논리연구 10-1(2007) pp.65~98

논리적 귀결과 논리 상항의 의미*†

이 종 권

【요약문】 정인교는 그의 최근 논문에서 논리적 귀결 관계에 의해 논리 상항의 의미를 정의함에 있어 통상적인 도입 규칙과 제거 규칙에 의거하는 포퍼의 접근법과, 도입 규칙에만 의존하는 정당화주의적 접근법, 그리고 제거 규칙에만 의존하는 실용주의적 접근법을 구분한 바 있다. 이 글에서는 연언과 선언의 연결어의 경우에는 그 세 가지가 동등하다는 것을, 그리고 조건과 부정의 연결어의 경우에는 제거 규칙에 의거하는 실용주의적 접근법과 포퍼의 접근법이 대등하다는 것을, 타르스키가 처음 확립한 논리적 귀결에 관한 공리적 체계에 의존하여 보일 것이다.

【주요어】 논리적 귀결, 증명 이론, 논리 상항, 정인교, 타르스키, 포퍼

* 접수완료: 2007. 1. 30. / 심사 및 수정완료: 2007. 2. 18.
† 이 논문은 2006년도 중앙대학교 학술연구비 지원에 의한 것임.

I.

논리적 귀결 관계(logical consequence)는 통상적으로 논리 상수(logical constant)에 의거하여 정의된다. 그러므로 논리 상수를 어떤 식으로 정의하는가에 따라서 서로 다른 의미의 논리적 귀결의 개념이 얻어진다. 논리 상수를 모델을 이용하여 정의한다면 모델 이론적(model-theoretical) 혹은 의미론적 귀결의 개념이 얻어질 것이며 추론 규칙에 의거하여 정의한다면 통사론적 혹은 증명 이론적(proof-theoretical) 귀결의 개념이 얻어질 것이다. 그런데 정인교는 그의 최근의 논문에서 거꾸로 논리적 귀결 관계에 의거하여 논리 상수의 의미를 정의하는 문제를 논의하고 있다.[1] 하나의 논리 상수의 의미를 일단 논리적 귀결 관계에 의해 정의하면 그러한 논리 상수와 관련하여 어떤 추론 규칙이 성립하는지는 자연스럽게 결정될 것이다. 그러한 정의로부터 그 논리 상수에 관한 통상적인 도입 규칙과 제거 규칙이 따라 나와야 함은 물론이다.

정인교는 위에서 언급한 글에서 논리적 귀결 관계에 의거하여 논리 상수를 정의하는 방식으로서 세 가지를 구분하고 있다. 하나는 포퍼의 방식으로서 그것은 "'논리적 귀결' 혹은 '연역 가능성'에 의거해 논리 상항의 의미를 명시적으로 정의하고, 이렇게 정의된 논리 상항의 의미에 의해 추론 규칙을 정당화"하는 것이다.[2] 반면에 "정당화주의적 의미론에서는 도입규칙들이 논리상항의 의미를 고정하는 역할을 하고, 주장의 귀결을 핵심 개념으로 하는 실용주의적 의미론에서는 제거 규칙들이 그런 역할을 하게 된다."[3] 정인

1) 정인교(2003). 그러나 그 글에서 정인교의 주목적은 단지 논리 상수가 논리적 귀결에 의거하여 어떻게 정의되는가 하는 것이 아니라 어떤 정의가 "자체적으로 정당한" 정의가 될 수 있는가 하는 것이다.

2) 정인교(2003), 11쪽.

교는 실제로 도입규칙에만 의거하여 논리 상항을 정의하는 방식과 제거규칙에만 의거하여 논리 상항을 정의하는 방식을 제시하고 그것들이 서로 등가임을 증명하려 하고 있다.

