

## LUKASIEWICZ - TARSKI NOTASYONU

*Şafak Ural*

Lukasiewicz - Tarski notasyonuna\* göre mantık ifadelerini parantezsiz olarak yazmak mümkündür\*\*. Bu notasyona göre bir ifadeyi parantez kullanmadan yazmak için, herhangi iki ifade arasında yapılacak işlemi gösteren operatör, bu iki ifadenin soluna yazılır. Matematikten bir örnek verirsek :

$a + b$   
yerine  
 $+ ab$

yazılır. Bu prensipten hareket ederek,

$a - (a + b)$   
ifadesi  
 $- a + ab$

şeklinde yazılır.

Mantıkta, temsilci harfler arasında işlem yapmaya yarayan önerme - eklemleri Lukasiewicz - Tarski (kısaca LT) notasyonunda

---

\* Bu notasyon, 1924 yılında Jan Lukasiewicz isimli Polonyalı bir mantıkçı tarafından geliştirilmiştir. Fakat, böyle bir notasyonun kurulması fikri, Lukasiewicz'in de belirttiği gibi (Nidditch 1960, S. 35), yine Polonyalı bir mantıkçı olan Alfred Tarski'ye aittir.

\*\* Peano-Russell notasyonuna göre de mantık ifadelerini parantezsiz olarak yazmak mümkündür. Bu notasyonda parantez yerine nokta kullanılır ve parantezlerin simetrik olma özelliklerinden yararlanır.

birer sembolle temsil edilirler. Uluslararası kabule göre bu semboller :

$$\begin{aligned} &\sim p ; Np \\ &p \wedge q ; Kpq \\ &p \rightarrow q ; Cpq \\ &p \vee q ; Apq \\ &p \leftrightarrow q ; Epq \end{aligned}$$

şeklinindedir. Bu sembolleri kullanarak,

$$(q \rightarrow r) \rightarrow (p \rightarrow r)$$

şeklindeki bir ifade, LT notasyonuna göre,

$$CCqrCpr$$

olarak yazılır. Fakat parantezlerin yeri değişik olsaydı,

$$\begin{aligned} &q \rightarrow [r \rightarrow (p \rightarrow r)] \\ &\text{ifadesinin} \\ &CqCrCpr \end{aligned}$$

şeklinde yazılması gerekirdi. Eğer ifadede değilme de varsa, örneğin :

$$\begin{aligned} &\sim (p \vee q) \leftrightarrow (\sim p \wedge \sim q) \\ &\text{ifadesi} \\ &ENApqKNpNq \end{aligned}$$

şeklini alır. Görüldüğü gibi, değilme tek bir temsilci harfe aitse, onun hemen solunda yer almaktadır. Halbuki, değilmenin dışında kalan semboller, *iki ifade* arasında yapılacak işleme işaret etmektedir. Genelliği en büyük olan sembol (bu son örnekte 'E' sembolü) en sola yazılmaktadır. Yapılan işlemler, aşağıdaki örnekten adım adım izlemek suretiyle daha açık olarak görülebilir :

$$[(p \wedge q) \rightarrow r] \rightarrow [(\sim r \wedge q) \rightarrow \sim p]$$

Bir ifadeyi LT notasyomma göre yazmak için sağ taraftan başlamak yerinde olur. Yani,

$\sim q$  yerine Np

yazıldıktan sonra,

$\sim r \wedge q$  yerine KNr $q$

yazılır. Daha sonra, sağdaki köşeli parantez içinde kalan bu iki ifade 'ise' eklemi ile birbirine bağlanırsa

CKNr $q$ Np

elde edilir. Aynı şekilde soldaki köşeli parantezin içi de dönüştürülür.

Yani :

CKpqr

yazılır. Daha sonra, köşeli parantezlerin oluşturduğu bu iki ana ifadeyi birbirine bağlayan eklemi temsil eden sembol ('ise' eklemi) en sola yazılırsa, yukarıdaki ifade,

CCKpqrCKNr $q$ Np

şekline getirilir.

Aynı işlemler tersinden de uygulanabilir. Yani, LT notasyonuna göre yazılmış bir ifade prantezli hale dönüştürülebilir. Böyle bir dönüştürmeyi yaparken de yine sağ taraftan başlamak yerinde olur. Örneğin :

CCKpN $q$ NrCKpr $q$

şeklindeki bir ifadenin sağındaki kısım, yani,

CKpr $q$

ifadesi sağdaki ilk bileşendir ve,

$(p \wedge r) \rightarrow q$

şeklinde yazılması gerekir. Diğer bileşen ise,

CKpNqNr

olup,

$$(p \wedge \sim q) \rightarrow \sim r$$

şeklini alır. Bu iki bileşen birbirine en soldaki 'ise' eklemiyle bağlanmıştır. Yani ifadenin tamamını;

$$[(p \wedge \sim q) \rightarrow \sim r] \rightarrow [(p \wedge r) \rightarrow q]$$

şeklinde yazmak gerekir.

