Mantık Araştırmaları Dergisi

Journal of Logical Studies

Üç Mantıksal Teori

Yazar(lar) | Author(s): John Corcoran

Çeviren(ler) | Translator(s): Fatmanur Berilgen

Bu makaleyi kaynak gösterin | Cite this article:

Corcoran, J . "Üç Mantıksal Teori". çev. Fatmanur Berilgen, Mantık Araştırmaları Dergisi 1 (2019): 140-174

Bu makaleye çevrimiçi ulaşın | See this article online:

https://dergipark.org.tr/mader/issue/46688/574482

İletişim | Contact

Adres: Erciyes Üniversitesi İlahiyat Fakültesi, 38030 Melikgazi/KAYSERİ

mantikder@gmail.com - https://dergipark.org.tr/mader

Mantık Araştırmaları Dergisi
Cilt: 1 Sayı: 1 – 2019 / 1

Üç Mantıksal Teori*

John CORCORAN**

Çeviren: Fatmanur Berilgen***

Özet

Bu çalışma, teori olarak düşünülen mantıksal sistemlerle ilgilidir. Geleneksel olarak verilen sistemleri makul bir şekilde çözmenin hedeflendiği sorunları araştırarak biz, mantıksal sistemlere yaygın olarak uygulanan yeterlilik ölçütlerinin mantığını açıklığa kavuşturuyoruz. Bu açıdan mantıksal sistemlerin üç temel türü var gibi görünüyor: bunlar mantıksal doğruluğa ilişkin olanlar, mantıksal doğruluk ve mantıksal gerektirme ile ilgili olanlar ve mantıksal doğruluk ve mantıksal gerektirme ile olduğu kadar, başlı başına çıkarım ile de ilgili olanlardır. İlk iki tipteki sistemler için yeterlilik ölçütleri şunları içerir: etkinlik, sağlamlık, tamamlanmışlık, Post tamamlanmışlık, "güçlü sağlamlık" ve güçlü tamamlanmışlık. Bir mantıksal sistemin bir çıkarım teorisi olarak düşünülmesi, ispat sistemleri için iki yeterlilik ölçütü formüle etmeye çalışmamıza sebep olur. Birincisi, sıkılık kavramı veya delillerdeki boşluksuzluk(gaplessness) ile ilgilidir. İkincisi, bir kanıt sistemi için bir tamamlanmışlık koşuludur. Makalenin sonunda yer alan tarihsel bir not, üst sistem hiyerarşisi ile bu mantık alanının gerçek tarihsel gelişimi arasında dikkate değer bir paralellik olduğunu belirtmektedir.

• • •

ÇEVİRİ MAKALE
Geliş Tarihi: 29-05-2019
Kabul Tarihi: 18-06-2019
Yayın Tarihi 01-07-2019

Bu makale esasen mantık bilimi felsefesine bir katkıdır. Birincil amacımız, teori olarak düşünülen mantık sistemlerine belirli bir metodolojik ilke uygulamaktır. İlke şu şekilde ifade edilebilir: Tamamlanmış bir teorinin formu kısmen teori konusunun doğası, kısmen de teorinin çözmeye çalıştığı spesifik problemlerle belirlenir. Makul olarak çözülmesi amaçlanan sorunlara dayanarak, üç mantıksal sistem türünü ayırdık. İlki, eski lojistik sistemlerini içermektedir. İkincisi,

---

* Makalenin orijinali: Philosophy of Science, Vol. 36, No. 2. (Jun., 1969), pp. 153-177. Makalenin tercümesini gözden geçiren Prof. Dr. Ahmet Kayacık ve Doç. Dr. Aytekin Özel hocalarıma teşekkür ederim.

** Prof. Dr., The State University of Newyork, University of Buffalo, Philosophy Department, corcoran@buffalo.edu

*** Uludağ Üniversitesi, SBE. Yüksek Lisans Öğrencisi fatmanurberilgen@gmail.com

rastlantısal varsayım serilerinden elde edilen kanıtların hâkim bir rol oynadığı daha sonraki bir dönemin mantıksal sistemlerini içerir. Üçüncü tür, son zamanlarda ortaya çıkmış doğal çıkarım sistemlerini içerir. (İlk türün sistemlerinin örnekleri için bakınız [20] ve [11]; ikinci türün açık bir örneği için bakınız [6]; üçüncü türün örnekleri için bakınız; [19] ve [1])

Mantıksal sistemlerin bu sınıflandırması iki önemli özelliğe sahiptir. İlki, bir üçlemedir: her mantıksal sistem tam olarak üç sınıfın birinde yer alır. İkincisi, şu anlamda hiyerarşiktir. Birinci tür sistem tarafından ele alınan problemler, ek problemlerden söz eden ikinci tür tarafından da ele alınır. Aynı şey, üçüncüye nazaran ikinci için doğrudur. Sistemlerin bu hiyerarşisi, üç tür önermeler mantığı, üç tür birinci dereceden mantık, üç tür ikinci dereceden mantık, üç tür modal mantık vb. olarak tamamen geneldir. Söylenilecek her şeyin bu mantıkların herhangi birine ve tümüne uygulanması amaçlanmıştır. Ancak, her sistem türünün basit bir örneğini vermek için, tartışmanın bir kısmını cümleler mantığının (önermeler mantığı) bir parçası ile sınırlayacağız.

Örneklerin amaçları için, mantıksal kelime dağarcığında "… ise, o halde…." nin doğruluk fonksiyonu anlamını veren ⊃ sembolünün, "değil" doğrusal anlamını veren ~ sembolünün ve gruplama amacıyla parantezlerin olduğu bir L dili varsayalım. L, gerçekten de 've', 'veya', 'ancak ve ancak', herhangi bir çeşit niceleyici, modal operatörler gibi semboller içerebilir. Örneklerin amaçları için varsaydıklarımızın tümü yukarıdakilerdir ve dilin (⊃ ve ~ nin anlamlarının sabit kaldığı) her yorumu altında, her bir cümle ya doğrudur ya yanlıştır ya da ikisi de değildir. Buradan hareketle L, küme teorisinin, geometrinin, biyolojinin vb. biçimsel dili yahut sırf mantıksal amaçlar için oluşturulmuş biçimsel bir dildir

1. Mantıksal Doğruluk: Lojistik Sistemler

""Tüm mantıksal olarak 'doğru' önermelerin önceden tanımlanması mümkündür."

WITTGENSTEIN

Bir dilin doğru cümleleri arasında, kişi mantıksal olarak doğru cümlelerin bir alt sınıfını ayırt edebilir. Mantıksal olarak doğru cümleler, yalnızca mantıksal sembollerin anlamından dolayı doğru olanlardır. Daha belirgin olarak, mantıksal olarak doğru cümleler, dilin tüm yorumları altında doğru olanlardır. (Bir yorumla, mantıksal sembollerin normal anlamlarına sahip olduğu ve tüm dil bilgisel ayrımların korunduğu bir yorumu kastediyoruz: özel isimler, bireyler olarak; tek yüklemler, bireyler kümesi olarak yorumlanırlar.) Mantıksal bir teorinin akla yatkın amaçlarından biri, mantıksal olarak doğru tüm cümlelerin kodlanmasıdır. Bu amaç, özellikle tüm matematik yasalarının mantıksal olarak doğru olduğuna inanan bir mantıkçı için makuldür; bir kere tüm mantıksal doğru cümlelerin bir kodlamasını alınca, tüm matematiksel yasaların kendi sisteminde olup olmadığını görmek için "yalnızca" bir kontrole ihtiyacı kalır.

Öncelikle mantıksal olarak doğru cümleleri kodlamanın amaçlandığı mantıksal sistemler, burada lojistik sistemler olarak adlandırılmaktadır. Kuruluşu sırasında Russell, yukarıda geçen mantıksal olarak doğru fikrinin (görüşlerinin karşılaştırması için bakınız [4] syf. 221-222) farkında olmamasına rağmen Principia Mathematica'nın [20] sistemi bu tiptedir. Lewis'in sistemlerinin tümü lojistik sistemlerdir.

Genellikle bir lojistik sistem, aksiyomlardan başka cümleler meydana getiren bazı üretim kurallarından ve aksiyomlar olarak adlandırılan bir dizi cümleden_oluşur. Sistem tarafından kodlanan cümleler, aslında aksiyomlar ve kurallar vasıtasıyla aksiyomlardan meydana getirilen cümlelerdir – bunların tümü teorem olarak adlandırılır.

Örneğimizde sadece mantıksal doğruluğu '⊃' ve '~' anlamlarına çeviren L'nin mantıksal olarak doğru cümlelerini kodlayacağız. Bu dolaylı olarak yapılacaktır. Tüm örnekleri mantıksal olarak doğru olan şemaları veya cümle formlarını kodlayacağız. Genel olarak, şemalar L'nin cümleleri değil, onun örnekleridir. Örneğin, L'nin geometri dili olması durumunda $(P \supset P)$ kesinlikle bir L cümlesi değildir. Ancak, eğer a bir geometri cümlesi ise, $(a \supset a)$ yine bir geometri cümlesidir ve dahası $(a \supset a)$ bir $(P \supset P)$ örneğidir. Tüm $(P \supset P)$ örnekleri mantıksal olarak doğrudur, bu yüzden $(P \supset P)$ kodlayacağımız şema türüdür. Bunun yanı sıra, bazı $(P \supset Q)$ örnekleri mantıksal olarak doğru olup bazıları doğru olmadığı için $(P \supset Q)$ kodlamasından kaçınacağız.

Kesinlik için şemaları, şematik harflerden (P, Q, R, P_1, Q_1, R_1, P_2 …) herhangi biri ve ⊃ ve ~ ile birlikte gruplama için kullanılan parantezleri kullanarak oluşturulan herhangi bir formül olarak tanımlarız. Sadece mantıksal olarak doğru cümleleri örnek olarak alan bir şema mantık yasası olarak adlandırılır. $(P \supset (Q \supset P))$ gibi bazı mantık kanunları '...ise, o halde...' formundayken $\sim\sim(P \supset P)$ gibi bazıları değildir. '...ise, o halde...'

formları, özellikle geçerli çıkarımlardaki rolleri nedeniyle önemlidir: geçerli yorumlar yapmadaki rolleri dolayısıyla; yukarıda yasa olarak kullanılan a'dan elde edilen a ve b cümleleri ile geçerli bir şekilde çıkarım yapabiliriz (b ⊃ a). Bazı mantıkçılar, mantıksal kanunlar sınıfının bu uygun alt sınıfını mantığın temel konusu olarak görürler ([2],syf. 2 ve 3).

Lojistik sistem teoremler denilen bir şemalar sınıfını kodlayacaktır. Bu teoremler, sadece söz konusu mantıksal yasaların tümünü içermeyi amaçlamaktadır. Teoremler sınıfı, bazı mantıksal yasaları aksiyomlar olarak alarak ve diğerlerinin aksiyomlardan elde edilebileceği işlemler vererek tanımlanır.

Lojistik sistem LS, aşağıdaki beş tanım ile tanımlanır.

Tanım: Aşağıdaki üç şema LS'nin aksiyomlarıdır.

(A1) $(P \supset (Q \supset P))$

(A2) $((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$

(A3) $((\sim P \supset \sim Q) \supset ((\sim P \supset Q) \supset P))$

Tanım: Φ şeması ancak ve ancak Φ şemasının, δ şemasındaki tek bir şematik harfin tüm oluşumlarının η şeması ile değiştirilmesi veya bir dizi şematik harf üzerine aynı anda yerine koyma işleminin uygulanması sonucu, δ şemasından elde edilir şeklinde tanımlanır.

Örneğin: Her Φ şeması P şemasından yerine koyma yoluyla elde edilir çünkü P'nin tüm oluşumlarını (bu durumda sadece bir tane) Φ yerine koyabiliriz. Bilhassa ~P, P'den yerine koyma yoluyla elde edilir. Tabii ki, yerine koyma çıkarımın sağlam bir kuralı değildir, ancak mantık yasalarından mantık yasaları üretmenin bir yoludur: ör; A1 de Q yerine $(P \supset \sim P)$ koymak başka bir mantıksal yasa olan $(P \supset ((P \supset \sim P) \supset P))$ üretir.

Tanım: Bir δ şemasının ancak ve ancak $\eta = (\Phi \supset \delta)$ ise çıkış yoluyla Φ ve η şemalarından elde edileceği söylenir.

