Ayrık Matematik
(Ayrık İşlemsel Yapılar)

Fırat İsmailoğlu, PhD

Hafta 3:
LOJİK II

Hafta 3
Plan

1. Predicate Logic (Yuklem Mantigi)
2. Evrensel Niteleyici
3. Varliksal Niteleyici
4. Niteleyicilerin Sirasi
5. De Morgan Kuralı

Boolean Logic – Predicate Logic (Yüklem Mantığı)

Önceki hafta gördüğümüz önermesel mantık, bilgisayar bilimleri ve matematikte karsilastigimiz butun turdeki ifadeleri anlatmaya yetmez.

Örnek olarak 'x *bir asal sayıdır*' bir önerme degildir (yani dogru yada yanlistir diyemeyiz), cunku x bir degiskendir, gerçek degerini bilemeyiz.

Yine, '*tüm kuşların uçtuğu yanlıştır*' önermesi ile '*bazı kuşlar uçamaz*' önermesi, önermesel mantıkta ayrı ayrı ifade edilirler ki bunlar anlamsal olarak birbirine denktir.

p: *tüm kuşların uçtuğu yanlıştır.*

q: *bazı kuşlar uçamaz.*

p ve q önermelerinin arasindaki ilişki önermesel mantık ile ortaya konulamaz (p ve q arasindaki iliski nedir?)

Boolean Logic – Predicate Logic (Yüklem Mantığı)

Yada örneğin aşağıdaki gibi iki ifademiz olsun:

- Her bilgisayar mühendisligi ogrencisi ayrık matematik dersini alır.
- Mehmet bir bilgisayar muhendisligi ogrencisidir.

Buradan Mehmet'in ayrik matematik dersini alacagi açiktir. Fakat bu çıkarım önermesel mantık ile ifade edilemez.

Bu nedenlerden dolayı önermesel mantıktan daha güçlü bır mantığa ihtiyacimiz vardir. Bu yeni guclu mantığa yüklem mantığı (predicate logic) diyecegiz.

Yüklem (predicate):

Yüklem, doğru yada yanlış degerlerini (boolean değerler) alan bir *fonksiyondur*. Genelde büyük harfle gösterilir.

$$P: S \rightarrow \{doğru, yanlış\}$$

S burada herhangi bir kümedir, ve P, S üzerine bir yüklemdir deriz.

ör. $P: \mathbb{Z}^+ \rightarrow \{doğru, yanlış\}$ fonksiyonu aldiği pozitif tam sayı asal olma *özelligini taşıyorsa*, bu sayıyı $doğru$'ya; bu özelligi taşımıyorsa sayıyı $yanlış'$a götürsun.

$P(2) = doğru; P(4) = yanlış.$

Bu fonksiyunun adına anlaşılabilir olsun diye $'asaldır'$ diyeceğız. ($asaldır(5) = doğru$)

Başka yüklem fonksiyonu örnekleri:

P: üçe tam olarak bölünür. $x = 8$ olsun, $P(x) = yanlış$ (8, 3'e tam olarak bölünmez).

R: en sevdigi sehir Sivas'tır. $b =$ Mehmet. $R(b) = doğru.$(Mehmet'in en sevdigi sehir Sivastir)

Yüklem fonksiyonlari birden fazla değişken de alabilir.

ör. $P: \mathbb{Z}^+ \times \mathbb{Z}^+ \rightarrow \{doğru, yanlış\}$ fonksiyonu eğer aldigi ilk degisken aldigi ikinci degiskenin tam sayı katı ise $doğru$'ya; degilse $yanlış'$a gitsin.

$P(16,2) = doğru$ (çünkü $2^4 = 16$)

$P(16,3) = yanlış$ (çünkü $3'$ün hicbir tam sayi kati 16'ya eşit değildir.)

Niteleyiciler (Quantifiers)

Yüklem fonksiyonlarinda değişkene bir değer vererek bir önerme elde edebiriz.

66 asal sayidir (yanlis onerme) ($asaldır(66) = yanlış$)

Yüklem fonksiyonlarindan önerme elde etmenin bir başka yolu niteleyiciler kullanmaktir.

Bunlar Evrensel (Universal) ve Varlıksal (Exisitential) niteleyicileridir.