LT notasyonunu niceleyicileri de içine alacak şekilde kullanmak mümkündür. Yukarıda verilen kurallar çerçevesinde kalarak, örneğin;

$$\forall xFx \leftrightarrow \sim \exists x \sim Fx$$

ifadesi

$$E \forall xFxN \exists xNFx$$

şeklini alır. LT notasyonuna göre yazılmış örneğin,

$$E \forall xCFx \exists yKGyAHyHxA \exists xKGxHxN \exists xFxK \exists xGx \vee xCFxHx$$

ifadesi,

$$\forall x \left\{ Fx \rightarrow \exists y [Gy \wedge (Hy \vee Hx)] \right\} \leftrightarrow \left\{ [ \exists x (Gx \wedge Hx) \vee \sim \exists x Fx ] \right. \\ \left. \vee [ \exists x Gx \wedge \forall x (Fx \rightarrow Hx) ] \right\}$$

şekline dönüştürülür.

LT notasyonu yalnızca bir sembolleştirme aracı olmayıp, çeşitli mantık işlemlerini yaparken de kullanılabilir\*. Böylece hesaplarda bir ekonomi sağlanabilir. Bunun için bilinen 'doğruluk toblosu', yani;

---

\* Nitekim, baştan sona kadar LT notasyonu ile yazılmış (örneğin, Prior, 1962) mantık kitapları vardır.

p	q	$P \wedge q$	K	1 0	K11=1
1	1	1	1	1 0	K10=0
1	0	0	0	0 0	tablosunu ve buradan : K01=0
0	1	0			K00=0
0	0	0			
p	q	$P \vee q$	A	1 0	A11=1
1	1	1	1	1 1	A10=1
1	0	1	0	1 0	tablosunu ve buradan : A01=1
0	1	1			A00=0
0	0	0			
p	q	$P \rightarrow q$	C	1 0	C11=1
1	1	1	1	1 0	C10=0
1	0	0	0	1 1	tablosunu ve buradan : C01=1
0	1	1			C00=1
0	0	1			
p	q	$P \leftrightarrow q$		1 0	E11=1
1	1	1	1	1 0	E10=0
1	0	0	0	0 1	tablosunu ve buradan : E01=0
0	1	0			E00=1
0	0	1			

gözönüne almak faydalıdır. Böylece, aşağıdaki ifadelerden yararlanarak sözcelimi :

$$(p \rightarrow q) \vee (q \rightarrow p) \quad \text{yani} \quad ACpqcq$$

ifadesinin doğruluk değerini bulalım :

p=1 , q=1 için	ACpqcq	ifadesi,	AC11C11 = A11 = 1
p=1 , q=0 »	»	»	AC10C01 = A01 = 1
p=0 , q=1 »	»	»	AC01C10 = A10 = 1
p=0 , q=0 »	»	»	AC00C00 = A11 = 1

Sonuç kısmını gösteren sağ taraf 'hep doğru' değerini aldığı için, sözkonusu olan ifadenin bir totoloji olduğu anlaşılır.

Bu işlem, soldaki tabloyu (her önerme eklemi için ezbere bilindiği için) tamamen terk etmek suretiyle de yapılabilir :

ECANqppCpNKqNp

ifadesinin doğruluk değeri :

$$\begin{aligned} ECAN111C1NK1N1 &= ECA011C1NK10 = EC11C1N0 = E1C11 = E11 = 1 \\ ECAN011C1NK0N1 &= ECA111C1NK00 = EC11C1N0 = E1C11 = E11 = 1 \\ ECAN100C0NK1N0 &= ECA000C0NK11 = EC00C0N1 = E1C00 = E11 = 1 \\ ECAN000C0NK0N0 &= ECA100C0NK00 = EC10C0N0 = E0C00 = E01 = 0 \end{aligned}$$

Şeklinde belirlenir. Yani bu ifade, (bir yanlış değer bulunduğundan) geçersiz ama tutarlıdır.

Bu yöntemle göre bir ifadenin doğruluk değeri tayin edilirken dikkat edilecek ilk önemli nokta, 'değilleme' işaretinin etki alanıdır. Sağdan sola doğru giderken, eğer 'değilleme' işareti tek bir temsilci harfe aitse (yani parantez önünde değilse), ilk adımda bu temsilci harfin 'doğruluk değeri'ni değiştirmeli ve 'değilleme' işareti elenmelidir. Eşitliğin sağma yazılan sonraki adımda, iki temsilci harf arasında yer alan önerme eklemine değeri bulunmalı ve böylece bu önerme eklemi elenmelidir. Bu elemanın sonucunda bir temsilci harfin solunda 'değilleme' işareti gelirse yukarıdaki işlem tekrarlanmalıdır. Bütün önerme - eklemeleri (ve 'değilleme' işaretleri) elendiğinde, verilen ifadenin doğruluk değeri de bulunmuş olur.