Örneğin; $(Q \supset P)$ çıkış yoluyla P ve $(P \supset (Q \supset P))$ den elde edilir. Bu işlem, çıkarımın geçerli bir kuralı olarak hemen tanınır ama çıkarımın geçerli bir kuralı olmasının eldeki problem ile alakası yoktur. Demek ki tüm şemaların ve onların tüm örneklerinin kodlanması problemi, uygun şeklin mantıksal olarak doğru cümleleridir.

Şimdi LS de türetmeleri tanımlayabiliriz. Türetmeler, sistem tarafından kodlanan belli şemaların ispatları olarak sunulabilen şemaların listeleridir. Türetmeler, sadece kullanılan ilkelerin sağlam olmaması ve olağan akıl yürütmelere uygun olmaması haricindeki mantıksal akıl yürütmeyi içeren kanıtlara uygundur.

Tanım: LS deki bir türetme, bir aksiyom ile başlayan şemaların sınırlı bir sırasıdır ve öyle ki listedeki takip eden her şema ya bir aksiyomdur ya yerine koyma yoluyla bir önceki şemadan elde edilmiştir ya da çıkış yoluyla önceki iki şemadan elde edilmiştir.

Aşağıdaki örnek bir türetme örneğidir. Bu türetme $(P \supset P)$nin LS nin bir teoremi olduğunu belirler. Her sıranın yanında, yazılan sıraya göre kuralı belirtiyoruz.

(1) $(P \supset (Q \supset P))$	A1
(2) $(P \supset ((P \supset P) \supset P))$	(1) Q nun yerine $(P \supset P)$ koyma
(3) $((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$	A2
(4) $((P \supset ((P \supset P) \supset P)) \supset ((P \supset (P \supset P)) \supset (P \supset P)))$	(3) Q yerine $(P \supset P)$ ve R yerine P koyma
(5) $((P \supset (P \supset P)) \supset (P \supset P))$	(2) ve (4) den çıkış
(6) $(P \supset (P \supset P))$	(1) Q yerine P koyma
(7) $(P \supset P)$	(5) ve (6) dan çıkış

Şimdi, aslında sistem tarafından kodlanan şemalar olarak teoremleri tanımlamaktayız.

Tanım: Bir LS teoremi LS deki bazı türetmelerin son şemasıdır.

Bu yüzden LS nin aksiyomları teoremlerdir çünkü her biri LS deki tek-sıralı türetmenin son sırasını oluşturur. Ayrıca teoremler olarak (2),(4),(5),(6) ve (7) numaralı sıraların şemalarına sahibiz.

Herhangi bir lojistik sisteme yerleştirilecek iki apaçık yeterlik koşulu vardır. İlki, sağlam olmalıdır. Yani, kodlanan bütün cümleler mantıksal olarak doğru olmalıdır. LS deki bütün teoremler mantık yasaları olmalıdır. LS nin sağlamlığını kanıtlamak için aksiyomların tümünün, mantık yasaları olduğunu göstermek ve türetme ve çıkış kurallarını sadece ve sadece mantık yasalarından oluşturmaya yarayacağını göstermek yeterlidir. İkincisi, lojistik sistemin tamamlanmış olması, yani mantıksal olarak doğru olan tüm cümleleri kodlaması istenir. LS de her mantık yasasının bir teorem olması istenir. Post [16] tarafından geliştirilen metotları kullanarak LS in sağlam ve tamamlanmış olduğu gösterilebilir. (LS in tamamlanmışlığı için daha kolay olan bir yaklaşım L sisteminin ([14],s.31) tüm teoremlerinin LS siteminin teoremleri olduğunu gösterir ve o zaman Kalmar'ın fikrini kullanır ([14],s.36).

Eğer bir lojistik sistem sağlam değilse o zaman teoremleri arasındaki cümleler doğru olmayan cümlelerdir. Böylece sağlam olmayan bir lojistik sistem terkedilmiş (ya da değiştirilmiş) olacaktır. Ancak tamamlanmamış bir lojistik sistem yine de değerlidir. Aslında birçok lojistik sistem özü itibariyle tamamlanmamıştır ama yine de -yalnızca tümünü kodlamakla kalmayıp – mantıksal olarak doğru cümleleri kodlama amacına hizmet ederler.

Bir lojistik sistem özü itibariyle nasıl tamamlanmamış olabilir? Türetilebilir olarak görülmeyen mantık kurallarını yeni aksiyomlar olarak eklemeye devam edebileceğimiz görülecektir. Bu kesinlikle doğrudur. Fakat sınırlı sayıda yeni aksiyomları ekledikten sonra tamamlanmışlığı elde edebileceğimizin garantisi yoktur. Aslında bir dilin mantıksal doğru önermeleri o kadar karmaşık olabilir ki sistemi tamamlayacak sonsuz sayıda aksiyomu belirtmenin bile etkin bir yöntemi yoktur.

Sınırlayıcı bir durum olarak, tüm mantıksal yasaları başlangıçta aksiyomlar olarak almayı engelleyen nedir? Bu öneri, genel itibariyle, kendisinin imkânsıza indirgemesidir. Yalnızca "haydi mantıksal doğruların böylece kodlanmasına izin verelim" gibi bir büyüyü dile getirerek kendimizi dilin mantıksal olarak doğru önermelerini kodlamış sayabilecek miydik?

Bir sistemi bir kodlama olarak saymak için aksiyomlar, söz konusu verilen herhangi bir cümlenin (ya da şemanın) bir aksiyom olup olmadığına karar vermemizi sağlayacak bir şekilde verilmiş olmalıdır. Dahası örnek olarak verilen böyle cümlelerin (yahut şemaların) verilen sınırlı herhangi bir listesi mantık yasası olarak adlandırılır. Kısacası, cümlelerin (ya da şemaların) sınırlı herhangi bir listesinin bir türetme olup olmadığına karar verebilmemiz gerekir.

Aşağıda gösterildiği gibi üçüncü bir koşulu formüle edebiliriz: bir lojistik sistemin türetmeleri karar verilebilir olmalıdır. Yani aksiyomlar ve işlemler karar verilebilir olmalıdır. Bu koşul sağlanmazsa, bir sistemi bir kodlama olarak saymak için hiçbir sebebimiz yoktur. Türetmeleri belirlenebilir olan bir sistemi etkili bir sitem olarak adlandıracağız.

(Parantez içinde belirtmeliyiz ki aksiyomlar-artı-kurallar formuna sahip olmayan mantıksal doğruların kodlamaları olan lojistik sistemler vardır. Bu yüzden yukarıda formüle edilen "etkili" kavramı üzerinde değişiklik yapılması zorunludur- fakat böyle bazı koşullar kodlama olduğu iddia edilen herhangi bir sistem tarafından yerine getirilmelidir.)

Post ([16], s. 173) tarafından formüle edilmiş olan lojistik sistemler için dördüncü bir yeterlik koşulu vardır. Tamamlanmış ve sağlam bir lojistik sistem düşünelim, ona S diyelim. Eğer yeni bir Φ aksiyomu, yeni bir S′ sistemi yapan S ye eklenseydi şu iki şeyden biri muhakkak olurdu. Birincisi, eğer Φ mantıksal olarak doğru ise, o zaten S teoremi olmuş olurdu, bu yüzden S′ tamamlanmış ve sağlam fakat gereksiz bir aksiyoma sahip olurdu. İkincisi, eğer Φ mantıksal olarak doğru değil ise o zaman S′ sağlam olmayandır – onun teoremlerinden biri (yani Φ) mantıksal olarak doğru değildir. S'nin üretim gücü yeterince verimli olsaydı o zaman bir tane geçersiz teorem bilhassa (LS örneğindeki gibi) hem P hem de ~P hususunda tüm cümlelerin üretilmesine izin vererek sistemi yıkacaktı. Eğer bu olsa, diyebilirdik ki; S sistemi o kadar tamamlanmış ki yeni aksiyomlar (uygun bir şekilde) eklenemez. Bu Post tamamlanmışlık fikridir. Post ayrıca, LS'nin Post tamamlanmış olduğunun gösterilebileceği yöntemler geliştirmiştir. LS'nin neden Post tamamlanmış olduğu fikrini anlamak için yeni aksiyom olarak P'yi eklemeyi düşünelim – (~P'yi de içeren) her şema hemen bir teorem olur. Aynı şey (Q ⊃ P) ekleyerek de sonuç verir: herhangi bir aksiyomu yerine koyarak, diyelim ki A1, (Q ⊃ P) deki Q için (A1 ⊃ P) yi elde ederiz ve böylece A1 den ve (A1 ⊃ P) den çıkış yoluyla P yi elde ederiz. P'den tüm şemalar yerine koyma yoluyla elde edilebilir. LS için, her ne zaman uygun herhangi bir aksiyom eklense, yani hali hazırda teorem olmayan bir şemayı aksiyom olarak eklesek bu şekilde olur. LS için Post tamamlanmış olmanın ifade ettiği şey budur.

Lojistik sistemler için yeterlik koşulları aşağıdaki gibi özetlenebilir: Her lojistik sistem etkin ve sağlam olmalıdır, yani bir kodlama olmalı ve sadece mantıksal olarak doğru cümleleri kodlamalıdır. "İdeal" bir lojistik sistem ayrıca tamamlanmıştır (tüm mantıksal olarak doğru cümleleri kodlar) ve Post tamamlanmıştır (öyle tamamlanmıştır ki uygun bir şekilde yeni herhangi bir aksiyom eklenemez). Dikkat edelim ki Post tamamlanmışlık istemek mantıksız olmamasına rağmen onun yokluğu lojistik sistemde bir yetersizlik değildir fakat tamamlanmışlık lojistik sistemin açıkça istenen bir özelliğidir.

2. Geçerli Argümanlar: Gerektirmeci Sistemler

"Gelişiminin çoğunu Aristoteles'in kıyasından günümüzün matematiksel mantığına doğru geçiren mantığın esas problemi şudur: Verilen herhangi Q, P_1, P_2,..., önermeleri, tam olarak Q önermesinin P_1, P_2, ... önermelerinin mantıksal bir gerektirmesi olduğu koşullar altında yer alır."

PAUL GILMORE

Mantıksal teorinin ikinci ve daha kapsamlı bir amacı da bütün geçerli argümanların kodlanmasıdır. Bu amaç geçerli argümanların sadece hukuktaki, felsefedeki, bilimdeki vs. değerini fark etmesi gereken lojistikçiler için değil aynı zamanda diğer tüm mantıkçılar için de akla yatkındır. Bir argüman, biri sonuç geri kalanı öncül önermelerinden oluşan bir sistemdir. Herhangi bir argümanı, S'nin bir önermeler kümesi olduğu ve p'nin bir önerme olduğu S ∴ p şeklinde temsil edebiliriz. Öncüllerin doğruluğu, sonuç önermesinin doğruluğunu sağlarsa veya diğer bir deyişle, öncüllerin tümü doğru iken sonuç önermesinin yanlış olması imkânsızsa S ∴ p argümanı geçerlidir. Yorumlar açısından bir S ∴ p argümanını, ancak ve ancak S'deki tüm önermeleri doğrulayan her bir yorum p'yi de doğrularsa geçerli olarak tanımlayabiliriz. (Tarski örneği aşağıdadır [19], syf. 309-420). Ancak ve ancak S'deki tüm önermeleri doğrulayan ve ayrıca p'yi yanlışlayan bir yorum yoksa S ∴ p'nin geçerliliği sürer.

Yaygın olarak geçerli bir argümanın sonucunun öncüllerin mantıksal sonucu olduğu söylenir. Bu sebeple gerektirmeci sistem terimi, sadece mantıksal olarak doğru cümlelerin değil, aynı zamanda geçerli argümanların da kodlamasını amaçlayan mantıksal sistemi kasteder. Bazı mantıkçılar ([13], s. 2; [12], s. v), mantıksal gerektirmelerin araştırmasını mantığın esas ilgi alanı olarak sayarlar. Bu referanslarda verilen sistemler, gerektirmeci sistemlerin örnekleridir.

Bir gerektirmeci sistem, dilin mantıksal olarak doğru cümlelerini, bir lojistik sistemle aynı tarzda kodlar. Geçerli argümanlar, *verilen bir önerme kümesinin mantıksal gerektirmeleri, verilen kümeden üretilir* çıkarımının kuralları ile kodlanır. Tabii ki çıkarımın kuralları daima, sadece verilen önerme kümesinin mantıksal gerektirmelerinden üretilmelidir. Bu yüzden yerine koyma kullanılamaz. Ayırma, mantıksal sonucu oluşturan üç cümlenin ikisinden üretir.