Evrensel Niteleyici (her '∀')

Yüklem fonksiyonun tanım kumesindeki her elemanını aynı anda hesaba kattığımız bir önerme oluşturur. Oluşan önerme formal olarak şöyle gösterilir:

$$\boxed{\forall x \in S: P(x)}$$

ör. $\forall x \in \mathbb{Z}^+ : asaldır(x)$ önermesi yanlış bir önermedir.

bu bir önerme

ör. Her canlı ölümlüdür önermesini şu şekide gösterebiliriz.

$$\forall c \in C: \, ölümlüdür(c)$$

(C burada canlılar kümesini gösteriyor)

Varlıksal Niteleyici (vardır '∃')

Yüklem fonksiyonun tanım kumesindeki en az bir eleman için geçerli bir önerme oluşturur.

Oluşan önerme formal olarak şöyle gösterilir:

$$\boxed{\exists\, x \in S: P(x)}$$

ör. Asal bir tam sayi vardır: $\exists x \in \mathbb{Z}^+: asaldır(x)$ önermesi doğru bir önermedir.

ör. $\exists m \in \mathbb{Z}^+: m^2 = m$ (karesi kendine eşit olan bir tam sayı vardır) doğru önerme

ör. $\exists k \in K: uçabilir(k)$ (uçabilen kaplumbağa vardır) (K, kaplumbağalar kumesi)

ör. $\exists x \in \mathbb{R}, \exists y \in \mathbb{R}$: $\mathrm{x} + \mathrm{y} = 1$ (toplamlari 1'eşit olan iki reel sayı vardır) ör. Daha önce gördüğümüz $asaldır$ yüklemini formal olarak şöyle ifade edebiliriz.

$$asaldır(x) \quad = \quad x \geq \mathbb{Z}^{\geq 2} \wedge [\forall\, b \in \mathbb{Z}^{+}\colon (b|x \Rightarrow b = 1 \vee b = x)]$$

her tamsayı için,
eğer sayı x i böler ise

bu sayı 1 'dir yada

x'e eşittir.

Niteleyicileri Döngü Olarak Düşünmek

$\forall\, x \in S\colon P(x)$ önermesinin doğru olması için $P(x)$'in <u>tüm</u> x'ler için doğru olması gerekir. Başka bir ifadeyle S kümesinin elemanlarini tararken hiçbir zaman $\sim P(x)$'in doğru olduğu bir $x \in S$ bulmamız gerekir, eğer böyle bir eleman bulursak önermemiz yanlıştır.

Bunu for loop ile ifade edebiliriz:

```
for x in S{
     if ~ P(x){
       return false}
return true}
```

Benzer şekilde $\exists\, x \in S: P(x)$ önermesinin doğru olması için $P(x)$'in *bir tane* x için doğru olması yeterlidir. Bunu ise for loop ile şu şekilde ifade edebiliriz:

```
for x in S{
     if P(x){
        return true}
return false}
```

Evrensel (∀) ve varliksal (∃) niteliyecileri anlamin bir baska yolu da söyledir:

$$\forall\, x \in \{x_1, x_2, \dots, x_n\}: P(x) \equiv P(x_1) \wedge P(x_2) \wedge \cdots \wedge P(x_n)$$

hepsinin doğru olması gerekir.

$$\exists x \in \{x_1, x_2, \dots, x_n\}: P(x) \equiv P(x_1) \vee P(x_2) \vee \cdots \vee P(x_n)$$

bir tanesinin doğru olması yeterlidir.

Teorem

Totoloji, önermesel mantıkta her zaman doğru olan önerme idi. Yüklem mantığında bunun karşılığı teoremdir.

Teorem, yüklem mantığında her zaman doğru olan önerme denir.

ör 1. $\forall x \ \in S: [P(x) \ \vee \sim P(x)]$

önermesi P yüklemi ne olursa olsun her zaman doğrudur. Şu halde bu önerme bir teoremdir.

P yüklemi $asaldır$ olsun. $\forall x \ \in \mathbb{Z}: [P(x) \ \vee \sim P(x)]$ (her tamsayı ya asaldır yada asal değildir

ör 2. $[\forall x \ \in S: P(x)] \vee [\forall x \ \in S: \sim P(x)]$ önermesi bir teorem değildir. Örnek olarak S kümesi yerine $\mathbb{Z}$ ve P yerine $asaldır$ yüklemini ele alalım. Bu durumda önerme şöyle olur:

$$[\forall x \ \in \mathbb{Z}: asaldır(x)] \vee [\forall x \ \in \mathbb{Z}: \sim asaldır(x)]$$

Bu ifadenin okunuşu: tüm tamsayılar ya asaldır yada tüm tamsayılar asal değildir. Bu önerme yanlıştır. $x = 4$ olsun $asaldır(4)$ yanlıştır. Veya'nin sol tarafi yanlış olur. $x = 5$ olsun $\sim asaldır(5)$ yanlıştır; böylece veya operatörünün sağ tarafı da yanlış olur.

