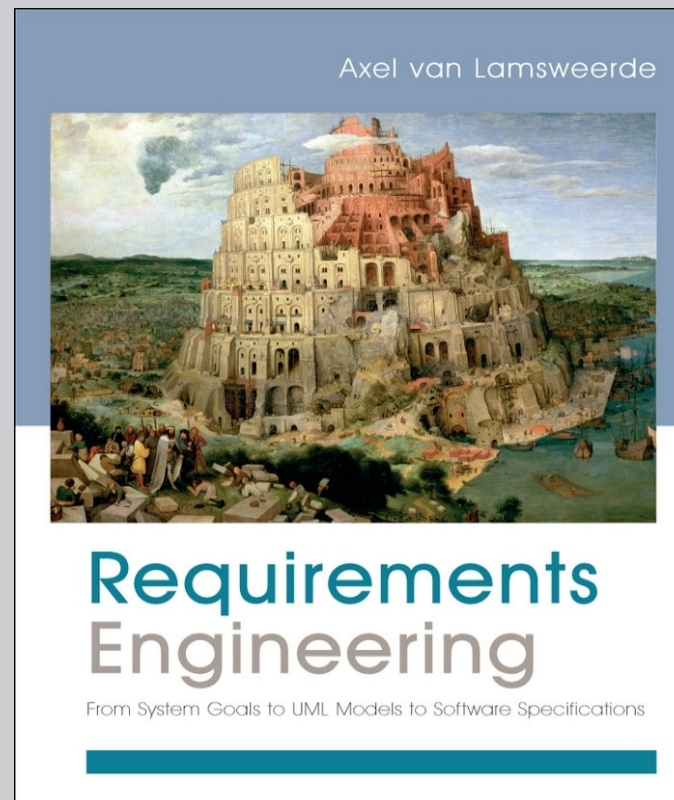
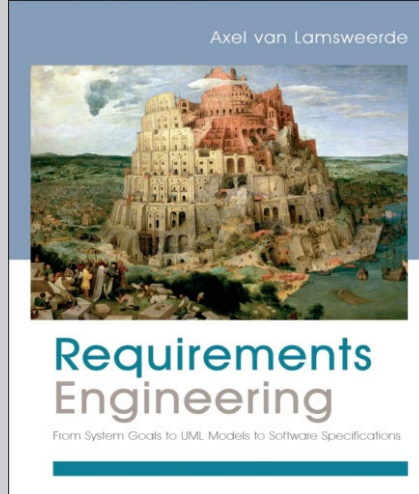