Bir örnek olarak, (1) LS tarafından kodlananla aynı olan L'nin mantıksal olarak doğru cümlelerini ve (2) geçerliliği yalnızca ⊃ ve ~ anlamlarını içeren L'nin geçerli argümanlarını kodlayacak olan bir CS gerektirmeci sistemini vereceğiz. Sonraki (2 olarak verilen)*, geçerli argüman formlarının kodlanmasıyla yapılacaktır. Bir argüman formu, biri sonuç ve diğerleri öncüller olmak üzere bir şema sistemidir. Σ bir şemalar kümesi olduğu ve Φ bir şema olduğu Σ ∴ Φ olarak bir argüman formu tasvir edebiliriz. Bir argüman formu, örneklerinin hepsi geçerli argümanlar ise geçerlidir.

Aşağıdaki 5 tanım CS gerektirmeci sistemini tanımlamaktadır.

Tanım: CS'nin aksiyomları, aşağıdaki formların bütün şemalarıdır.

(B1) $(\Phi \supset (\delta \supset \Phi))$

(B2) $((\Phi \supset (\delta \supset \eta)) \supset ((\Phi \supset \delta) \supset (\Phi \supset \eta)))$

(B3) $((\sim\Phi \supset \sim\delta) \supset ((\sim\Phi \supset \delta) \supset \Phi))$

Bu yüzden, LS'nin sadece üç aksiyomu vardı fakat CS, sonsuz sayıda aksiyoma sahiptir. Bu tanım, yerine koymanın sağlam bölümünü birleştirmeyi ve sağlam olmayan kısmından kaçınmayı sağlar. Bizim tek üretme işlemimiz yukarıda tanımlandığı üzere ayırmadır.

Tanım: (yukarıda geçen, ayırma tanımı)

Gerektirmeci sistemdeki bir türetme, öncüllerden bir sonuç üretmenin bir kaydıdır.

Türetmeler, bir türetmeyi kurarken kullanılan esasların zorunlu olarak standart ya da sıkı olmaması durumu haricinde öncüllerden çıkan standart ve sıkı kanıtlarla örtüşür.

Tanım: Σ den Φ türetmesi, ya Σ deki bir önermeyle ya da bir aksiyomla başlayan (1); ve sonraki her bir şemanın, ya Σ deki bir öncül, ya bir aksiyom, ya da ayırma yoluyla dizideki önceki iki şemadan elde edilmiş olduğu (2); ve Φ ile biten (3) şemalarının sınırlı bir sırasıdır.

Mantıksal yasalar, öncüllerin boş kümesinden türetmeler yoluyla kodlanır. Bu tam olarak, sadece boş kümeden türetmelerin olduğu lojistik sistem haricinde mantıksal yasaların bu sistemde kodlanması metodudur.

Tanım: Bir CS teoremi, boş küme olan ʌ den türetmenin son şemasıdır.

Geçerli argümanlar türetilebilen argüman formlarınca kodlanır.

Tanım: Ancak ve ancak Σ den Φ türetmesi var ise, CS de bir argüman formu olan Σ ∴ Φ türetilebilirdir.

Farkına varalım ki CS, sonsuz sayıda öncülleri olan argümanları kodlar. Herhangi bir gerektirmeci sistem, bir kapsamlılık iddiası için bunu yapmalıdır çünkü sonsuz sayıda aksiyomu ve bir gerektirmeci sistemi olan çok sayıda bilim vardır ve bunlar bilimdeki geçerli argümanları kodlayabilmelidirler. Daha genel olarak, sonsuz sayıda önermenin aslında mantıksal gerektirmeleri olduğu için bir gerektirmeci sistem, sonsuz sayıda öncülleri olan argümanları kodlayabilmelidir.

Genel tanımlarımızın bir sonucu olarak, hiç öncülü olmayan argüman formları vardır. ∧ ∴ Φ formunda, türetilebilir argüman formları vardır. Bu, sistemin bir yapaylığı gibi görünebilir. Fakat tanım vasıtasıyla, ancak ve ancak ∧ ∴ Φ türetilebilir ise Φ nın bir teorem olduğu fikrine sahibiz. Bu, Leibniz'in, mantıksal olarak doğru bir önermenin, başka hiçbir ilke kullanmadan, yalnızca mantığa dayanarak türetilebilir olan önerme olduğu gözlemi ile uyum sağlar.

Aşağıdakiler birkaç türetme örneğidir. Türetilen argüman formu ile birlikte türetmeyi kullanacağız ve sağında da kullanılan kuralı vereceğiz.

$\wedge \therefore (P \supset P)$

(1) $(P \supset ((P \supset P) \supset P))$	(B1) formundaki şema
(2) $((P \supset ((P \supset P) \supset P)) \supset ((P \supset (P \supset P)) \supset (P \supset P)))$	(B2) formundaki şema
(3) $((P \supset (P \supset P)) \supset (P \supset P))$	(1) (2) çıkış
(4) $(P \supset (P \supset P))$	(B1) formundaki şema
(5) $(P \supset P)$	(3) (4) çıkış

$[P] \therefore P$

(1) P	(P) de öncül

$[P, (P \supset Q), (Q \supset R)] \therefore R$

(1) P	$[P, (P \supset Q), (Q \supset R)]$ de öncül
(2) $(P \supset Q)$	$[P, (P \supset Q), (Q \supset R)]$ de öncül
(3) $(Q \supset R)$	$[P, (P \supset Q), (Q \supset R)]$ de öncül

(4) Q (1), (2) çıkış

(5) R (3), (4) çıkış

Dilin sadece mantıksal doğru önermelerinin tümünü kodlama amacına (1) ve dilin sadece geçerli argümanlarının tümünü kodlamaya (2) yeterli bir şekilde olanak tanımak için bir gerektirmeci sistem dört koşulu karşılamalıdır. Birinci ve ikincisinde, adı geçen duyularda tamamlanmış ve sağlam olmalıdır. Üçüncüsü, güçlü bir şekilde tamamlanmış olmalıdır, yani bütün geçerli argümanları kodlamalıdır. Sonuncusu, güçlü bir şekilde sağlam olmalıdır, yani sadece geçerli argümanları kodlamalıdır.

CS nin bilindiği kadarıyla, LS için tamamlanmışlık ve sağlamlık tartışması ufak değişikliklerle devam etmektedir. Güçlü geçerlilik kurmak için, tüm aksiyomların mantıksal yasalar olduğunu ve ayırmanın daima mantıksal bir gerektirme olan üç şemanın ikisinden üretildiğini göstermek gereklidir. CS için güçlü tamamlanmışlığın en basit kanıtı muhtemelen Henkin'in fikirlerinden [6] biridir.

Sanki uygun yeni bir aksiyomun eklenmesi sistemi kullanılmaz hale getirecekmiş gibi, Post tamamlanmışlık, bir gerektirmeci sistem için açıkça uygun bir yeterlik koşulu değildir. O zaman sistem, güçlü bir şekilde sağlam olma konusunda başarısız olurdu. S de, ~P nin, P den türetilebilir olmasını bekleyemeyiz.

Bir gerektirmeci sistemin etkili olması gerektiğini (makul bir düşüncede) söylemeden olmaz. Diğer yandan, ona herhangi bir şeyi kodladığı nazarıyla bakamayız.

Hem lojistik sistemler hem de gerektirmeci sistemler, gördüğümüz üzere, bir tür türetmeye sahiptir. Bu iki sistemin arasındaki temel fark kendi şahsi türetmelerinin doğasında yatar. Gerektirmeci sistemdeki türetmeler, öncüllerin seçmeli kümelerinden olan türetmeler iken, lojistik sistemin türetmeleri sistemin aksiyomlarına dayanmak zorundadır. Gerektirmeci sisteminin, mantıksal doğruları olduğu kadar geçerli argümanları da kodlayabilmesini sağlayan bu farktır. Diğer yandan lojistik sistem yalnızca mantıksal doğruları kodlama görevinde uygundur.

Geçerli argümanların kodlaması olarak lojistik sistemleri kullanma girişimleri olmuştur (bakınız [11], syf. 341 ya da [7], syf. 23). Eğer bu, varsayımlardan türetmeleri sunarak yapılmak zorunda olsaydı, o zaman tabii ki sistemin bir gerektirmeci sisteme dönüştürüldüğünü söylemek akla uygun olurdu. Varsayımlardan türetmeleri sunmadan, lojistik sistemlerin nasıl geçerli argümanların kodlaması olarak görülebileceğini düşünmek adına kısa bir parantez açmama izin verin.

Örneğin LS de, bir [Φ] ∴ δ öncülü ile argüman formlarını kodlamak için ($\Phi \supset \delta$) formunun teoremlerini kullanabilirdik. Fakat bu yetersiz olur çünkü o sadece tek öncüllü argümanları kodlar. Bu sorun, geçerli argümanlardan mantıksal olarak doğru önermelere kadar ilgili olan eş şartlı (corresponding conditional) ilkesini kullanarak kısmen önlenebilir. Bu ilkeyi düzgünce belirtmek için, doğruluk fonksiyonu anlamının içinde & de içermesi haricinde L gibi bir dili varsaymamız gerekir. Ayrıca ⊃, ~ ya da & bağlı olan L' nin tüm mantıksal doğrularını o kodlaması için LS nin dönüştürüldüğünü varsayalım. Örneğin; ((P & Q) ⊃ P), LS olarak adlandıracağımız bu yeni lojistik sistemin bir teoremi olacaktı.

Eş şartlı ilkesi, boş olmayan ve öncüllerin sınırlı bir kümesine sahip olan bir argümanın geçerliliği ile önbileşeni, öncüllerinin tümel-evetlemesi ve artbileşeni ise sonucu olan koşullu önermenin mantıksal doğruluğu ile ilişki kurar. Daha açık olarak:

Ancak ve ancak ((Φ_1 & Φ_2 & … & Φ_n) ⊃ δ) mantıksal olarak doğru ise [Φ_1, Φ_2, …, Φ_n] ∴ δ geçerlidir.

Bu yüzden, argümanlar ile eş şartlı tarafından kodlanan önermelerin boş olmayan sınırlı kümelerini alabiliriz. Dahası, öncülleri olmayan argümanlar sonuçları tarafından kodlanabilir. Çünkü:

Ancak ve ancak δ mantıksal olarak doğruysa ∧ ∴ δ geçerlidir.

Bu sebeple, bütün sonlu argümanlar bir lojistik sistemin içerisinde kodlanır. Sonsuz bir argümanı kodlamak için, önbileşeni öncüllerinin bazısının tümel –evetlemesi olan ve artbileşeni sonucu olan koşulluyu kullanabiliriz. Örneğin; H tüm şematik harflerin kümesi olsaydı, o zaman H ∴ P geçerli argüman formu (P ⊃ P) tarafından kodlanacaktı. Çünkü [P], H'nin alt kümesidir.

Bu düzenleme, şu koşullarda yeterli olmayacaktı: (1) sonucu içeren öncüllerin sınırsız bir kümesinin olduğu koşulda (2) öncüllerin, aynı sonucu içeren hiçbir sınırlı alt kümesinin olmadığı koşulda. Ama türetmeler yalnızca sınırlı uzunlukta olduğu için gerektirmeci sistem böyle bir argümanı kodlayamayacaktı.

Dolayısıyla bir lojistik sistem, mantıksal doğruları olduğu kadar geçerli argümanları da kodlamak için kullanılabilir. Fakat bu, bir turtayı kalemle kesmek gibidir – alet görev için uygun değildir ama kişi karmakarışık bir iş olmasını önemsemiyorsa kullanılabilirdi. Genel olarak, eş şartlının türetmelerini anlamak, öncüllerden çıkan sonucun türetmesini anlamaktan daha uzun sürecek ve daha zor olacaktır. Dahası, bir gerektirmeci sistemde güzel bir şekilde gerçekleştirilebilen ama bir lojistik sistemde neredeyse anlaması imkânsız olan geçerli argümanlar arasında birçok ilginç ilişki vardır. Bu konuda bir kişi, bir gerektirmeci sistemdeki çıkarım teoreminin gelişimini, onun benzer lojistik sistemdeki gelişimi ile kıyaslayabilir.