ör. her harf ya seslidir yada sesli değildir $\not\equiv$ her harf seslidir yada her harf sesli değildir.

ör. $\exists\, x \in S : [P(x) \vee Q(x)] \iff [\exists\, x \in S: P(x)] \vee [\exists\, x \in S: Q(x)]$

ör. $\exists\, x \in S : [P(x) \wedge Q(x)] \Rightarrow [\exists\, x \in S: P(x)] \wedge [\exists\, x \in S: Q(x)]$

ör. $\forall\, x \in \emptyset: P(x) \equiv \sim \exists x \in \emptyset: P(x)$

Yukarıdaki önermelerin tamamı teoremdir.

Bir önermenin totoloji olup olmadiğini doğruluk tablosu olusturarak ortaya koyuyorduk. Teoremleri ispatlarken ise belirli,sabit bir yöntem yoktur.

Dahası girilen bir önermenin teorem olup olmadığını çıktı olarak verecek bir algoritma yoktur. Bu, verilen bir bilgisayar programın sonsuz döngüye girip girmeyeceğine (her zaman sonlanacağına) karar veren bir programın yazılamayacağı gerçeğine paralel bir gerçekliktir.

Niteleyicileri Olumsuzlaştırmak (değilini almak)

Önerme 1: ‘*Tüm bilgisayar mühendisleri gözlüklüdür*’. Bir ve yalnız bir tane dahi gözlüklü olmayan bir bilgisayar mühendisi bulursak bu iddamız çürür. Bu durumda iddiamız değil, iddiamızın tersi (olumsuzu, değili) doğru olur.

Önerme 1’in formal hali: $\forall\, b \in B: gözlüklüdür(b)$ (B bilgisayar mühendisleri kümesi)

Bu önermenin tersi: $\exists\, b \in B: \sim gözlüklüdür(b)$ (vardır bir bil. Müh. öyleki gözlüklü değil)

burada $\sim gözlüklüdür(b)$ doğru
($gözlüklüdür(b)$ yanlış)

Önerme 2: ‘*Kanatlı bir köpek vardır*’. Bu önermenin tersi ‘*kanatlı bir köpek yoktur*’, yani *tüm köpekler için kanadın olması yanlıştır*.

Önerme 2’nin formal hali: $\exists\, k \in K: kanatlıdır(k)$

Bu önermenin tersi: $\forall\, k \in K: \sim kanatlıdır(k)$

ör. Kanadı olamayan bir kuş vardır: $\exists\, k \in K: \sim kanatlıdır(k)$

Tersi: Kanadı olmayan bir kuş yoktur $\equiv$ Tüm kuşların kanadı vardır: $\forall\, k \in K: kanatlıdır(k)$

Teorem 1: $\sim [\forall\, x \in S: P(x)] \Leftrightarrow \exists\, x \in S: \sim P(x)$

De Morgan Kuralı

1. İspat: $\sim [\forall\, x \in S: P(x)] \Leftrightarrow \sim [P(x_1) \wedge P(x_2) \wedge \cdots] \Leftrightarrow \sim P(x_1) \vee \sim P(x_2) \vee \cdots$

$\Leftrightarrow \exists\, x \in S: \sim P(x)$

2. İspat

$\Rightarrow$

$\sim [\forall\, x \in S: P(x)]$ doğruysa $\forall\, x \in S: P(x)$ yanlıştır, yanı bütün örnekler için $P(x)'$ doğru olduğu yanlıştır, demek ki en az bir x $P(x)$ yanlıştır: $\sim P(x)$ doğrudur ($\exists\, x \in S: \sim P(x)$)

$\Leftarrow$

$P(x)$ yanlış olduğu ($\sim P(x)$'in doğru) bir tane $x \in S$ bulabiliyorsak, $\forall\, x \in S: P(x)$ yanlış olur, bu durumda $\sim [\forall\, x \in S: P(x)]$ doğru olur.