논리 상항의 의미를 논리적 귀결 내지는 연역 가능성에 의거하여 정의한다는 것은 곧 논리 상항의 의미를 정의하기 전에 논리적 귀결의 개념이 이미 주어진 것으로 간주한다는 것을 의미한다. 만일 논리적 귀결의 개념을 공리적 방식에 의해 정의하고 또 그것에 의거하여 논리 상항의 의미를 정의하는 것은 논리 상항의 의미를 공리적 방식으로 정의하고 또 그렇게 정의된 논리 상항에 의거하여 논리적 귀결을 정의하는 것과는 정방대의 방식이지만 그러한 방식에 의해서도 온전한 공리적 논리 체계가 수립될 것으로 기대할 수 있을 것이다. 실제로 정인교는 논리적 귀결 관계에 관한 공리를 제시하고 있다. 이 글에서는 그와 맞먹는, 타르스키식의 공리 체계에 의거하여 그로부터 우선 논리적 귀결에 관한 몇 가지 유용한 정리를 유도할 것이다. 그리고 그러한 정리를 이용하여 연언과 선언의 연결어에 대한 정인교의 정의가 포퍼식의 정의와 대등하다는 것을 보일 것이며, 그리고 조건과 부정의 연결어의 경우에는 제거 규칙이 포퍼식의 정의와 동치라는 것을 보일 것이다.

Ⅱ.

이 글의 논의를 편의상 명제 논리 $\mathcal{L}$로 국한하기로 하고 $\mathcal{L}$의 논리식들의 집합을 S라고 하자. S는 통상적인 재귀적인(recursive) 방식에 따라 정의된다. 재귀적인 방식의 정의란 어떤 집합을 그 집합의 폐포 기저(closure base)라고 부르는 하나의 집합과 연산들

3) 정인교(2003), 10쪽.

의 집합에 의거한 정의를 의미한다. 지금 집합 P가 집합 R에 속한 모든 연산들에 대해 닫혀져 있다고 할 경우 다음과 같은 집합을 생각할 수 있다. 그러한 집합을 R에 대한 폐포 기저 P의 폐포 (closure of P under R)라고 부른다.

$C_R(P)=\cap\{X$: $P\subseteq X$이고 X는 집합 R에 속한 모든 연산에 대해 닫혀져 있다$\}$.[4)]

$\mathcal{L}$의 논리식들의 집합 S는 연산 ¬, ∧, ∨, →에 대한 명제 변항들의 집합 P의 폐포로서 정의할 수 있다. 그러한 집합은 가부번, 즉 card$(S)\leq\aleph_0$이다. 아래에서 X, Y, Z 등은 S의 멱집합(power set) $\mathscr{P}(S)$의 원소, 즉 논리식들의 집합을 나타내며 A, B, C 등은 S의 원소, 즉 $\mathcal{L}$의 논리식을 나타내는 기호로 사용된다. 이후 오해의 여지가 없는 한 단위 집합 $\{A\}$를 단순히 A로도, 그리고 합집합 $X\cup Y$는 단순히 X, Y로도 쓰기로 한다. 또한 언어 $\mathcal{L}$에 관해 진술하는 메타언어에서의 '이고'. '이거나', '이면', '인 경우 또 오직 그 경우에 한해'와 같은 연결어를 각각 '&', '∨'. '⇒', '⇔'등의 기호를 사용해서 나타낼 것이며, '모든'과 '어떤'을 나타내기 위해서는 각각 '( )', '(E)'를 사용할 것이다.[5)]

어떤 언어가 지니는 논리적 특징은 그 언어에서 어떤 논리식들이 어떤 다른 논리식들로부터 논리적으로 귀결되는가에 의해, 다시

4) 어떤 집합이 한 연산에 대해 닫혀 있다(closed under an operation)는 것은 그 집합의 원소들에 연산이 적용된 결과가 항상 그 집합에 속한다는 의미이다. 문제의 집합이 어떤 연산들의 집합에 속하는 모든 연산에 닫혀 있을 경우, 그 집합은 그 연산들의 집합에 대해 닫혀져 있다고 한다. 폐포 $C_R(P)$의 정의는 Pollock(1990), 39쪽에서 제시된 것이다.