Muhtemelen, göreve uymayan bir sistemdeki geçerli argümanları ele almaya çalışmanın en sıkıntılı zorluğu kavramsal bir argüman olmasıdır. $((\Phi_1 \,\&\, \Phi_2, \ldots, \&\, \Phi_n) \supset \delta)$ olarak kodlanan bir $[\Phi_1, \Phi_2, \ldots, \Phi_n]$ argümanını alırsak o zaman "$\supset$" ile "içerir" kavramlarını birbirine karıştırma eğilimi vardır. Matematikçilerin, "$\supset$" i ifade etmek için kısa bir kelimeye ihtiyaç duyarak "içerir" kelimesini kullandıkları sosyal bir gerçektir. Bu gerçek, kafa karışıklığını çözmeye hiçbir şey katmaz ama aslında onu sağlama konusunda yetki verdiği görülür. Kişi lojistik sistem üzerinde çalışıyorsa karışıklığı çözen apaçık gerçek görmezden gelinmeye meyillidir (örneğin; "içerir" olmaması gerekirken $\supset$, L dedir).

Özetle gerektirmeci sistemler mantıksal doğruluğu olduğu kadar mantıksal gerektirmeyi de işlerken lojistik sistemlerin doğal olarak sadece mantıksal doğrulukla ilgilendiği düşünülür. Gerektirmeci sistemler, problemlerin daha kapsamlı bir bölümünü işlediği için, bir gerektirmeci sistem için yeterlik koşulları lojistik sistemdekilerden daha kapsamlıdır. Sağlam olmanın yanında, bir gerektirmeci sistem güçlü bir şekilde sağlam da olmak zorundadır. Bir gerektirmeci sistem için sadece tamamlanmışlık değil, güçlü tamamlanmışlık da istenilir. Üstelik bir lojistik sistemden Post tamamlanmış olmasını beklemek mantıklı olmamasına rağmen yeterli bir gerektirmeci sistem bu özelliğe sahip olabilir çünkü o güçlü bir şekilde sağlamlık ile çelişir. Son olarak geçerli argümanların, öncüllerden türetmeleri olmayan bir sistemde kodlanabilmesine rağmen, kodlamanın detayları gereksiz bir şekilde karmaşık, verimsiz ve aydınlatıcı olmanın da zıttıdır.

3. İspatlar: Dedüktif Sistemler

> *İspatlar…, bir ispatın yapısı mantıksal olarak analiz edilmeden önce var olmak zorundaydı; ve bu analiz matematiksel eserlerin geniş alanına dayanmış olmalıydı. Başka bir deyişle mantık, -biz matematikçilerin bildiği kadarıyla- kullandığımız dilin, gramer oluşturulmadan önce var olmak zorunda olan dilin, gramer yapısından ne eksik ne de fazladır.*
>
> BOURBAKI

Mantıksal olarak doğru cümlelerin ve geçerli argümanların kodlanmasına ek olarak bir mantıkçı, dilin ispatlarını kodlamak ile ilgilenebilir. Başka bir deyişle bir kişi, öncüllerden çıkan bir sonucu göstermeyi içeren akıl yürütmenin adımlarını incelemek isteyebilir. Bir kişi, bir dilin ispatlarının ne olduğuna dair fikir almak için dedüktif akıl yürütmenin (geometri gibi) var olan birkaç külliyatında olanların bazılarını inceleyebilir.

Birtakım ispatları inceleyerek bir kişi; ispatın, çıkarımın kuralları olarak adlandırılabilen belli kurallara göre oluşturulmuş sınırlı sayıdaki söylemler olduğu sonucunu çıkarmaya neredeyse mecbur olur. Bu sebeple bizler, çıkarımın kurallarını ve nasıl uygulandığını belirterek dilin ispatlarını kodlayabiliriz.

Bizler mantıkta, meydana gelen bütün kanıtları kodlamayı amaçlamayız fakat iki yönde idealize ederiz. İlk etapta hataları kodlamayı istemeyiz, bu yüzden çıkarımın kurallarının sağlam olmasını, yani öncüllerden aslında takip ettiği sonuçlara varmalarını istemeyiz. İkinci olarak, mantıksal sezginin büyük sıçramalarını kodlamak istemeyiz, kurallarımızın mutlaka basit olmasını isteriz, yani olası en fazla mantıksal detaya sahip ispatları kodlamak isteriz. Kısacası mantıksal olarak sıkı ispatları kodlamak isteriz.

Örnek olarak; L'nin ispatlarının kodlanması problemini ve yalnızca $\supset$ ve ~ anlamlarını çevirenleri düşüneceğiz. Onları: örneklerinin tümü, sadece $\supset$ ve ~ anlamlarını içeren kanıtlar olan, söylem şemaları olan ispat formlarını kodlayarak kodlayacağız. CS nin türetmeleri, ispat formları için makul bir adaydır. Fakat sağlam olsalar da bunları sıkı ispat formları olarak almak için apaçık bir engel vardır.

Aksiyomların yazımı (özellikle uzun ve anlaşılması zor olanlar) aslında ne normal akıl yürütmede bulunan bir ilkedir ne de sıkıdır.

Aksiyomların CS deki gibi yazımları sıkı değildir çünkü mantık çerçevesinde daha mantıksal bir ayrıntı önerilebilir ve kolaylıkla sağlanabilir. Bu yüzden, böyle bir aksiyomu aktaran bir ispat mantıksal boşluk (logical gap) içerecektir. Öncüllerden türetmeler, sağlam olmalarına ve bir kıyasın sağlam olması için gereken bir şey olmalarına ve onun için mantıksal olarak basit olmanın oldukça başka bir şey olduğu gerçeğine rağmen reddedilirler. Bir kişi, doğrudan aksiyomlardan sağlam bir şekilde çok karmaşık bir geometri teoremi çıkarabilir ama bir kişi de bunu aksiyomlardan çıkan teoremlerin ispatı olarak görmeyebilir.

(a) Akıl Yürütmenin İlkeleri

Aşağıda, ilke türlerinin aslında normal akıl yürütmede bulunduğu düşüncesini inceleyeceğiz. Amacımız, kesinlikle sıkı olmayan doğal akıl yürütmeyi ya da normal akıl yürütmeyi tanımlamak değildir. Ancak aşağıda göreceğimiz üzere, normal akıl yürütmede sık sık kullanılan sıkı ilkeler vardır. Bu ilkelerin bazılarını, sadece mantıksal doğruları ve geçerli argümanları değil hem de iyi bir gerekçe ile sıkı olarak adlandırılabilecek ispatları kodlayan bir dedüktif sistem kurmak için kullanacağız.

Geometrinin (veya dedüktif akıl yürütmenin başka herhangi bir kısmının) kanıtlarını incelemede, çıkarımın birçok türünü buluruz; bazıları sağlam ve bazıları -daha olası olarak- sağlam olmayan, bazıları boşluksuz (gapless) ve bazıları da ek olarak birkaç mantıksal ayrıntı doldurulmuş olanlarına katlanabilenlerdir. İspatın, boşluksuz yahut sıkı ilkelerinin üç türünü bulabiliriz. İlkenin fark edilen ilk türü muhtemelen dolaysız çıkarım kuralları olacaktır. Bir dolaysız çıkarım kuralı, önceki bir ya da iki sıraya dayanılarak yeni bir sıranın yazımına izin veren kuraldır. Ayırma bu türdendir. Diğeri, örneğin (P & Q) dan P ayırma ilkesidir. Aşağıda önemi bahsedilecek olan dolaysız çıkarım kuralı tekrar olarak adlandırılır. Bu kural, kişinin önceki sırayı tekrar etmesine izin verir. (Burada dolaysız kelimesini apaçık anlamında kullanmadığımıza dikkatinizi çekerim. Burada bahsedilen dolaysız olmayan çıkarımların birçoğu aynı zamanda apaçıktır.)

Böyle bir araştırmada dikkat edilen sonraki ilke, ispattaki bir sıra olarak, bir sonuç çizmeyi istediğimiz varsayımlardan birine giriş yapmaya olanak veren bu kuraldır – eğer ki bunun (üstündeki adımlara dayanan bir sonuç değil) bir varsayım olduğunu açıkça belirtirsek. Bu kurala çıkarım deriz ve onu kendi başına bir sınıfa koyarız.

Aksiyomların girişine olanak veren kural haricinde, gerektirmeci bir sistemin genellikle yalnızca dolaysız çıkarım kurallarına ve varsayım kuralına sahip olduğuna dikkat ediniz. Mantıksal aksiyomların girişi genellikle ispatta boşluklar yarattığı için bu türden ilkeleri sıkı olarak değerlendirmiyoruz. Dikkat edilen üçüncü kural, sadece (normal akıl yürütmede asla görülmeyen) CS'nin karmaşık aksiyomlarının sıkı ispatlarını veren değil; aynı zamanda normal akıl yürütmede bulunan (P v ~P) ve ~(P & ~P) gibi basit aksiyomların sıkı ispatlarını veren bir anahtar sağlar. İspatın sıkı ilkesinin üçüncü tipi normal akıl yürütmede yaygın olarak bulunur ama genellikle gerektirmeci sistemlerde bulunmaz. Gerektirmeci sistemlerdeki türetmeleri doğal olmayan ve sıkı olmayan hale getiren üçüncü tip kuralın yokluğu sağlanabilir.

İspatın üçüncü tip ilkesini yapısal kural olarak adlandırırız. Yapısal kuralın ayırt edici özelliği, bir ispattaki çıkarımın önceki sıralardan değil, akıl yürütme modeline dayanılarak yapılmasına olanak vermesidir. Örneğin; aşağıdaki (boş kümeden) ((P & Q) ⊃ (Q & P)) ispatını düşünün:

Farz edelim ki

	(1) (P & Q).	(1)den
	(2) Q.	Yine (1)den
	(3) P.	(2) ve (3)ten
	(4) (Q & P).	
Böylece	(5) ((P & Q) ⊃ (Q & P)).	

(2), (3) ve (4) sıralarının tümünün, sıra (1) e dayandığına ve tümünün önceki satırlardan olan dolaysız çıkarımlar olduğuna dikkat ediniz. Ancak sıra (5) sıra (1)e, ne başka bir sıraya ne de yukarıdaki diğer sıralara dayanmaz. Sıra (5), akıl yürütmenin modeline dayanarak çıkarılmıştır. (P & Q) dan (Q & P) ye olan akıl yürütmeye dayanarak ve önceki herhangi bir sıraya dayanmadan ((P & Q) ⊃ (Q & P)) çıkarımını yaptık.

Genel olarak şartlılık ilkesi, Φ den δ çıkarımına dayanarak ve Φ'a dayanmayıp belki de yalnızca akıl yürütmedeki önceki diğer varsayımlara dayanarak yapılan (Φ ⊃ δ) çıkarımına olanak sağlar. Dolaylı ispatı içeren iki yapısal kural vardır. İlki değillemeye giriş olarak adlandırılır: Φ den δ ve Φ den ~δ akıl yürütmesine dayanarak ~Φ sonucunu çıkarırız. İkinci değillemeden çıkış: ~ Φ'den δ'ye ve ~Φ'den

~δ'ye akıl yürütmeye dayanarak Φ sonucunu çıkarırız. Bunlar indirgeme "reductio" tipi ispatlarında ardındaki ilkelerdir.

Şu kuralları kullanarak LS'nin üç aksiyomunun boşluksuz ispatlarını (gapless proofs) vereceğiz: varsayım (A), tekrar (R), ayırma veya ⊃ ayırmadan çıkış (⊃E), koşullu veya ⊃ koşulluya giriş (⊃*I*), değillemeden çıkış (~E) ve değillemeye giriş (~*I*). Varsayımları göstermek için (+) artı işareti kullanacağız ve akıl yürütme modelini kutuya koyacağız. Basitlik için, her akıl yürütme modelinin sadece bir tane varsayım içermesi ve varsayımın modeldeki ilk şema olması gerekir.

(A1)	(1)	$+P$	A
	(2)	$+Q$	A
	(3)	P	R
	(4)	$(Q \supset P)$	$\supset I$
	(5)	$(P \supset (Q \supset P))$	$\supset I$

Bu ispatın fikri şöyledir: (P ⊃ (Q ⊃ P)) yi kanıtlamak istediğimiz için, P yi ve sonrasında (Q ⊃ P) akıl yürütmesini varsaymayı deneyebiliriz. Böylece P (satır 1) yi varsayarız. Varsayımlarımızdan şimdi de (Q ⊃ P) yi ispatlamak istediğimiz için Q ve P ye akıl yürütmesini varsaymayı deneyebiliriz. Böylece Q (satır 2) varsayarız. P yi, P (satır 3) varsayımımızın tekrarı takip eder. Böylece şu an elimizde Q dan P ye akıl yürütme modeli vardır – bu yüzden onu kutuya alır ve (Q ⊃ P) (satır 4) sonucunu çıkarırız. Şimdi de elimizde P den (Q ⊃ P) ye akıl yürütme modeli olur – böylece onu kutuya alır ve sonucu (satır 5) yazarız.