Teorem 2: $\sim [\exists\, x \in S: P(x)] \Leftrightarrow \forall\, x \in S: \sim P(x)$

De Morgan Kuralı

İspat: $\sim [\exists\, x \in S: P(x)] \Leftrightarrow \sim [P(x_1) \vee P(x_2) \vee \ \ldots] \Leftrightarrow \sim P(x_1) \wedge \sim P(x_2) \wedge \ldots$

$\Leftrightarrow \forall\, x \in S: \sim P(x)$

$\Rightarrow$

$\sim [\exists\, x \in S: P(x)]$ doğruysa $\exists\, x \in S: P(x)$ yanlıştır, yani $P(x)$'in doğru olduğu bir x elemanı yoktur. Buradan tüm x'ler için $P(x)$ yanlıştır: $\forall\, x \in S: \sim P(x)$

$\Leftarrow$

$\forall\, x \in S: \sim P(x)$ doğruysa tüm x'ler için $P(x)$ yanlıştır, $P(x)$'in doğru olduğu bir tane bile x yoktur. $\exists\, x \in S: P(x)$ yanlıştır, $\sim [\exists\, x \in S: P(x)]$ doğrudur.

ör. Bazı bilgisayar programları sonsuzdur: $\exists\, b \in B: sonsuzdur(b)$

tersi: $\forall\, b \in B: \sim sonsuzdur(x)$ = Her bilgisayar programı sonsuz değildir (sonludur) ör: Hatasız kul olmaz: $\sim [\exists\, k \ \in K: hatasız(k)] = \forall\, k \ \in K: \sim hatasız(x)$ (herkes hatalıdır)

ör: Masum değiliz hiç birimiz. (Hiçbir insan masum değildir):

$$\forall\, i \ \in \text{İ}: \sim masum(i) \equiv \underbrace{\sim masum(i_1)}_{\boxed{\text{masum değil}}} \wedge \underbrace{\sim masum(i_2)}_{\boxed{\text{masum değil}}} \wedge \ldots$$

Tersi:

$\sim [\forall\, i \ \in \text{İ}: \sim masum(i)] \Leftrightarrow \exists i \ \in \text{İ}: masum(i)$ (masum bir insan vardır)

Evrensel Şartlı İfade

Evrensel şartlı ifade: $\forall\, x, P(x) \Rightarrow Q(x)$

ör. *Sarışın insanlar mavi gözlüdür:* $\forall\, i \ \in \text{İ}: sarışın(i) \Rightarrow maviGözlü(i)$

ör. *100.000* satirdan uzun her kodda bug vardır:

$\forall\, k \ \in K: yuzbinSatırdanFazla(k) \Rightarrow bugVardir(k)$

Evrensel Şartlı İfadenin Tersi

Evrensel şartlı ifadenin tersi: $\sim [\forall\, x, P(x) \Rightarrow Q(x)] \Leftrightarrow \exists\, x, P(x) \wedge\sim Q(x)$

(öyle bir x vardırki, $P(x)$ dir ama $Q(x)$ değildir)

ör. '*Sarışın insanlar mavi gözlüdür*' ifadesinin tersi:

Mavi gözlü olmayan sarışın insanlar vardır: $\exists\, i \in \text{İ}: sarışın(i) \wedge \sim maviGözlü(i)$

(Sarışın olmasina ragmen mavi gözlu olmayan biri vardir)

ör. '*100.000 satirdan uzun her kodda bug vardır*' ifadesinin tersi

100.00 satırdan uzun bug olmayan kod vardır.

$\exists\, k \in K: yuzbinSatırdanFazla(k) \wedge \sim bugVardir(k)$

Gereklilik – Yeterlilik

$\forall\, x: P(x) \Rightarrow Q(x)$ evrensel şartlı ifadesinde

1. P, Q için yeterlidir (sufficient)

2. Q, P için gereklidir (necessary)

ör. En az 35 yaşında olmak cumhurbaşkanı olmak için gereklidir.

$P = \text{cumhurbaskanidir}$ ve $Q = \text{enAzOtuzbesYasindadir}$ yüklemlerini alalım.

$\forall\, x, \text{cumhurbaskanidir}(x) \Rightarrow \text{enAzOtuzbesYasindadir}(x)$

1. Cumhurbaşkanı olmak otuzbeş yaşından fazla olmak için yeterlidir.
2. Otuzbeş yaşından fazla olmak cumhurbaşkanı olmak için gerekllidir.

ör. Sarışın olmak mavi gözlü olmak için yeterlidir.