Requirements Engineering

From System Goals
to UML Models
to Software Specifications



Axel Van Lamsweerde



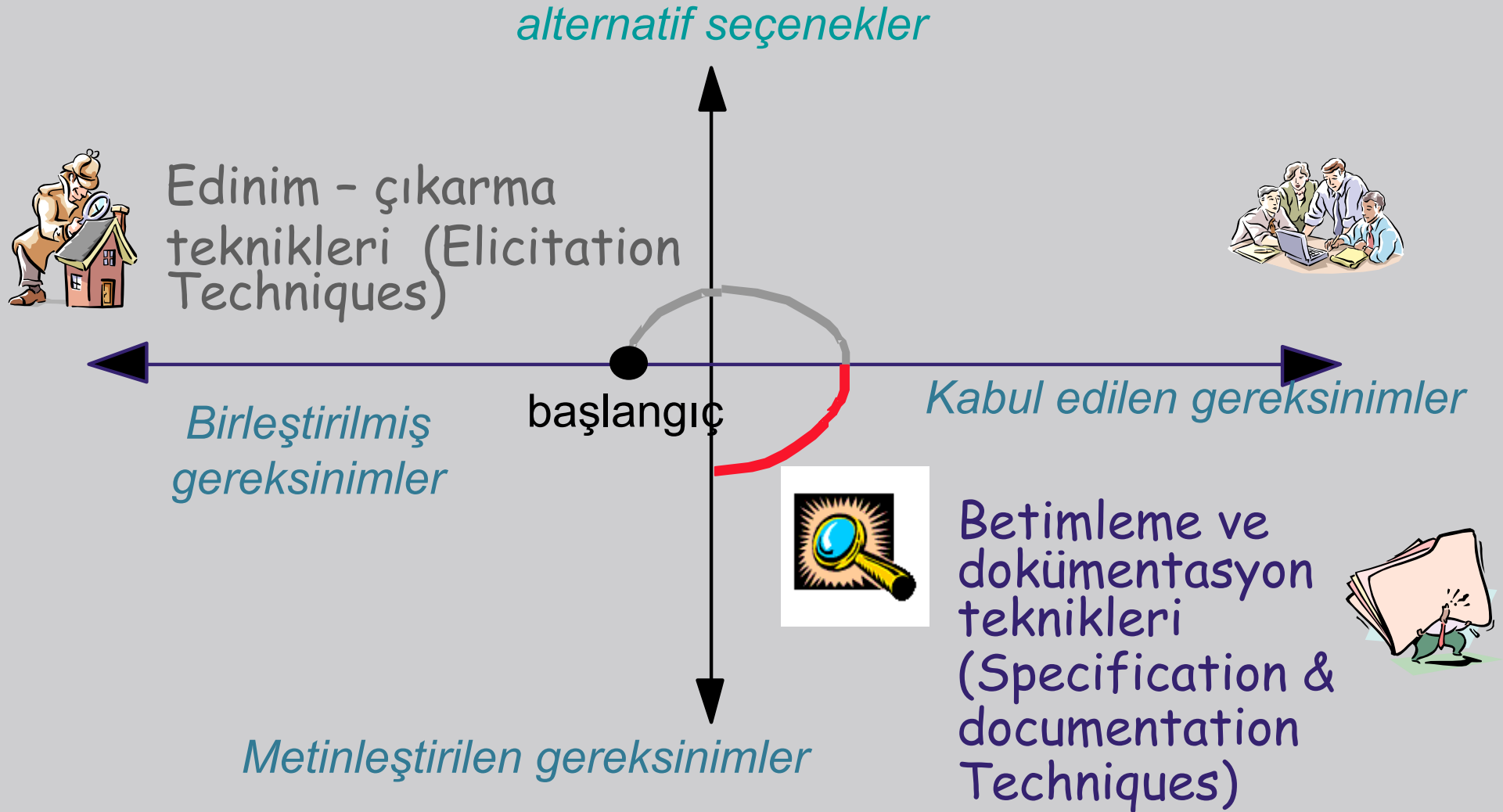
Gereksinimler Mühendisliğinin Temelleri

Bölüm 4

Gereksinimlerin Betimlenmesi ve Dokümantasyonu

Requirements Specification
& Documentation

RE ürünleri ve süreçleri (RE products and processes)



Diyagramların Kullanılması (UML- Devam)

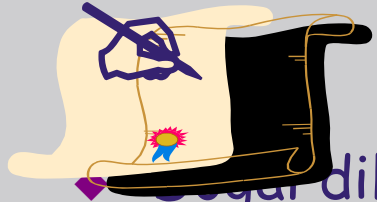
Biçimsel Betimlemeler (Kavramsal Modelleme)



Gereksinimlerin Betimlenmesi ve Dokümantasyonu

Gereksinimlerin betimlenmesi ve dokümantasyonuna ilişkin 4 farklı yaklaşım incelenecektir.

- ◆ Herhangi bir **koşul olmaksızın (unrestricted)** gereksinimlerin dokümantasyonun **doğal dilde (natural language)** hazırlanması
- ◆ Yapılandırılmamış olarak (**unstructured**) doğal dilde gereksinimlerin ifadesi
- ◆ Belli kurallara uygun olarak (**yapılandırılmış /structured**) **doğal dilde** gereksinimlerin disiplinli olarak dokümantasyonu
- ◆ Diyagram notasyonu ile gereksinimlerin belirlenmesi diğer bir ifade ile: **Görsel Diyagramların Kullanımı**
 - «use case» ve «sequence diyagramları» anlatıldı
 - Diğer görsel diyagramların incelenmesine devam edilecek.
- ◆ **Biçimsel (formal)** betimlemeler