Dikkat ediniz ki, bir kutu bir varsayımı barındırabilir dediğimizde, sadece bir tane kutuya koyulmamış varsayımı kastettik –bazı akıl yürütmeler, bir kez içindeki bireysel şemalardan çıkarıldığında artık ispatın bireysel adımları olarak fonksiyonları yoktur - çıkarımlar, artık tekil şemalardan yapılmaz çünkü onlar artık bir akıl yürütme modelinin parçasıdır.

(A2)

$+(P \supset (Q \supset R))$	A
$+(P \supset Q)$	A
$+P$	A
Q	$\supset E$
$Q \supset R$	$\supset E$
R	$\supset E$
$(P \supset R)$	$\supset I$
$((P \supset Q) \supset (P \supset R))$	$\supset I$
$((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$	$\supset I$

(A3)

$+(\sim P \supset \sim Q)$	A
$+(\sim P \supset Q)$	A
$+ \sim P$	A
Q	$\supset EE$
$+ \sim P$	A
$\sim Q$	$\supset E$
P	$\sim EE$
$((\sim P \supset Q) \supset P)$	$\supset I$
$((\sim P \supset \sim Q) \supset ((\sim P \supset Q) \supset P))$	$\supset I$

Bu ispatlar maksimum miktarda mantıksal detay içerir. Onlar kesinlikle sıkıdır. İspatta yeni bir sıra her yazıldığında, talep edilen ve (arda kalan olmadan) sağlanan hiçbir ek mantıksal detay yoktu. Bu kadar bariz bir şekilde ölçüsüz iddiaların yapılabiliyor olma sebebi kabaca: kanıtların, yeni bir sıra eklenmesi ile her uzayışı ya bir varsayım ya tam olarak bir mantıksal sembole girişi veya çıkışı içerir.

(b) Dedüktif Sistemler

Dedüktif sistem terimini, mantıksal doğruları, geçerli argümanları ve (sağlam, sıkı) ispatları kodlayan bir mantıksal sist emi belirtmek için kullanırız. Dedüktif terimi, böyle bir sistemin en kapsamlı amacının dedüksiyonun kodlanması olduğunu vurgulamak amacıyla seçilmiştir. (yazarın notu: son zamanlardaki dedüktif sistem terimi kısmen yukarıdaki gibi kullanılmıştır. Fakat geçmişte, gerektirmeci bir sisteme ilişkin kendi gerektirmelerini içeren bir cümleler sınıfını gösteren modern teknik bir terim olan 'teori' ile eş anlamlı olarak kullanılıyordu.).

Aşağıda bir dedüktif sistem örneği olarak LS sistemini vereceğiz ki bu sistem; (1) L'nin, LS tarafından kodlanmış olanlarla aynı mantıksal doğrularını kodlar; (2) L'nin CS tarafından kodlananlarla aynı geçerli argümanlarını kodlar; (3) doğruluğu ⊃ ve ~ sembollerini oluşturan L'nin (sağlam ve sıkı) ispatlarını kodlar. Sonra, bütün örnekleri, yukarıdaki türün ispatları olan söylem formları anlamına gelen ispat formlarının (PFs) kodlanması ile tamamlanmış olacak. Yukarıda; (A1), (A2) ve(A3) ispatları olarak bahsedilen formların kendileri L deki ispatlar değillerdir. Çünkü P, Q ve R sembolleri şemalardır. Fakat P, Q ve R L'nin cümlelerine yerleştirildiği zaman sonuçlar, formların örnekleri olurlar ve bu önekler aslında ispatlardır.

İspatları, geçerli argümanları ve mantıksal doğruları kodlayan bir dedüktif sistemi tanımlamak için, bir kişi (1) ispatları tanımlayabilir, (2) S öncüller kümesinden p sonucuna varan ispatları tanımlayabilir, (3) öncüllerinden sonucuna ulaşan bir ispatı olan bir argüman gibi türetilebilir bir argümanı tanımlayabilir ve son olarak (4) $\wedge$ $\therefore$ p nin türetilebilir bir argüman olduğu bir p cümlesi olarak bir teoremi tanımlayabilir.

Bir + işaretinden önceki bir şemayı belirtmek için 'eklenmiş (plussed) şema' terimini, bir kutu içerisine iliştirilmiş bir şemayı belirtmek için 'kutuya alınmış (boxed) şema' terimini alarak, LS yi tanımlayarak aşağıdaki dört tanımı sunuyoruz. PF nin endüktif tanımında, ilkenin adının kısaltmasını kullanarak çıkarımın çeşitli ilkeleri ile örtüşen cümleleri vurgulayacağız.

Tanım: Bir DS dedüktif sisteminin PF si şöyle tanımlanır:

(i) (A)(a) bir eklenmiş şema bir PF dir.

(b) her PF, herhangi bir eklenmiş şema eklenerek başka bir PF ye genişletilebilir.

(ii) (R) her PF, eğer eklenecek şema eklenmiş şema ise, zaten içinde olan artıyı çıkaran kutuda olmayan herhangi bir şema eklenerek başka bir PF ye genişletilebilir.

(iii) ($\supset$ E) kutuda olmayan Φ ve (Φ $\supset$ δ) (eklenmiş şema olsun olmasın) içeren herhangi bir PF, δ eklenerek genişletilebilir.

(iv) (1) eklenmiş şema kutu dışı olan bir şema ile başlayan; (2) başka (kutuya alınmamış şema) eklenmiş şema içermeyen bir dizi ile biten bir PF, eğer kutuya alınmamış ve eklenmemiş (unplussed) şema ile bitiyorsa sonraki sıra kutuya alınarak başka bir PF ye genişletilebilir.

(v) ($\supset$ *I*) (1) +Φ ile başlayan ve (2) δ ile biten kutuya alınmış bir şema ile biten herhangi bir PF, (Φ $\supset$ δ) eklenerek başka bir PF ye genişletilebilir.

(vi) (~*I*) biri +Φ ile başlayıp δ ile biten ve diğeri +Φ ile başlayıp ~δ ile biten iki başarılı kutuya alınmış dizilerde biten herhangi bir PF ~Φ eklenerek genişletilebilir.

(vii) (~E) biri +~Φ ile başlayıp δ ile biten ve diğeri +~Φ ile başlayıp ~δ ile biten iki başarılı kutuya alınmış dizilerde biten herhangi bir PF, Φ eklenerek genişletilebilir.

(viii) sadece yukarıdaki kuralların uygulamalarının sınırlı bir sayısı ile oluşturulabilir olanlar PF lerdir.

Aksiyomlara giriş ile örtüşen hiçbir kural yoktur. Bu mantıksal sistem aksiyom içermez. İlk cümle ((i)a), her PF nin bir varsayım ile başladığını söyler. (iv) cümlesi akıl yürütme modelinin kutuya alınmasının kuralıdır. Kalan diğer cümleler, yeni bir şemanın eklenmesi ile verilen bir ispatın nasıl uzatılacağını belirler. Bu kurallar aşağıda diyagram olarak verilmiştir:

DS'deki İspat Formları Kurallarının Diyagramı

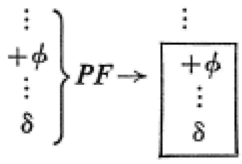
(i) Varsayım: A

$$\text{(a)}\ +\phi \qquad \text{(b)}\ \begin{array}{l} \left.\vdots\right\} PF \\ +\phi \end{array}$$

(ii) Tekrar: R

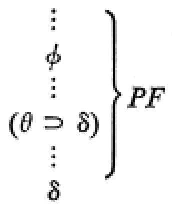
$$\text{(a)}\ \begin{array}{l} \left.\begin{array}{c} \vdots \\ +\phi \\ \vdots \end{array}\right\} PF \\ \phi \end{array} \qquad \text{(b)}\ \begin{array}{l} \left.\begin{array}{c} \vdots \\ \phi \\ \vdots \end{array}\right\} PF \\ \phi \end{array}$$

(iv) Akıl yürütme kalıbını kutudan çıkarma

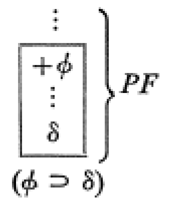


Not: kutudan çıkarılmış ispatın bu kısmı kutuya alınmamış sadece bir varsayım içerebilir (eklenmiş şema)

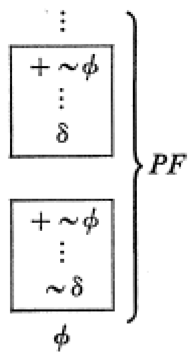
(iii) Ayırma: ⊃ E



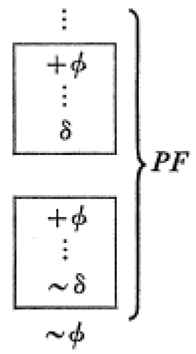
(v)Koşullandırma: ⊃ I



(vii) Çıkarım: ~E



(vi) Çıkarım: ~I



Şimdi, PF nin, Φ nin bir şema ve Σ nin bir şemalar kümesi olduğu Σ den Φ, PF olmasının ne anlama geldğini tanımlayacağız.

Tanım: Bir PF, ancak ve ancak Φ, ɑ nın sırası ise ve Φ kutuya alınmıştır ve ɑ daki her kutuya alınmış eklenmiş şema Σ nın bir üyesidir.

Bir örnek olarak [(P ⊃ (Q ⊃ R)), (P ⊃ Q)] den bir (P ⊃ Q) PF veririz.

$+(P \supset (Q \supset R))$
$+(P \supset Q)$

$$\boxed{\begin{array}{c} +P \\ Q \\ (Q \supset R) \\ R \end{array}}$$

$(P \supset R)$

PF yi yukarıdaki gibi genişleterek, boş bir kümeden ve (A2) den bir PF elde ediyoruz, yani yalnızca mantıksal ilkelere dayanan (A2) nin bir kanıtıyla uyumlu olanı elde ediyoruz. CS de boş bir kümeden (A2) türetmesi sadece bir tek sıra türetmedir:

$((P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R)))$

Bu, akıl yürütmede bir boşluğu özetlemek gibi görünür. Eğer daha çok mantıksal ayrıntı istersek, o zaman DS gibi bir sisteme gitmeliyiz. Çünkü CS de başka boşluklar yaratmadan boşluğu doldurmanın bir yolu yoktur.

Tanım: Ancak ve ancak Σ den Φ ya bir PF varsa Σ ∴ Φ DS de türetilebilir bir argüman formudur.

Tanım: Ancak ve ancak ∧ ∴ Φ DS de türetilebilir bir argüman formu ise Φ DS nin bir teoremidir.

Bu özel dedüktif sistemin ilgilendiği kadarıyla; sağlamlık, tamamlanmışlık, güçlü sağlamlık ve güçlü tamamlanmışlık; CS nin güçlü sağlamlığından ve güçlü tamamlanmışlığından kolaylıkla çıkarılabilir. Tamamlanmışlıktan ve sağlamlıktan sırasıyla güçlü tamamlanmışlık ve güçlü sağlamlık sonucu çıktığı için CS nin özgülüklerinden DS nin bu sonraki özgülüklerini kanıtlamak yeterlidir. Her iki ispat da öğreticidir.