$P = \text{sarışındır}$ ve $Q = \text{maviGözlüdür}$: $\forall\, x, \text{sarışındır}(x) \Rightarrow \text{maviGözlüdür}(x)$

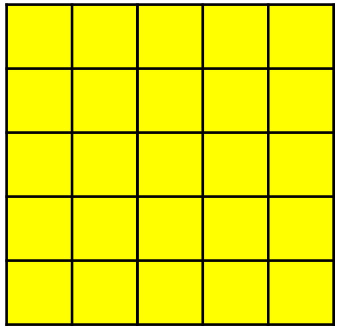
1. Sarışın olmak mavi gözlü olmak için yeterlidir.
2. Mavi gözlü olmak sarışın olmak için gereklidir.

Niteleyicilerin Sırası

İçiçe geçmiş (nested) niteleyicilerde sıra onemlidir. Aşağıdakı iki ifade *farklı* anlamlardadır:

1. $\forall\, x \colon \exists\ \mathrm{y} \colon x, \mathrm{y}$'yi sever. (Herkesin sevdigi biri vardir)
2. $\exists\ \mathrm{y} \colon \forall\, x \colon\ x, \mathrm{y}$'yi sever. (Biri herkes tarafindan sevilir)

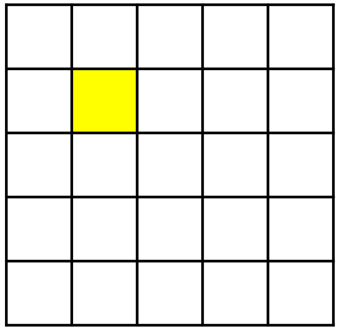
Örnekler:



$\forall\, sat \colon \forall\, kol \colon boyalıdır(\mathrm{sat}, \mathrm{kol})$

(her satır, her kolon boyalıdır)

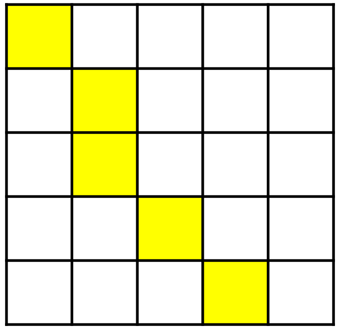
```
for s = 1:5 {
    for k= 1:5{
        if ~ boyalidir(s,k){
           return false}
}
return true
```



$\exists\ sat \colon \exists\ kol \colon\ boyalıdır(\mathrm{sat}, \mathrm{kol})$

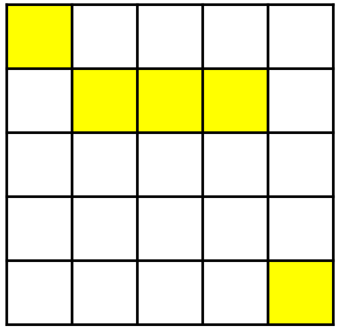
(vardır bir satır öyleki
o satırda boyalı bir kolon vardır)

```
for s = 1:5 {
    for k = 1:5{
        if boyalidir(s,k){
           return true}
}
return false
```



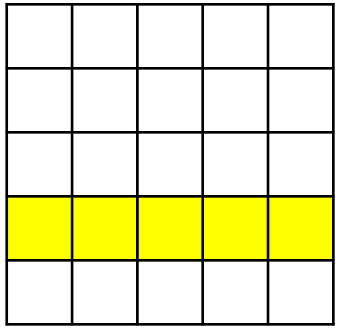
$\forall\ sat: \exists\ kol: boyalıdır(\mathrm{sat}, \mathrm{kol})$

(her satırda boyalı bir kolon vardır)



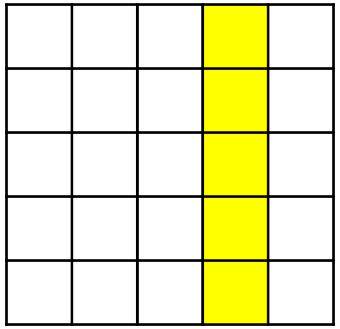
$\forall\ kol: \exists\ sat: boyalıdır(\mathrm{sat}, \mathrm{kol})$

(her kolonda boyalı bir satir vardır)



$\exists\ sat: \forall\ \ kol: boyalıdır(\mathrm{sat}, \mathrm{kol})$

(vardır bir satır öyleki
bu satırda her kolon boyalıdır)



$\exists\, kol\colon \forall\, sat\colon boyalıdır(\mathrm{sat}, \mathrm{kol})$

(vardır bir kolon öyleki
bu kolondaki her satir boyalıdır)