Gereksinimlerin Doğal Dilde Koşulsuz olarak (unrestricted) Belgelendirilmesi

- ◆ Doğal dilde herhangi bir kısıtlama olmadan yazılmış gereksinim bildirimleridir (**statements**). Gereksinim olarak kabul edilen tüm deyimler ifade edilebilir.
- ◆ **Avantajları** : Sınırsız ifade istenilenin diğer ilgililere iletimini kolaylaştırır, yazılması için de özel bir bilgi gerekmez.
- ◆ **Dezavantajları:** i) Problem dünyasının (sistem gereksinimleri - çevresel olgular) herhangi bir özelliği gereksinimlerde ifade edilmeyebilir. Ya da belirtilen amaç ya da gereksinim açık değildir. Ayrıca, bazı girdilerin yazılım karşılığı olmayabilir (**omission- ihmal**)
ii) Aynı gereksinimin farklı şekillerde ifade edilmiş olması belirsizlik (**ambiguity**) oluşturur.
iii) Herhangi bir istem ifadesi problem dünyasının özelliklerine uygun olmayabilir (**inadequacy**)
iv) Gereksinim ifadesi problem dünyasının özellikleri ile hiçbir bilgi vermez (**noise**) .
v) Herhangi bir gereksinimin betimlenmesinde sorun yoktur. Fakat makine çözümü (software phenomena) olmayabilir (**overspecification**).

Gereksinimlerin Yapılandırılmamış (unstructured) ve Doğal Dilde Betimlenerek Bildirimi

“Bir trende frenleme sistemi, herhangi bir hızlanma komutu aldığı anda **ya da** belirlenenden daha yüksek bir hızda trenin istasyon bloğuna girmesi ile **ve** öndeki trene belirlenmiş değerden daha yakın olması durumunda etkinleştirilecektir»

Bu kritik **güvenlik gereksinimi** iki farklı şekilde şöyle yorumlanabilir:

Tren istasyona çok hızlı girdiği durumda

i) Tam fren sistemi eski bir komut alındığında **ya da** öndeki tren çok yaklaştığında aktive edilecektir.

if Case1 then <Statement1> F1

or if Case2 then <Statement2>

Bu bildirim hatalı sonuç verir.

ii) Tam fren sistemi **sadece** önceki tren çok yaklaştığında etkinleştirilecektir.

Doğru bildirim **If Case1 then <Statement1> F2**

and if Case2 then <Statement2> olmalıdır.

Yapılandırılmamış (Unstructured) Gereksinim Bildirimi Örneği

- ◆ F1 bildirimi önermeler mantığı (propositional logic) ile yazıldığında $(\text{not Case1 or Statement1}) \text{ or } (\text{not Case2 or Statement2})$ ifadesi $\text{not (case1 and case2) or Statement1 or Statement2}$ önermesi **True** ile sonuçlanır.

F1 bildirimi, gereksinim bildirimiminin (problem ifadesinde verilen) tümünü sağlamadığı için tanımlamadaki bazı koşullar sağlanmayacaktır.

- ❖ Çünkü koşulların irdelemesi yoktur (case 1 ve case 2)
- ❖ Statement1 ya da Statement2 ile sonuçlanacaktır.

Bu da problemin çözümü sırasında problem oluşturabilir.

Laws of Propositional Logic

De Morgan's laws	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
Absorption laws	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
Conditional identities	$p \rightarrow q \equiv \neg p \vee q$	$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Yapılandırılmamış olarak Doğal Dilde Gereksinim
Betimlemesinde koşulların sağlanması ile ilgili mantıksal
kurallar

Yapılandırılmamış (Unstructured) ve Doğal Dilde Gereksinim Bildirimi Örneği

- ◆ F2 bildirim önermeler mantığı (propositional logic) ile yazıldığında

(not Case1 or Statement1) and (not Case2 or Statement2)

ifadesi

(not Case 1 and not Case 2) or Statement1 and or Statement 2

not (Case 1 or Case 2) or (Statement1 or Statement2)

(Case 1 or Case 2) or not (Statement1 or Statement2)

(Case 1 or Case 2) and (Statement1 or Statement2)

(Case 1 and Statement1) or (Case2 and Statement2)

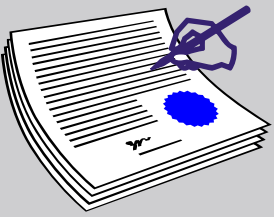
önermesi True ile sonuçlanır.