DS nin güçlü bir şekilde tamamlanmış olduğunu göstermek için, her geçerli argüman formunun DS de türetilebilir olduğunu göstermeliyiz. CS nin güçlü tamamlanmışlığından biliyoruz ki her geçerli argüman CS de türetilebilirdir. Bu yüzden CS de türetilebilir her geçerli argüman formunun LS de de geçerli olduğunu göstermek yeterlidir. Σ∴Φ nın CS de türetilebilir olmasına izin verelim. Sonra, CS de Σ den Φ türetmesi var olsun. Bu türetme α olsun. Eğer α yı, DS de Σ den Φ nin PF si olsun diye nasıl değiştireceğimizi gösterebilirsek, tamamlanmış oluruz. Artık α çoktan Σ den Φ nin PF si olabilir. Bu durumda hiçbir şeyin gösterilmesi gerekmez. Fakat α bir PF değil ise, o zaman içerisinde aksiyomlar olmalıdır çünkü boşluklar türetmelerde, sadece ve sadece aksiyomlar; çıkış ve varsayım, türetme yapmanın diğer tek yolu olarak tanıtıldığı zaman oluşurlar. α daki her bir aksiyom yerine (yukarıdaki örnekteki gibi) aksiyomların bir PF sini koyun. Böylece α, α' olarak değiştirilir ve α' artık elenmiş olan tüm boşluklar olarak bir PF'dir. Bu yüzden CS de türetilebilir olan her argüman formu, DS de de türetilebilirdir. Bu sebeple eğer CS güçlü bir şekilde tamamlanmış ise DS de öyledir.

DS nin güçlü bir şekilde sağlam olduğunu göstermek için her türetilebilir argüman formunun geçerli olduğunu göstermeliyiz. CS nin güçlü sağlamlığından, CS deki türetilebilir her argüman formunun geçerli olduğunu biliriz. Bu yüzden, DS deki türetilebilir her argüman formunun CS de de türetilebilir olduğunu gösterebilirsek bitirmiş olacağız. Göstermemiz gereken şey, DS deki R, $\supset I$, ~E uygulamalarınca izin verilen akıl yürütmenin basit kalıpları CS deki türetmelerin büyük bir parçası tarafından yürütülebilir. Standart terminolojide, CS de R, $\supset I$ ve ~E nin türemiş çıkarım kuralları ([3], s.83)olduklarını göstermeliyiz. $\supset I$ nin CS de türemiş bir çıkarım kuralı olduğunu göstermek Herbrand'ın meşhur dedüksiyon teoremini kanıtlamaktır. Σ ve Φ varsayımlarından δ nın CS de bir türetmesi var ise, o zaman yalnızca Σ den $(\Phi \supset \delta)$ nın CS de bir türetmesi vardır. Başka bir deyişle, $\Sigma' \therefore \delta$ CS de türetilebilir ise, o zaman Σ' ve Σ içeren $\Sigma \therefore (\Phi \supset \delta)$ da Φ yı silerek Σ' den sağlanmıştır. Aslında, $\supset I$ nın türemiş bir kural olduğunu göstermek için ispat biraz daha fazlasını yapmalı; Σ ve Φ den δ türetmesine uygulandığı zaman yukarıdaki Σ den $(\Phi \supset \delta)$ türetmesini üreten bir tek düzenli yöntem vermelidir. Bu, iyi bilinen metotlarca (örneğin; [12], sayfa 32 deki gibi) kolayca yapılabilir. Benzer metotlar, R, ~E ve $\sim I$ nın türemiş kurallar olduğunu göstermek için kullanılabilir. Örnek olarak; tekrarın (R), türemiş bir kural olarak nasıl elde edilebileceğini görelim. Bir Φ sırası olan verilmiş bir α türetmesinden, sıraları arasında, α'nın tüm sıraları bulunan ve son sırası Φ olan başka bir türetme üretecek olan tek biçimli metodu (uniform method) vereceğiz. Verilen türetmeye (yukarıdaki gibi) $(\Phi \supset \Phi)$ türetmesini ve bu türetmeyi de Φ'a ekleyin. Sonuç bir türetme olacaktır çünkü Φ son sırası, kabul edilmiş önceki Φ sırasından ve $(\Phi \supset \Phi)$ sırasından çıkış yoluyla elde edilmişti. Bu sonuçları kullanarak, bir kişi, DS de türetilebilir olan herhangi bir argüman formunun, CS de de türetilebileceğini ve bu şekilde (CS nin güçlü sağlamlığı ile) geçerli olduğunu gösterebilir.

Aslında doğrudan PF uzunluğundaki matematiksel tümevarım yoluyla DS nin güçlü sağlamlığını kanıtlamak, belki daha az öğretici fakat daha kolaydır.

Bu yüzden DS, sağlam, tamamlanmış, güçlü bir şekilde sağlam ve güçlü bir şekilde tamamlanmıştır. Ayrıca, söylemeye bile gerek yok, DS etkilidir. Eğer DS, salt bir gerektirmeci sistemden fazlası olarak düşünülürse, (PFs) türetmeleri yoluyla yerine getirilmiş daha güçlü yeterlik koşulları olmalıdır. Aksi halde, onu gerektirmeci sistemin farklı bir türü olmak dışında herhangi bir şey olarak değerlendirmek için tam eklemli sebebe (precisely articulated reason) sahip değiliz. Onun türetmelerinin sağlam ve etkili olması yeterli değildir; sıkı da olmalı, yani maksimum miktarda ilgili mantıksal detay içermelidirler. Sezgisel olarak, tabii ki PFs bu koşulu karşılar. Ama sıkı bir PF kavramı tam olarak tanımlanmış(precisely defined) olmadığı için PFs nin sıkı olduğunu ispatlayamayız.

Burada, Tarski'nin mantıksal gerektirme veya geçerli argümanları tanımlamasından önce biz kendimizi, mantıkçılar tarafından oluşturulmuş aynı tür durumlar içinde buluruz: onlar sezgisel olarak sistemlerinin bazılarının güçlü bir şekilde sağlam olup, fakat geçerlilik tanımından yoksun olduklarını ve ispat sunamadıklarını biliyorlardı.

Sıkı terimi ile ne kastedildiğini anladığımızı varsaydığımızda, fark ederiz ki DS sıkı ispatları kodlamaya niyetli olduğu için DS den onların tümünü kodlaması istenir. Eğer bir dedüktif sistem tüm sıkı ispatları (uygun bir türünkileri) kodlarsa, o zaman biz bir dedüktif sistemi dedüktif olarak tamamlanmış olarak tanımlayabiliriz. Şimdi, sıkı bir ispatın tanımlanması ihtiyacı daha da acil hale gelir. Çünkü ya DS nin dedüktif olarak tamamlanmış olduğunu ya da olmadığını gösterebilmek istiyoruz. Tanımdan yoksun olduğumuz için sadece akla yatkın argümanlara başvurabiliriz. DS nin dedüktif olarak tamamlanmış olmadığını ileri sürüyoruz, çünkü sıkı ispatların aşağıdaki iki türünü kodlamada başarısızdır. Bunlar: (1) Σ deki varsayımlar girildikten sonra, yeni bir varsayım olan ~Φ ile başlayıp Φ ile biten akıl yürütme kalıplarına dayanarak Φ çıkarımını yapan Σ den Φ ya olan ispatlar ; (2) Σ deki varsayımlar girildikten sonra, yeni bir varsayım olan Φ ile başlayıp ~Φ ile biten akıl yürütme kalıplarına dayanarak çıkarım yapılan Σ den ~Φ ye olan ispatlar.

İspatların türlerinin ayrı ayrı örnekleri aşağıdadır.

(1) $[(\sim P \supset P)] \therefore P$

$$\begin{array}{lll} 1. & +(\sim P \supset P) & (\text{in } \Sigma) \\ 2. & \boxed{+\sim P} & (\sim\phi) \\ 3. & \boxed{\quad P} & \\ 4. & P & \end{array}$$

(2) $[(P \supset \sim P)] \therefore \sim P$

$$\begin{array}{lll} 1. & +(P \supset \sim P) & (\text{in } \Sigma) \\ 2. & \boxed{+P} & (\phi) \\ 3. & \boxed{\sim P} & \\ 4. & \sim P & \end{array}$$

Bu türlerin ispatları, kolaylıkla DS de PFs nin kısaltmaları olarak görülebilir. (aşağıdaki) (a) yı (1) deki 3. ve 4. sıranın arasına ve (aşağıdaki) (b) yi (2) deki 3. ve 4. sıranın arasına koyun – sonuç DS deki PFs olur.

(a) $\boxed{\begin{array}{r} +\sim P \\ \sim P \end{array}}$ (b) $\boxed{\begin{array}{r} +P \\ P \end{array}}$

Ancak, (1) ve (2) deki gibi akıl yürütmelerin ilkeleri sağlam ve her (klasik) matematiksel akıl yürütmenin bir parçasıdır. Dahası, sezgisel olarak bu ispatlar sıkıdır.

Genel olarak bir dedüktif sistem, mantıksal doğruları, geçerli argümanları ve sağlam ve sıkı ispatları kodlar. Bu yüzden bir dedüktif sistemin etkili, sağlam, güçlü bir şekilde sağlam ve sıkı olması gerekir. Dahası bir dedüktif sistemin tamamlanmış, güçlü bir şekilde tamamlanmış ve dedüktif olarak tamamlanmış olması istenir. Gördüğümüz gibi, bu durumların her birinin gerekçesi, bir dedüktif sistemin istenilen amaçlarında bulunmasıdır.

Tablo 1 de sistemlerin üç türünün karşılaştırmasını özetledik.

Tablo 1 Sistem Türlerinin Karşılaştırması

Mantıksal Sistemler	İlgi Alanı	Gerekli Özellikler	İstenen Özellikler
Lojistik	Kodlama	Etkili	
	Mantıksal Doğruluk	Sağlam	Tamamlanmış (post tamamlanmış)
Gerektirmeci	Kodlama	Etkili	
	Mantıksal Doğruluk	Sağlam	Tamamlanmış
	Geçerli Argümanlar	Güçlü Bir Şekilde-Sağlam	Güçlü Bir Şekilde-Tamamlanmış
Dedüktif	Kodlama	Etkili	
	Mantıksal Doğruluk	Sağlam	Tamamlanmış
	Geçerli Argümanlar	Güçlü Bir Şekilde-Sağlam	Güçlü Bir Şekilde-Tamamlanmış
	İspat	Sıkı	Dedüktif Olarak-Tamamlanmış

Bu sırada iki şeyi not edelim. İlki: bazı mantıkçılar, post-tamamlanmışlığın tamamlanmışlığa dayanması gibi bir ilişki gibi, sırasıyla güçlü tamamlanmışlığa ve dedüktif tamamlanmışlığa dayanan en fazla sağlam tamamlanmışlık ve en fazla dedüktif tamamlanmışlık kavramlarını fomüle etmeyi istenir bulabilir. İkincisi: Knelae, mantığın düşünceli tarihinde, mantığın, çıkarım kurallarını kodlamak ile ilgilenmesi gerektiğini ileri sürmüştür. Bu öneri, doğal olarak çıkarımsal sistem olarak adlandırılan dördüncü bir tür mantıksal sisteme yol açar. Bu bağlamda, DS yanlış yola girmiş olabilir çünkü geçerli argüman formlarını kodlayarak geçerli argümanları kodlamada o, çıkarım kurallarını kodluyor gibi görünüyor. Ancak, (1) çıkarımın hiçbir yapısal kuralı, salt geçerli argüman formları olarak düşünülemez; (2) geçerli argüman formu kavramını tanımlamış olsak da, bir çıkarım kuralının genel bir tanımını formüle etmiş değiliz.

4. Sıkı Kurallar ve Yapısal Kurallar

Sıkı Dedüksiyon: Yukarıda mantığın, mantıksal doğruları ve geçerli argümanları olduğu kadar ispatları kodlamak ile de ilgili olduğunu vurgulamıştık. Dedüktif sistem terimini bu üç amacı gerçekleştirmeyi isteyen mantıksal bir sistemi ifade etmek için kullanılmasını da önerdik. Akıl yürütmedeki ve mantıksal sezginin büyük atlayışlarını kodlamadaki hataların ikisinin kodlanmasından da sakınılır.

Böylece dedüktif sistemlerin amaçları tartışmasında ve sonuç olarak onlar için uygun yeterlilik koşullarının formülasyonunda, sağlam ve sıkı terimleri mühimdir. Tarski'nin mantıksal gerektirme kavramı yorumu sonucunda, sağlamlık kavramı artık problematik değildir. Ancak,sıkılık kavramı benzer dikkatli bir yorumdan yoksundur. Bu bölümün bir amacı böyle bir yorumun oluşturulabileceği sıraları önermektir.