5) 기호 ∨는 대상 언어 $\mathcal{L}$과 메타언어에서 공통적으로 사용되지만 어떤 언어에 속하는 것인지는 그 기호들이 사용되는 문맥에 의해 구분될 수 잇을 것이다.

말해 그 언어에서 성립하는 논리적 귀결 관계에 의해 규정된다고 할 수 있다. 이러한 의미에서 논리적 귀결 관계는 논리학에서 가장 핵심적인 개념이라고 말할 수 있다. 그러므로 타당한 추론과 타당하지 않은 추론을 구별하는 원리를 확립하는 것을 궁극 목표로 하는 한에서 논리학이 최종적으로 노리는 것은 문제되는 언어에서 성립하는 논리적 귀결 관계를 정의하는 일이 된다. 어떤 언어에서의 논리적 귀결 관계는 그 언어에 속하는 논리식들의 집합과 다른 하나의 논리식 간에 성립하는 관계, 즉 $\wp(S)\times S$의 부분 집합으로 볼 수 있다. 논리적 귀결 관계의 전제는 $\wp(S)$의 원소, 곧 S의 부분 집합이고 결론은 S의 원소, 즉 논리식이 된다.

힐버트(David Hilbert)와 카르납(Rudolph Carnap) 같은 현대 논리의 개척자들에 의해 확립된 논리적 귀결 개념에 의하면 어떤 일단의 논리식으로부터 그 언어에서 허용된 추론 규칙을 (유한번) 반복 적용함으로써 하나의 논리식을 추론해 낼 수 있을 때, 전자의 논리식들의 집합과 후자의 논리식 사이에 논리적 귀결 관계가 성립된다. 경우에 따라서는 추론 과정에서 공리라고 불리는 논리식을 사용하는 것이 허용되기도 한다. 추론 규칙이나 공리는 논리식의 구조만을 언급할 뿐 그것의 참, 거짓과는 무관하므로 그에 의거한 논리적 귀결의 정의는 통사론적(syntactical) 정의라고 말할 수 있다. 형식 언어에서의 진리를 정의하기 이전 타르스키가 한 다음과 같은 언급은 당시 타르스키도 오직 이러한 의미의 논리적 귀결만을 생각하고 있었다는 시사하는 것으로 보인다.

> A를 특정 학문에 속하는 임의의 문장들의 집합이라고 하자. 집합 A로부터 이른바 추론 규칙이라고 하는 어떤 연산의 도움을 빌어 새로운 문장들이 이끌려 나오는데, 이들을 집합 A의 귀결이라고 부른다. 이러한 추론 규칙을 확립하고 그러한 규칙의 도움을 빌어 귀결의 개념을 정의하는 일은 특정한 메타 학문의 작업에 속한다. 통상적인 집합론의 용어로 표현한다면 그러한 정의의 틀을 다음과 같이 정식화할 수 있다.

집합 A의 귀결은 집합 A를 포함하고 주어진 추론 규칙에 대해 닫혀진 모든 집합의 교집합이다.[6]

언어 $\mathcal{L}$에서 허용된 추론 규칙이 모두 연산이라고 하고 그것들로 이루어진 집합을 R이라고 하자. 또한 그 언어에서 가정한 공리들의 집합을 A_0라고 하자. 이때 다음과 같은 폐포를 생각할 수 있다.

(1) $C_R(A_0 \cup X) = \cap\{Y: A_0 \cup X \subseteq Y$이고 Y는 집합 R에 대해 닫혀져 있다$\}$.