Yapılandırılmamış Gereksinim Bildirim Örneği

- ◆ (Case 1 and Statement1) or (Case2 and Statement2) olarak elde edilen F2 bildirimini sonucu da problemin gereksinim ifadesini tümüyle simgelememektedir.
- ◆ Çünkü sonuç olarak elde edilen or bildirimini sadece tek tarafının gerçekleşmesi ifadenin T sonucu vermesi için yeterlidir.
- ◆ problem için tanımlanan , case 1: herhangi bir hızlanma komutu alınması, case 2:belirlenenden daha yüksek bir hızda istasyon bloğuna girilmesi, case 3: öndeki trene belirlenmiş değerden yaklaşma önermelerinden **ya** case 1 **ya da** case 2 **True** ise gereksinim sonucu **T** olacaktır; **YA DA** case 3 **True** ise gereksinim sonucu **T** olacak , fren sistemi devreye girecektir.

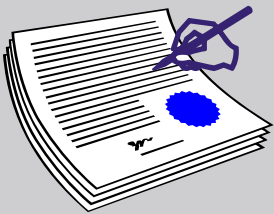
Bıradan **ya** sadece trenlerin yaklaşması koşulu (case 3) ile **ya da** bu koşul (case 3) hiç gerçekleşmeden, sadece **case 1** ya da **case2** ile gereksinim ifadesinin doğru sonuçlanması demektir.

Problem ifadesi ile ÇELİŞME ortaya çıkmıştır.



Yapılandırılmış (Structured) olarak Doğal Dilde Gereksinim Bildirimi

- ◆ Yapılandırılmadan (unstructured) doğal dilde belgelendirilmiş gereksinimlerdeki **problemlerin düzeltilmesi gerekir.**
- ◆ Yeterli ve etkin teknik çözümlenmeleri sağlayabilmek için kurallara uygun olarak, daha disiplinli bir gereksinim belgelendirmesi gerçekleştirilmesidir.
- ◆ **Gereksinimlerin belgelendirilmesi için KARAR TABLOLARI oluşturulabilir.**
- ◆ Karmaşık kombinasyonların oluşturulması için, if-then koşullarından ve ön tanımlı deyim şablonlarından genişletilerek elde edilir.



Gereksinim Bildirimlerinin Kurallı Yazıldığı Gereksinim İfadesi

- ◆ Koşulların karmaşık kombinasyonu için karar tabloları (decision tables)
- ◆ Karar tabloları karmaşık if-then koşullarını yapılandıran standart bir tekniktir.

Girdi if-koşulları

Doğruluk değerlerinin girilmesi

Train receives outdated acceleration command	T	T	T	T	F	F	F	F
Train enters station block at speed $\geq X$ mph	T	T	F	F	T	T	F	F
Preceding train is closer than Y yards	T	F	T	F	T	F	T	F
Full braking activated	X		X		X			
Alarm generated to station computer	X	X	X	X				

Çıktı then-koşulları

bir durum= AND-kombinasyonu

- ◆ Sistemattir, oluşturulması basittir ve yorumlanmasında avantajları vardır

Tablonun ve Yorumlamanın Basitleştirilmesi

Tamlık Kontrolü (Completeness check): Tüm tablo için 2^N sütun gereklidir.

Tablonun basitleştirilmesi (Table reduction):

- i) Giriş koşullarının (input conditions) AND kombinasyonu, alan özelliklerini belirlemediği durumda o sütun göz önüne alınmayabilir.
- ii) Giriş koşullarının aynı çıktı kombinasyonları şeklinde sonuçlanması durumunda iki sütun birleştirilebilir. Örnekte birinci ve üçüncü sütunlar birleştirilebilir. Çünkü her iki sütunda ikinci koşullardan biri T diğeri F dir. Aynı doğruluk tablosu vermemektedir.