Dedüktif sistemler araştırmasında bir "sıkı ispat" tanımına ihtiyaç duyulur. Böyle çalışmalarda, iki tür soruya cevap verebilmek istenir. İlki; dedüktif bir sistem sıkı mıdır? Yani verilen sistem tarafından kodlamış her ispat sıkı bir ispat mıdır? İkincisi; dedüktif bir sistem, dedüktif olarak tamamlanmış mıdır? Yani her sıkı ispat aslında sistem tarafından mı kodlanmıştır? Aşağıda biz kendimiz direkt olarak sıkı ispat kavramı ile ilgilenmeyiz, bunu yerine ilgili kavram olan çıkarımın sıkı kuralı ile ilgileniriz. Dahası, tartışmamızı mutlak anlamda çıkarım için kullanılan kurallarla sınırlayacağız, yani varsayım ve tekrar kuralları düşüncesini dışarıda bırakacağız.

Evvela bir kural, sıkı olarak değerlendirilmesi için sağlam olmalıdır. Dahası, bir kural olarak değerlendirilmesi için etkili olmalıdır. Sağlamlık ve etkililiğin yanında, sıkılığı da içeren iki uzak fikir gibi görünür. İlkin, sıkı bir kural maksimum miktarda ilgili mantıksal detay içeren ispatlarda çıkmalıdır – sıkı bir kural bir akıl yürütmede boşluklara izin vermemelidir. İkinci olarak, sıkı bir kural bazı anlamda basit olmalıdır.

Biz, maksimum miktarda mantıksal detayın, ya tanıtan ya da tam olarak bir mantıksal sembol oluşumunu çıkaran kurallar tarafından başarıldığını ileri sürüyoruz. Örneğin; modus tollens (~δ ve (Φ ⊃ δ) den ~Φ çıkarımı yapmak) ~ in bir oluşumunu tanıtırken ~ in ve ⊃ in bir oluşumunu çıkarırken; ayırma ⊃ nın bir oluşumunu çıkarır, koşullandırma bir oluşumu tanıtır. Terse evirme (~δ den ~Φ ye (Φ ⊃ δ) çıkarımını yapmak) ~ in iki oluşumunu çıkarır ve ⊃ nın bir oluşumunu tanıtır. İkili değilleme kuralı (~~Φ den Φ çıkarımı, Φ den ~~Φ çıkarımı yapmak) aynı zamanda iki sembol oluşumlarını açar.

Diğer yandan, basitlik mantıksal sembolün sadece bir türünü içeren tarafından örneklenebilir gibi görünmektedir. Modus ponens, çıkış, değillemeye giriş ve değillemeden çıkış bu iki türe de sahiptir, yani bu kuralların her biri : (1) mantıksal bir sembolün tam olarak bir oluşumunu ya tanıtır ya da çıkarır. (2) yalnızca bir tür mantıksal sembol içerirler. Sadece ~ i içeren ikili değilleme kuralları hariç olmak üzere, az önce yukarıda bahsedilen kurallar bu özelliklerin hiç birine sahip değildir. Mantıksal bir kuralın tam olarak bir oluşumunu tanıtan veya çıkaran her kuralın aynı zamanda kullanımda (veya ifadede) sadece bir mantıksal sembol içerdiği doğru olabilir.

Çıkarımın sıkı kural kavramının yorumunun taslağı olarak şunu sunuyoruz:

Bir çıkarım kuralı, ancak ve ancak (1) etkili ise, (2) sağlam ise, (3) mantıksal bir sembolün tam olarak bir oluşumunu tanıtıyor veya çıkarıyor ise, (4) kullanımı sadece bir mantıksal sembol içeriyor ise sıkıdır.

Bu arada belirtmek isteriz ki sadece bir mantıksal sembol içermenin bazı ilginç sonuçları vardır. Bir dedüktif sistemin gelişiminde, varsayım ve tekrar ilk sanılabilir. Daha sonra her bir mantıksal sembol diğer sembollerin tamamlanmış bağımsızlığı içinde ele alınabilir. Dilde üzerine düşünülen her bir sembol için, kişi bir tanıtım kuralını ve dildeki başka herhangi bir mantıksal sembole referansı olmaksızın bir çıkış kuralını sunabilir. Dedüktif sistem kurallarının belirtilmesinde mantıksal semboller arasında hiçbir etkileşim olmadığı için karmaşık bir dedüktif sistemi, karşılıklı olarak bağımsız basit dedüktif sistemlerin bir ürünü olarak değerlendirmek mümkündür. Bu basit sistemlerin her biri bir çıkış kuralına, bir tanıtım kuralına, bir varsayım ve bir tekrar kuralına dayanır. LS, değillemenin ve 'ise' nin basit bir sisteminin ürünüdür. Her bir basit sisteme tamamlanmışlık ve güçlü tamamlanmışlık soruları sorulabilir. Karmaşık bir sistemi basit bir sistemlere ayırarak, belirlenememezlik (undecideability), tamamlanmamışlık (uncompleteness), yoğun olmama (noncompactness) gibi istenebilir çeşitli özelliklerin sebeplerini dışlamak muhtemel olabilir.

Bu fikrin diğer bir muhtemel kullanımı da, başka bir sisteme dayalı bir sistemin kurallarını çevirerek bir küme mantıksal sembole dayanan gerektirmeci sistemin oluşumuyla ilgilenen Hiz[8]'in sonuçlarını anlamak için daha sezgisel bir bakış açısı sağlamada bulunur. Hiz örnek yoluyla, tamamlanmış bir sistemi çevirerek bir kişinin tamamlanmış olmayan bir sistem sağlayabileceğini vurgulamaktadır. Altında yatan fikir, bir çıkarım kuralını çevirirken sağlamlığın korunduğu fakat "dedüktif güç"ün korunmadığıdır.

Sıkı bir sistemde, dedüktif güç, bir sembolü başlıca sembol olarak giriş ve bir sembolü başlıca sembol olma durumundan ayırma becerisi olarak düşünülebilir. Mantıksal sembolün her tanımı başka en az iki sembol içerdiği için, herhangi bir sıkı sistemin çevirisi sıkı olmayan bir sistemde sonuçlanmalıdır. Örneğin; eğer $(P \supset Q) =_{df} \sim(P \ \& \sim Q)$ kullanarak ayırmayı çevirirsek, o zaman aşağıdaki çıkış kuralını elde ederiz: P ve ~(P & ~Q) dan Q çıkar. Böylece çıkışın sıkı kuralını çevirerek ~ in iki oluşumunu ve & in bir oluşumunu çıkaran bir kural elde etmiş oluruz ki dahası bu, ifadesinde iki farklı mantıksal sembol içerir. Ayrıca ayırma, ⊃ yi başlıca durumdan çıkarma gücüne sahipken, yeni kural & yi sadece ~(P & ~Q) karmaşık yapısında geçtiğinde çıkarabilir. Şartlandırmayı çevirerek şu kuralı elde ederiz: P den Q ya akıl yürütme kalıbına dayanarak ~(P & ~Q) çıkarımı yapılır. Bu kural sıkı değildir. Dahası özü itibariyle şartlandırmadan daha zayıftır çünkü & tanıtımının esas pozisyona girmesine izin vermez. Bağlayıcılara ve yukarıdaki iki çeviri haricinde DS için aynı kuralları olan ~ e dayanan dedüktif bir sistemi düşünün. &-çıkış kuralı ~(P & ~Q) bağlamından sadece & yi çıkardığı için kişi bu sistemde kanıtlayamayacağı bir varsayıma neden olur.

$[(P \ \& \ Q)] \therefore P$

Aslında aşağıdaki iki matris, içerisinde 1 olarak bir P çıkarımı, (P & Q) doğru kılan 1 olarak (belirtilmiş) Q yu ve Q yanlış (belirtilmemiş) barındıran bir sistem için bize sağlam bir üç değerli semantik sağlar.

&	2	1	0	~
2*	2	0	0	0
1	0	2	0	2
0	0	0	0	2

Diğer bir ilginç uygulama da modal mantıktaki denemelerdir. Modal yöneticilere sahip bir dile dayanan modal olmayan bir mantık verilerek, bir kişi her yönetici için iki sıkı kural göstererek modal kuralları oluşturabilir. Modal problemler, sistemin geri kalanından yalıtılmışlık içerisinde görülebilir. Uygulanabilir olan yerde, bu yaklaşım modal sistemlerle geniş ölçüde deneme imkanı sağlar.

Bu makalenin, makul bir şekilde çözülüyor olarak düşünülebilecek problemlerin bakış açısından mantıksal sistemleri araştırmak olan esas amacına nazaran bu düşüncelerin ikincil oldukları vurgulanmalıdır. Lojistik sistemler açıkça mantıksal doğruları kodlamayı hedefler. Gerektirmeci sistemler açıkça geçerli argümanları kodlamayı hedefler. Bu iki tip sistemin tamamlanmış olarak yeterli olmak için yerine getirmesi gereken durumlar zaten iyi bilinmektedir. Dedüktif sistemler açıkça ek bir amacı hedefler, o da; ispatların kodlanmasıdır. Dedüktif sistemlerin yeterlik durumlarını formüle etme problemiyle yüzleşerek, basit ve "boşluksuz" ispatların varlıklarını varsaymaya zorlandık. Bu, sırayla, yukarıda gösterilen sıkı kavramının formülasyonunda bildirildi.

Bizim sıkı kavramı formülasyonumuz en az iki yönden kusurludur.

İlkin, temel "sıkı" fikri etkililiği, sağlamlığı, basitliği ve ilgili maksimum mantıksal detayı içermesine rağmen, aslında bizim tanımımız her birine karşılık gelen bileşenlere sahiptir – bizim tanımımız hala fazla özel görünüyor. İstememiz gereken şey, daha genel bir kavrama sahip olmaktır. Aslında, şöyle bir sıkı kavramına sahip olmayı istemeliyiz: (1) kavram öyle geneldir ki DS sisteminin sıkı olduğunu gösteren matematiksel bir ispat gerektirecektir, (2) kavram açıkça fikri simgeler.

İkinci olarak - ve belki de önemsizce - mantıksal bir sembole giriş in veya çıkışın, matematiksel olarak tam bir tanımını vermedik. Sezgisel fikir kusursuzca açıktır ve her sembol için girişin ve çıkışın amaca özel bir tanımını vermek kolaydır. Fakat, özel herhangi bir sembole işaret etmeyen ve amaca özel tanımların takip ettiği genel bir tanımın olması çok daha iyi olacaktır. Bu eksiklikler, bu çalışmanın esas amacıyla ilgili değildir. Şu anki amacımıza göre, ispat kavramı için yeterlik koşullarının bazı tam formülasyonunun gerekliliğinin ortaya çıkmasını sağlamış olmak yeterlidir. Dahası bizim "sıkı" fikrimiz için yukarıda önerilen uygulamalar, fikir söylenmek istenen amacına uygunluğu kanıtlasa da kanıtlamasa da, hala geçerlidir.

Çıkarımın Yapısal Kuralları: Dolaysız çıkarım kuralları ve yapısal çıkarım kuralları arasındaki fark yukarıda tanıtılmıştır. Bu tartışmada varsayım, çıkarımın bir kuralı olarak düşünülmemektedir çünkü kendisinde çıkarımlara izin vermez. Her bir çıkarım kuralı – yukarıda tartışılmış olan veya olmayan – ya belli bir sayıda sıraya dayanılarak uygulanır ya da uzunluğu önceden belirlenmemiş bir akıl yürütme kalıbına dayanılarak uygulanır. Örneğin; şartlandırma Φ den δ ye akıl yürütmeye dayanarak uygulanırken ve bu akıl yürütme sonsuz uzunlukta olabilirken çıkış daima iki sıraya uygulanır. Benzer şekilde, tikel özelleme (existential instantiation), (1) bir (∃x) Φ sırasına ve (2) Φ den δ ye bir akıl yürütme kalıbına dayanarak (varsayımlarının hiçbiri bağsız x içermeyen) bir ispatta (bağsız bir x içermeyen) bir δ çıkarımına izin verirken, tikel genellemesi (existential generalisation) daima bir sıraya uygulanır ve uygulandığı ispatın geri kalanına bir bağlılığı yoktur. Koordineli olarak, tümel genellemeyi (universal generalisation) ispatın tümüne uygulamak için düşünülmesi gerekirken, tümel genellemeyi (universal generalisation) tek bir sıraya dayanan çıkarıma izin verir.

Yapısal olan bir çıkarım kuralı için uygulandığı kurala göre sıraların sayısının, sabit olmaması gerekli görünmektedir. Bu, kuralları yapısal olarak sınıflandırmak için bir ölçüt sağlar.