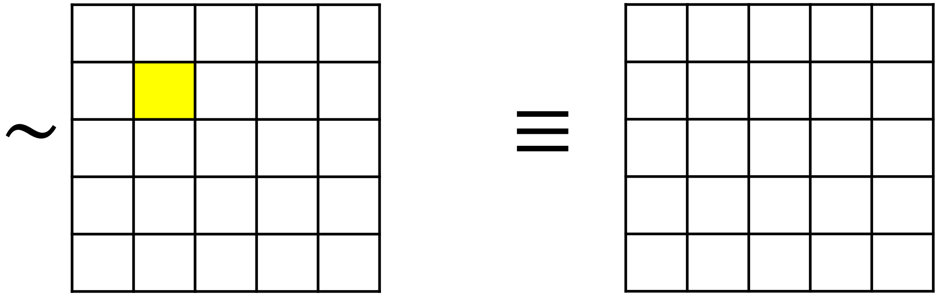
İçiçe Niteleyicileri Olumsuzlaştırmak

Burada da yine De Morgan kurali uygulanır:

vardır'ın (∃) tersi her (∀) olur; her'in (∀) tersi vardır (∃) olur.

Ve yine P yükleminin tersi , $\sim P$ olur.

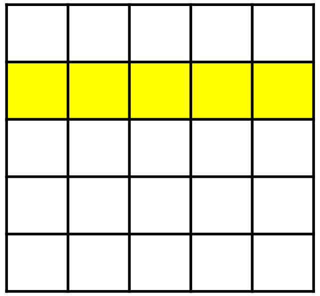
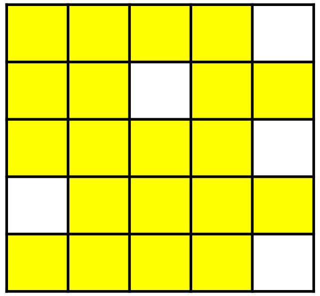
ör. $\sim [\exists\, sat\colon \exists\, kol\colon\ boyalıdır(\mathrm{sat}, \mathrm{kol})\ ] \equiv \forall\, sat\colon \forall\, kol\colon \sim boyalıdır(\mathrm{sat}, \mathrm{kol})$



ör. $\sim[\exists\, sat: \forall\ kol: boyalıdır(\mathrm{sat}, \mathrm{kol})\ ] \equiv \forall\, sat: \exists kol: \sim boyalıdır(\mathrm{sat}, \mathrm{kol})$

vardir bir satir oyleki bu satirdaki tum kolonlar boyalidir'in tersi

tum satirlarda boyali olmayan bir kolon vardir.

$\sim$  $\equiv$ 

ör. *Her iphone kullanıcısının telefonunda öyle bir app vardır ki; bu app o kullanicinin iphone kullanan butun arkadaslari tarafindan indirilmistir.*

K iphone kullanicilari kumesi, A applikasyonlar kumesi olsun.

Bu durumda yüklem mantığıiyla bu ifadeyi şu sekilde gosterebiliriz:

$$\forall\, k\ \in K: \exists\, a\ \in A: \forall\, m\ \in K: \ [arkadaştir(k, m) \Rightarrow indirmistir(m, a)]$$

$\forall\, k \in K : \exists\, a \in A : \forall\, m \in K :\ [arkadaştir(k,m) \Rightarrow indirmistir(m,a)]$

Önermesinin tersini alalim. Hatırlayalım: $\sim (\mathrm{p} \Rightarrow q) \equiv \sim (\sim \mathrm{p} \vee q) \equiv p \wedge \sim q$

$$\sim [\forall\, k \in K : \exists\, a \in A : \forall\, m \in K :\ [arkadaştir(k,m) \Rightarrow indirmistir(m,a)]]$$
$$\equiv\ \exists\, k \in K : \forall\, a \in A : \exists m \in K : \sim [arkadaştir(k,m) \Rightarrow indirmistir(m,a)]$$
$$\equiv\ \exists\, k \in K : \forall\, a \in A : \exists m \in K : \sim [arkadaştir(k,m) \Rightarrow indirmistir(m,a)]$$
$$\equiv \exists\, k \in K : \forall\, a \in A : \exists m \in K : [arkadaştir(k,m) \wedge \sim indirmistir(m,a)]$$

Yukarida son bulduğumuz satırı okursak:

Öyle bir iphone kullanicisi vardir ki, her bir applikasyon icin bu aplikasyonu indirmemis bir iphone kullanicisi arkadasi vardir.