Karar tabloları (Decision Table) **KABUL TESTİ**ni (Acceptance Test) sağlar. Her sütun farklı bir **giriş-çıkış test verisidir**. Bu test **Kara Kutu** (Black-Box) sınıflandırmasına aittir.

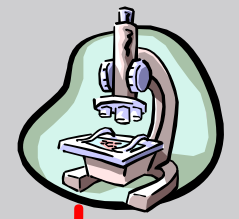
Gereksinim Betimlemeleri için Biçimsel Betimleme Kurallarının Kullanımı

Teknik belgelerin hazırlanması için kullanılan standart ilkeler gereksinimler belgesinin hazırlanmasına uyarlanır.

- Bildirimi ifade etmeden önce ne olduğunun açıklanması
- Önce motivasyonun (ifade edilecek olan bildirim) sağlanması, daha sonra özetlenmesi
- Her kavram kullanılmadan önce tanımlanmış olmalı
- Bildirimin uygunluğunun (relevant) , anlaşılabilirliğinin (comprehensible) yeterliliğinin (enough) kontrol edilmesi
- Bir cümlede hiçbir zaman birden fazla gereksinim, tahmin (assumption) ya da alan özelliği (domain property) olmaması.
- Cümlelerin kısa olması
- Gereksiz kısıtlamalar kullanılmamalı
- Soyut ifadeleri açıklığa kavuşturmak için akılcı örnekler kullanılmalı
- Parçalar arasındaki karmaşık ilişkiler için diyagram oluşturulması



Gereksinimlerin Betimlenmesi ve

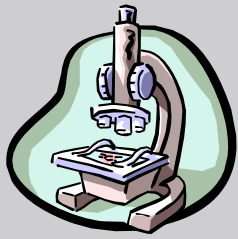


Dokümantasyonu için Biçimsel Betimlemeler

Biçimsel Betimlemeler (Formal Specification) aşağıdaki gibi farklı betimlemelerle gerçekleştirilir.

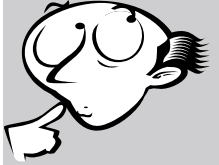
- Bildirimlerin gösterimi için temel mantıksal bilgiler (Logic as a basis for formalizing statements)
- Hikaye odaklı betimleme (History-based specification)
- Durum odaklı betimleme (State-based specification)
- Olay odaklı betimleme (Event-based specification)
- Cebirsel betimlemeler (Algebraic specification)





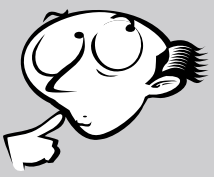
Biçimsel Betimleme Teknikleri (Formal Specification Techniques)

- ◆ Logic : bildirimlerin formüle edilmesinin temelidir
 - Propositional logic (önermeler mantığı)
 - First-order predicate logic (birinci dereceden yüklem mantığı)
 - First-order specification languages (birinci dereceden betimleme dilleri)



Biçimsel Betimleme Nedir ve Niçin Gereklidir?

- ◆ Biçimsel = makinede işlenebilen dil
 - Çoğunlukla matematiksel mantıkla ifade edilir.
 - Yeni bilginin sentaksı, semantiği ve çıkarım kurallarıdır
- ◆ Yararları
 - Bildirimin formüle edilmesinde daha yüksek kesinlik
 - Yorumlamada daha kesin kurallar
 - Daha karmaşık kontroller ve türetmelerin otomasyonu



Biçimsel Betimleme Niçin Gereklidir?