위에서의 타르스키의 언급에 의하면 집합 $C_R(A_0 \cup X)$가 바로 집합 X로부터의 귀결이라는 것이다. 따라서 논리식 A가 X로부터 논리적으로 귀결된다는 것은 곧 $A \in C_R(A_0 \cup X)$라는 뜻이다. 그런데 $C_R(A_0 \cup X)$의 정의에 의해 A가 집합 $C_R(A_0 \cup X)$에 속하는 것은 집합 X의 원소와 A_0의 원소에 R에 속한 추론 규칙을 (유한번) 반복 적용함으로써 A가 얻어질 수 있을 경우 또 오직 그 경우에 한한다. 이 후자의 경우가 성립하면 그 마지막 항이 A이며 각 항은 X 혹은 A_0의 원소이거나, 혹은 그 항보다 이전에 있는 항들로부터 R에 속한 어떤 추론 규칙을 이용하여 이끌려 나오는 그러한 배열(sequence)이 존재하게 된다. 이러한 배열을 문제의 언어에서 X로부터 A의 연역(deduction)이라고 하면 $A \in C_R(A_0 \cup X)$인 것은 $\mathcal{L}$에서 X로부터 A의 연역(deduction)이 적어도 하나 존재할 경우 또 오직 그 경우에 한한다. 이제 통사론적 혹은 증명 이론적 논리적 귀결을 '$X \vdash A$'로 나타내면 다음과 같이 쓸 수 있다.

(2) $X \vdash A \Leftrightarrow A \in C_R(A_0 \cup X) \Leftrightarrow \mathcal{L}$에서 X로부터 A의 연역이 적어도 하나 존재한다.

6) Tarski(1956b), 63쪽.

그러나 Tarski(1956c)는 이러한 논리적 귀결의 개념을 배격한다. 타르스키는 그곳에서 적절한 귀결 개념의 정의가 임의적인 것이 되어서는 안 되며 일상적 언어의 통상적 용법에 부합해야 할 것임을 강조하면서,[7] 위의 증명 이론적 논리적 귀결의 개념을 아무리 다듬어도 귀결 개념이 지니고 있는 완전한 직관적인 내용을 파악하는데 성공하지 못할 것이라고 주장한다.[8] 타르스키가 그 근거로 지적하는 것은 괴델의 불완전 정리이다. 그는 괴델의 정리에 의해 "(특정한 기초적인 성격의 어떤 이론을 제외한) 모든 증명 이론에서 통상의 추론 규칙을 전혀 새로운 구조적인 규칙으로 아무리 보완한다고 하더라도 통상적인 의미에서는(in the usual sense) 이 이론의 정리로부터 귀결됨에도 불구하고 받아들여진 추론 규칙에 의거해서는 이 이론에서 증명이 불가능한 문장들을 구성해 내는 것이 가능하다"는 사실을 지적하면서 증명 이론적 논리적 귀결의 개념으로는 포착되지 않는 '통상적인 의미'의 논리적 귀결의 개념을 정의할 것을 주장하고 있다.[9] 타르스키가 새로이 정의하려고 하는 논리적 귀결의 개념은 의미론적 개념이라고 말할 수 있는데 그러한 정의는 명제 논리 언어인 $\mathcal{L}$에 대해서는 다음과 같이 전개된다.

$\mathcal{L}$에서의 논리적 귀결에 대한 의미론적인 정의는 이가 명제 논리의 경우 집합 S로부터 집합 $\{t, f\}$로의 함수인 배치(配値) 함수(valuation)를 기초로 주어진다. 만일 어떤 논리식 A에 대해 배치 함수 v가 t의 값을 할당할 경우, 즉 $v(A)=t$일 경우 v는 A를 만족한다고 하고, 또한 집합 X에 속하는 모든 논리식 B에 대해

7) Tarski(1956c), 409쪽. "…in defining this concept (of logical consequence), efforts were made to adhere to the common usage of the language of everyday life."

8) Tarski(1956c), 412쪽.

9) Tarski(1956c), 같은 곳.