İçinde ikiden fazla temsilci harfin geçtiği ifadelerin doğruluk değerinin bulunmasına kullanılan Quine Metoduna da LT notasyonunun uygulamak mümkündür :

CAKpqKNpNrEqr

(i) CAK1qKN1NrEqr	(xi) CAK0qKN0NrEqr		
(ii) CAq0Eqr	(xii) CA0NrEqr		
(iii) CqEqr	(xiii) CNrEqr		
(iv) C1E1r	(ix) C0E0r	(xiv) CN1Eq1	(xvii) CN0Eq0
(v) C1r	(x) 1	(xv) C0q	(xviii) C1Eq0
(vi) r		(xvi) 1	(xix) Eq0
(vii) 1	(viii) 0	(xx) E10	(xxii) E00
		(xxi) 0	(xxiii) 1

Görüldüğü gibi, gerek yapılan sadeleştirmeler ve gerekse şema olarak Quine Metodu aynen korunmaktadır. Değişiklik yalnızca LT sistemi içinde yer alan sembollerin kullanılmasıdır.

Pratik değeri olmasa da, (Kullanılan sembollerin azlığı dolayısıyla ekonomik değeri yüksek) tek bir önerme eklemi aracılığıyla yazılmış bir ifadeyi LT notasyonuna dönüştürmek mümkündür. Bilindi-

ği gibi, çeşitli önerme - eklemelerini 'bağdaşmazlık' (Sheffer's Stroke\*) önerme - eklemi cinsinden tanımlamak mümkündür. (Aynı tanım bir de 'birlikte değilleme' adıyla bilinen eklem aracılığıyla yapılabilir). Böyle bir tanım için önce 'bağdaşmazlık' eklemine doğruluk değerini yani :

p	q	p q
1	1	0
1	0	1
0	1	1
0	0	1

ya da

D	1	0
1	0	1
0	1	1

gözünde bulundurmamak gerekir. Bu eklemine doğruluk değeri yardımıyla diğer eklemeler şöyle tanımlanır :

$$\begin{aligned}\sim p &= p|p \\ p \vee q &= (p|p) | (q|q) \\ p \wedge q &= (p|q) | (p|q) \\ p \rightarrow q &= p | (q|q)\end{aligned}$$

Bu durumda, mesela 'p∨q' ifadesi sonuçta :

$$DDppDqq$$

şeklini alır. Benzeri şekilde,

$$\sim p \wedge q$$

ifadesini önce 'bağdaşmazlık' cinsinden yazalım :

$$[(p|p) | q] | [(p|p) | q]$$

ifadesini elde ederiz. Bu ifadeyi LT notasyonuna çevirdiğimizde :

$$DDppqDDppq$$

elde edilir. Yine sözcüğümleri,

$$\sim (p \vee q)$$

ifadesini;

$$[(p|p) | (q|q)] | [(p|p) | (q|q)]$$

şeklinde yazdıktan sonra,

\* Bu operatör 'Sheffer' adıyla tanınırsa da daha önce C.S. Peirce tarafından keşfedildiği bilinmektedir. (Rkz W-M Kneale 1975. s. 423).

DDDppDqqDDppDqq

ifadesi elde edilir. Yapılan işlemleri biraz yakından izlemek amacıyla,

$$(p \wedge q) \rightarrow (p \vee q)$$

ifadesini gözönüne alalım :

$$\begin{array}{l} p \wedge q \quad \text{ifadesi} \quad (p|q) | (p|q) \quad \text{ile,} \\ p \vee q \quad \quad \quad \gg \quad \text{ise, } (p|p) | (q|q) \quad \text{ile eşdeğerdi.} \end{array}$$

Bu iki ifadeyi 'içerme' eklemiyle bağlarsak,

$$[(p|q) | (p|q)] \rightarrow [(p|p) | (q|q)]$$

elde edilir. Şimdi de 'içerme' eklemine 'bağdaşmazlık' eklemi cinsinden yazarsak :

$$[(p|q) | (p|q)] | \left\{ [(p|p) | (q|q)] | [(p|p) | (q|q)] \right\}$$

ifadesini elde ederiz. Bu ifadeyi LT notasyonuna çevirirsek :

DDDpqDpqDDDppDqqDDppDqq

böylece, içinde üç ayrı önerme eklemi geçen (eklemlerin sayısı üç den fazla da olabilirdi) bir ifadeyi tek önerme - eklemi cinsinden ve LT notasyonuna göre yazmış oluruz.

LT notasyonu, çeşitli mantık ispatlarında da uygulanabilir. Sözelimi, Kalish - Montague' veya 'ağaç Metodu' gibi yöntemlerle yapılan ispatlar LT notasyonuna çevrilerek de yapılabilir. Ancak bu uygulama, alışılmış sembolizmin tamamen dışında kalan özellikleri ve daha fazla dikkat istemesi yüzünden güçlükleri de beraberinde getirmektedir. Fakat buna karşılık ekonomik olması bakımından kesin bir üstünlüğü vardır.

#### Referanslar

- Nidditch, P.H. (1960), Elementary Logic of Science and Mathematics, University Tutorial Press Ltd., London
- Prior, A.N. (1962), Formal Logic, Oxford, Clarendon Press.
- Kneale, W, Kneale, M. (1975), The Development of Logic, Oxford.