- ◆ Biçimsel betimleme doğal dildeki ve diyagram olarak verilen (görsel) betimlemeleri tamamlar.
- ◆ Özellikle kritik çözümler için önemlidir.
- ◆ Gereksinimlerin Dokümantasyonu 2 kısımda gerçekleştirilir
 - Bildirim Bölümü (Declaration Part): Parçanın yapısıdır (Örneğin diyagramlar...)
 - Beyan kısmı (Assertion Part): Parçanın özellikleri - (Örneğin «prescriptive», «descriptive» tanımlamalar...)
 - Bu tanımlamalarla büyük parça betimlemelerden (coarse grained) daha küçük parçacıklı (fine grained) betimlemelere geçiş mekanizmasının yapılandırılması sağlanır.



Önermeler Mantığı : Sentaks

Propositional logic: syntax

- ◆ Mantıksal sistem en basit biçimsel sistemdir.
- ◆ Önermeler, and (\wedge), or (\vee), not (\neg), implies (\rightarrow), equivalent to (\leftrightarrow) gibi mantıksal birleştiriciler ile rekürsif(özyineli) olarak birleştirilir.
 - İfade edilebilirlik sınırlıdır: Değişken içermez
 - Miktar ,çokluk (quantification) içermez

Dilbilgisi kurallarına uygun olarak well-formed bildirimler **sözdizim (sentaks) kurallarıdır.**

Biçimsel Betimlemelerle Örnek Bir Gereksinimin Belirtilmesi

- ◆ Trenlerin seyir halindeyken de, istasyonda seyir dışında bekleme durumundayken de kapılarının kapalı tutulması gerekmektedir.
- ◆ Bu gereksinimin biçimsel (formal) betimlemelerle ve çıkarım kurallarından biri ile ifade edilmesi istenmektedir.
- ◆ Problemin gereksinim ifadesi sonraki slaytta verilmiştir.

Bunun için aşağıdaki cümleler kullanılmıştır:

P: Trenin hareket halinde olması durumunun simgelenmesi

Q: Trenin sefer dışında olarak beklemesi durumunun simgelenmesi

R: Trenin kapılarının kapalı olması durumunun betimlenmesi



Önermeler Mantığı: İspat Teorisi

Propositional logic: proof theory

- ◆ Mevcut deyimlerde çıkarım kuralları (inference rules) ile yeni deyimler oluşturulması
 - sound rule: herhangi bir yorumlamaya (interpretation) göre eğer sonuç doğru ise, varsayım-önermeyi (premise) true yapar.

- ◆ Çıkarım kuralları (inference rules)

$$\frac{P \rightarrow Q, P}{Q}$$

$$\frac{P \rightarrow Q, Q \rightarrow R}{P \rightarrow R}$$

$$\frac{P \vee Q, \neg P \vee R}{Q \vee R}$$

- ◆ Yukarıdaki 3. kuralın uygulaması

$\neg \text{trainMoving} \vee \text{doorsClosed}, \text{trainStopped} \vee \text{trainMoving}$
kurallarından

$\text{doorsClosed} \vee \text{trainStopped}$ sonucu elde edilir.



TABLE 1 Rules of Inference.		
<i>Rule of Inference</i>	<i>Tautology</i>	<i>Name</i>
$\frac{p}{\therefore p \vee q}$	$p \rightarrow (p \vee q)$	Addition
$\frac{p \wedge q}{\therefore p}$	$(p \wedge q) \rightarrow p$	Simplification
$\frac{p}{q}$ $\frac{q}{\therefore p \wedge q}$	$((p) \wedge (q)) \rightarrow (p \wedge q)$	Conjunction
$\frac{p}{p \rightarrow q}$ $\frac{p \rightarrow q}{\therefore q}$	$[p \wedge (p \rightarrow q)] \rightarrow q$	Modus ponens
$\frac{\neg q}{p \rightarrow q}$ $\frac{p \rightarrow q}{\therefore \neg p}$	$[\neg q \wedge (p \rightarrow q)] \rightarrow \neg p$	Modus tollens
$\frac{p \rightarrow q}{q \rightarrow r}$ $\frac{q \rightarrow r}{\therefore p \rightarrow r}$	$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$	Hypothetical syllogism
$\frac{p \vee q}{\neg p}$ $\frac{\neg p}{\therefore q}$	$[(p \vee q) \wedge \neg p] \rightarrow q$	Disjunctive syllogism

Çıkarım Kuralları