$\upsilon(B)=t$인 경우, 즉 $(B \in X)(\upsilon(B)=t)$인 경우 υ는 집합 X를 만족하는 것으로 정의한다고 하자. 논리식의 집합 X로부터 논리식 A가 의미론적인 뜻에서 논리적으로 귀결되는 것은 모든 X를 만족하는 모든 배치 함수는 또한 A를 만족한다는 것으로, 다시 말해 X를 만족하면서 A를 만족하지 않는 배치 함수는 존재하지 않는다는 것으로 정의된다. 논리식 A가 의미론적인 뜻에서 집합 X로부터 귀결된다는 것을 기호 '$X \vDash A$'로 나타내기로 하고 '$(B \in X)(\upsilon(B)=t)$'를 '$\upsilon(X)$'로 줄여 쓰기로 하면 다음과 같이 쓸 수 있다.

(3) $X \vDash A \Leftrightarrow (\upsilon)(\upsilon(X) \Rightarrow \upsilon(A))$

논리적 귀결 개념에 관한 통사론적 혹은 증명 이론적 정의 (2)는 연역의 존재에 의거하고 있는 반면에 위의 의미론적 정의 (3)은 배치 함수에 대한 보편 양화에 의존하고 있다는 특징을 지닌다.[10] 정의 (2)에서 연역의 개념은 $\mathcal{L}$의 공리와 추론 규칙에 의존하고 있는데 그것에 의해 $\mathcal{L}$의 연결어의 의미가 한정된다고 말할 수 있

10) Tarski(1956c)에서의 타르스키의 의미론적 정의는 일계 술어 논리 언어를 대상으로 한 것이며 따라서 배치 함수 이외에 다른 장치를 이용하고 있다. 타르스키가 이용하고 있는 장치는 배열(sequence) 함수와 배열 함수에 의한 만족(satisfaction) 관계이다. 오늘날 논리학자들은 통상 타르스키의 정의를 모델(model)에 의거한 정의로 받아들이고 있다. 배열 함수는 모델의 한 요소를 이룬다. 하나의 모델은 수많은 배열 함수를 갖는다. 논리식 A가 모델 M에서의 배열 σ에 의해 만족된다는 것을 기호 $\vDash^M_\sigma A$로 표시하고 집합 X에 속한 모든 논리식이 모델 M에서의 배열 σ에 의해 만족된다는 것을 $\vDash^M_\sigma X$로 표시하면 타르스키에 의한 논리적 귀결의 의미론적 정의는 $X \vDash A \Leftrightarrow (M)(\sigma)(\vDash^M_\sigma X \Rightarrow \vDash^M_\sigma A)$에 의해 주어진다. 이러한 의미에서 술어 논리 언어에서의 타르스키 식의 정의를 논리적 귀결의 모델 이론적(model-theoretical) 정의라고 부른다. 그 정의도 모델에 대한 보편 양화에 의거하고 있다. 배치 함수도 단순한 종류의 모델로 생각할 수 있다. 그러므로 (3)과 같은 정의도 모델 이론적 정의로 간주할 수 있다. 타르스키와 같은 논리적 귀결의 개념의 원조는 볼짜노(B. Bolzano)였다.

다. (3)에서와 같은 의미론적 정의를 위해서는 배치 함수를 규정하는 것이 선행되어야 하는데 그 과정에서 각 논리 상항에 대한 의미론을 제시하지 않을 수 없다.[11] 즉 어떤 귀결 개념도 논리 상항의 의미에 의존한다. 통사론적 논리적 귀결의 개념이 의존하는 논리 상항의 의미는 그것과 관련된 추론 규칙에 의해, 혹은 추론에서 그 논리 상항이 하는 역할에 의해 규정되지만 의미론적 귀결의 개념이 의존하는 논리 상항의 의미는 그것에 의해 결합된 복합 문장의 참, 거짓을 결정하는 역할에 의해 규정된다. 그 두 논리적 귀결 관계의 외연이 일치하는 경우 어떤 귀결 개념을 선택해도 무방하지만 그렇지 않은 경우, 예를 들어 의미론적 귀결 관계가 성립함에도 불구하고 통사론적 귀결 관계가 성립하지 않는 경우, 타르스키처럼 의미론적 귀결 개념을 선호하게 될 수도 있을 것이다. 반면에 정인교가 말하는 진리 조건 대신에 주장 가능성 조건을 논리적 귀결 개념과 관련시키려는 정당화주의적 의미론자들은[12] 통사론적 귀결 개념을 선호하게 될 것이다. 이러한 선택을 할 경우에는 자신이 원하는 통사론적 귀결 개념이 의존하는 논리 상항의 의미를 적절하게 규정하는 추론 규칙을 확립해야 하는 부담을 안게 될 것이다.