Bir dedüktif akıl yürütme külliyatına giderek veya farkında olmadan kendi kendine birkaç ispat yazarak, bu kuralların içinde bulunan akıl yürütme kalıplarının mantığını keşfetmek mümkündür. Yapısal kurallardaki önemli fikir, akıl yürütme kalıbıdır – yukarıda verilen gerekli ve yeterli koşullar, salt gramere ait ölçütlerdir.

Belli ki, bazı çıkarım kurallarının yapısal doğası daha önce açık bir şekilde tanınmamıştır. Aslında, kısıtlamalar ile dolaysız çıkarım kuralları formundaki yapısal kuralları ifade etme eğilimi ile belli yapısal kuralları formüle etme çabasında, yetkin mantıkçılar tarafından deneyimlenen zorlukları açıklamak mümkündür.

5. Sonuç

Diğer Hiyerarşi: Belirtilmelidir ki yukarıda tartışılan sistem hiyerarşileri aşağıdaki hiyerarşiden tamamen bağımsızdır: eklemler mantığı, birlik (monadic) yüklemler mantığı, yüklemler mantığı, ikinci dereceden mantık vs. bu sonraki hiyerarşi, mantıksal analizin derinliği ile ilgilenir: ikinci dereceden mantıkta mantıksal doğrular olarak analiz edilebilir olup yüklemler mantığında analiz edilebilir olmayan cümleler vardır. Yüklemler mantığında mantıksal doğrular olarak analiz edilebilir olup, birlik yüklemler mantığında analiz edilebilir olmayan cümleler vardır. Bu durum birlik (monadic) yüklemler_ve eklemler mantığı karşılaştırıldığında da aynıdır. Örnekler aşağıdadır:

1) Her şeyin bir tabiatı vardır.

İkinci derece: (Ax)(ƎP)Px

Yüklem: (Ax)Hx

2) Her şeyin bir nedeni varsa o zaman her şey bir nedene sahiptir.

Yüklem: ((Ǝx)(Ay) Cxy ⊃ (Ay)(Ǝx)Cxy)

Birli: ((Ǝx) Cx ⊃ (Ay)Hy)

3) Tüm insanlar Yunan ise ve Sokrates bir insan ise o halde Sokrates Yunan'dır.

Birli: ((Ax) (Mx ⊃ Gx) ⊃ (Ms ⊃ Gs)

Önermesel: (A ⊃ (M ⊃ G))

Bu hiyerarşi, tabii ki yukarıya doğru genişletilebilir ve modal mantıklar için tekrar edilebilir. Yukarıda konusu açılan problemlere (örneklerimizi çok basit bir önermesel mantıktan almamıza rağmen) genel olarak değinilmiştir. Tabii ki mantıksal sistemlere ilişkin tüm gözlemler, bahsedilen hiyerarşilerin her bir seviyesine uygulanır.

Mantıksal Sistemler Hiyerarşisi: Mantıksal sistemlerin git gide karmaşık hale gelen üç türünü gördük. Lojistik sistemler, mantıksal doğruları kodlama problemine uygun görüldü. Gerektirmeci sistemler mantıksal doğruları kodlamıştır. Bu görev büyük zorlukla sadece lojistik sistemler tarafından yapılabilir olmasına rağmen, gerektirmeci sistemler aynı zamanda geçerli argümanları kodlama görevine de uygundur. Son olarak dedüktif sistem mantıksal doğrulara ve geçerli argümanlara ek olarak ispatları da kodlamıştır.

Bu üç tip sistemin artan kapsamlılığını karşılaştırmak, sistemler için artan kapsamlı yeterlik koşullarıdır. İdeal lojistik sistem sağlam ve tamamlanmıştır. İdeal gerektirmeci sistem güçlü bir şekilde sağlam ve güçlü bir şekilde tamamlanmıştır. Sağlamlık ve tamamlanmışlık, sırasıyla güçlü sağlamlığın ve güçlü tamamlanmışlığın özel durumları olduğu için gerektirmeci sistemler özü itibariyle lojistik sistemlerden daha kapsamlıdır. Dahası, bu iki tür sistem arasındaki ayrım, bir lojistik sistemin mantıken güçlü sağlamlık yani 'post tamamlanmışlık' ile bağdaşmayan bir durumu yerine getirmesi beklenebilir. Lojistik sistemler ve gerektirmeci sistemler için yeterlik koşullarının net gerekçeleri için kişi onların ayrı ayrı istenilen amaçlarına atıfta bulunabilir.

Gerçek dedüktif uygulamayı düşünerek geçerli argümanları ve ispatları birbirinden ayırmaya sürükleniriz. Böylece ispatların kodlanmasını ve bu amacı güden bir sistemi göstermek için dedüktif sistem teriminin kullanımını makul bir amaç olarak kurabiliriz. Resmi olmayan fakat düşünceli bir bakış açısından, iki nedenden ispatların kodlanması olarak gerektirmeci sistemleri reddetmeye sürükleniriz. İlkin; onlar sezgisel olarak sıkı ispatları (şartlandırmayı veya tikel özellemeyi(existential instantation) kullanan), boşluklara sahipmiş gibi işlerler, ikincisi; gerektirmeci sistemlerdeki öncüllerden olan türetmeler, apaçık boşluklara sahiptir. Dahası, gerekli koşullar genelde gerektirmeci sistemler için anlaşma sağlar, yani güçlü sağlamlığın ve etkililiğin herhangi bir sıkı olma fikrine referansı yoktur. Böylece; (1) ispatların sıkı olmak için karşılaması gereken ve (2) salt güçlü sağlamlığı ve etkililiği aşan bir matematiksel koşulu formüle etme problemi ile yüzleşiriz.

Bu sonuç bölümünün amacı, yukarıda tanıtılan sıkı olma kavramının doğru olup olmadığı noktasının hemen yanındadır. (1) İspatın kodlamasının makul bir amaç olduğu gerçeğine ve (2) gerektirmeci sistemlerin bu amacı yerine getirmediği açık ama bilimsel olmayan bu gerçeğe dayanarak, üçüncü tip bir sisteme ihtiyaç olduğu sonucunu çıkarırız.

Tarihsel Notlar: Bu makaleyi, yukarıdaki mantıksal sistemler hiyerarşisi ve onların modern zamanlardaki gerçek tarihsel gelişimi arasındaki ilişkinin kısa bir değerlendirmesi ile sonuçlandırmak istiyoruz.

Bu değerlendirme, modern mantık tarihinin bir taslağı olarak düşünülemez; yukarıda değerlendirilen sorular, modern mantığın önemli fakat nispeten küçük bir bölümünü oluşturur.

Mantık tarihinin bu bölümünde birbiriyle ilişkili iki temaya odaklanıyoruz. Birincisi; mantıksal sistemlerin kuruluşu, ikincisi; bu sistemler için yeterlik koşullarının formülasyonu. Açıkçası, oluşturmak istediğimiz sonuçlar aşağıdadır:

1) Sistemlerin kendisine ilişkin olarak; lojistik sistemler, gerektirmeci sistemler ve dedüktif sistemler bu sırada geliştirilmiştir.

2) Yeterlik koşullarına ilişkin olarak; ilk olarak tamamlanmışlık, sağlamlık ve 'post tamamlanmışlık' kavramları formüle edildi. Sonra güçlü sağlamlık, güçlü tamamlanmışlık ve son olarak sıkı kavramına değinildi.

Öncelik kriterimiz yayım tarihidir. Daha karmaşık bir kriter kullanımı hatırı sayılır derecede tarihsel araştırma içerirdi.

En eski sistemler kesinlikle lojistik sistemlerdir. Frege ve Russell için motivasyon, matematiğin tüm teoremlerinin mantıksal yasalar olduğu inancıydı. 1934'de ispatın doğasına ilişkin ilk araştırmalar yayınlandı ([10] syf, 538,539). Biri Gertzen, diğeri Jaskowski sayesindedir. Bu alanda daha sonraki gelişmeler bu iki çalışmaya dayanır ([10] syf. 539 ve [1] syf. V) 1934'den önce geliştirilen gerektirmeci sistemleri kurmak için ilk 1930'da yayınlanan Tarski'nin gerektirmeci sistemlerin önemli metodolojik çalışmasından bahsetmek yeterlidir ([19] syf. 30-37).

Lojistik sistemler için yeterlik koşulları, 1920'de Post tarafından formüle edildi. Post, sağlamlığı, tamamlanmışlığı ve post tamamlanmışlığı (bu adlarla olmasa da) tartışmıştır. Görünür bir şekilde, güçlü sağlamlık ve güçlü tamamlanmışlık basılı olarak ilk kez 1951'de Abraham Robinson tarafından tartışılmıştır ([18], sırasıyla teorem 3.1.2 ve 3.2.2). Bu kavramların önemi açısından bu, tamamen şaşırtıcı bir sonuçtur; ancak birçok mantıkçının, 1951'den önce düşüncelerinde ve öğretimlerinde bu kavramları kullanmış oldukları muhtemelen doğrudur. Bu bağlamda birinci derece mantığın güçlü tamamlanmışlığını önemsiz bir şekilde belirten teoremlerin 1931'de Gödel [5] ve 1949'da Henkin [6] tarafından yayımlandığı not etmeye değerdir. Bu teorem kavramsal olarak aslında onların ispatladıkları teoremlerden daha ilginç olmasına rağmen ikisi de güçlü tamamlanmışlıktan bahsetmez.

Belirleyebildiğim kadarıyla bu makale haricinde, etkililiğin ve sağlamlığın üzerinde ispatlar için yayımlanmış hiçbir durum değerlendirmesi yoktur.

Teşekkür: Bu makalenin daha önceki halinin meslektaşlarım James Munz ve William Snavely ile yapılan ilginç ve ilham verici tartışmaları için teşekkür etmek bir zevktir. Ayrıca, bu çalışmaya çeşitli şekillerdeki katkıları için aşağıda ismi geçen öğrencilere de teşekkür etmek isterim: Uwe Henke, Nicolas Noviello, Howard Wasserman ve George Weaver. Bu çalışma aslen Pennsylvania Üniversitesi Felsefe Bölümüne Aralık 1966'da bir ders olarak verilmek üzere kaleme alınmıştır.

• • •

Kaynakça

[1] Anderson, J. M., and Johnstone, H. W., *Natural Deduction*, Belmont, California, 1962.

[2] Bochenski, I. M., *A History of Formal Logic* (tr. Thomas, Ivo), Notre Dame, Indiana, 1961.

[3] Church, A., *Introduction to Mathematical Logic*, Princeton, 1956.

[4] Copi, I. M., and Gould, J. A., *Readings on Logic*, New York, 1964.

[5] Gödel, K., "Die Vollstandigkeit der Axiome das logischen Functionenkalkuls," *Monatshefte für Mathematik und Physik*, vol. xxxvii, 1930, p. 349.

[6] Henkin, L., "The completeness of the first order functional calculus," *Journal of Symbolic Logic*, vol. 14, 1949, p. 159.

[7] Hilbert, D., and Ackermann, W., *Principles of Mathematical Logic* (tr. Hammond, Leckie, and Steinhardt), New York, 1950.

[8] Hiż, H., "A warning about translating axioms," *American Mathematical Monthly*, vol. LXV, 1958, p. 613.

[9] Kalish, D., and Montague, R., *Logic: Techniques of Formal Reasoning*, New York, 1964.

[10] Kneale, W., and Kneale, M., *The Development of Logic*, Oxford, 1962.

[11] Lewis, C. I., and Langford, C. H., *Symbolic Logic*, 2nd ed., New York, 1959.

[12] Lightstone, A. H., *The Axiomatic Method*, Prentice-Hall, Englewood Cliffs, N.J., 1964.

[13] Mates, B., *Elementary Logic*, New York, 1965.

[14] Mendelson, Elliot, *Introduction to Mathematical Logic*, Princeton, 1964.

[15] Parry, W. T., "Comments on a variant form of natural deduction," *Journal of Symbolic Logic*, vol. 30, 1965, p. 119.

[16] Post, E. L., "Introduction to general theory of elementary propositions," *American Journal of Mathematics*, vol. 43, 1921, p. 163.

[17] Quine, W. V. O., *Methods of Logic* (revised edition), New York, 1959.

[18] Robinson, A., *On the Metamathematics of Algebra*, Amsterdam, 1951.

[I9] Tarski, A., Logic, *Semantics and Metamathematics* (tr. Woodger, J. H.), Oxford, 1956.

[20] Whitehead, A. N., and Russell, B., *Principia: Mathematica to 56*, Cambridge, 1962.