어떤 이유에서건 논리 상항의 의미를 추론에서의 역할에서 찾으

11) 구체적으로 υ는 다음을 만족하는 S로부터 집합 $\{t, f\}$로의 함수로 정의할 수 있다. 모든 논리식 A와 B에 대해,

(i) $\upsilon(\neg A) \neq \upsilon(A)$
(ii) $\upsilon(A \wedge B)=t \Leftrightarrow \upsilon(A)=t \;\&\; \upsilon(B)=t$
(iii) $\upsilon(A \vee B)=f \Leftrightarrow \upsilon(A)=f \;\&\; \upsilon(B)=f$
(iv) $\upsilon(A \rightarrow B)=f \Leftrightarrow \upsilon(A)=t \;\&\; \upsilon(B)=f$

위의 (i)-(iv)는 각각 연결어 ¬, ∧, ∨, →에 대한 의미론적 정의라고 말할 수 있다.

12) 정인교(2003), 1쪽.

려 할 경우에는 그 의미를 규정하는 추론 규칙이 어떤 것인가 하는 물음에 대한 답변을 제시해야 한다. 정인교에 의하면 "주장(정당화) 조건을 핵심 개념으로 하는 정당화주의적 의미론에서는 도입 규칙들이 논리 상항의 의미를 고정하는 역할을 하고, 주장의 귀결을 핵심 개념으로 하는 실용주의적 의미론에서는 제거 규칙들이 그런 역할을 하게 된다."[13] 이에 반해 "포퍼는 '논리적 귀결' 혹은 '연역 가능성'에 의거해 논리 상항의 의미를 명시적으로 정의하고, 이렇게 정의된 논리 상항의 의미에 의해 추론 규칙들을 정당화 하려 했다."[14] 정당화주의자들이나 실용주의자들과는 달리 포퍼의 노선을 취할 경우 논리 상항의 의미를 결정하기 이전에 논리적 귀결의 개념이 먼저 주어진 것으로 간주되므로 위에서 우리가 한 것처럼 논리 상항의 의미를 먼저 정의하고 그러한 정의에 의거하여 논리적 귀결의 개념을 정의할 수가 없게 된다. 다시 말해 포퍼의 노선에서는 먼저 논리적 귀결의 개념을 미리 규정하지 않으면 안 된다. 그리고 논리적 귀결의 개념을 규정하는 한 가지 방법은 그것에 관한 공리를 제시하는 것이다.

Ⅲ.

이제 논리 상항의 의미를 추론 규칙에 의해 규정하려 할 때 의존하는 논리적 귀결을 '$\vdash$'로 나타낸다고 하자. $\vdash$는 집합 $\wp(S)$의 원소와 S의 원소간의 관계이다. 이러한 관계에 대해 다음과 같이 $\wp(S)$로부터 $\wp(S)$로의 함수 Cn을 대응시킬 수 있다. 즉, 모든 $X \in \wp(S)$와 모든 $A \in S$에 대해

13) 정인교(2003), 10쪽.

14) 정인교(2003), 11쪽.

(4) $X \vdash A \Leftrightarrow A \subseteq Cn(X)$[15)]

Wólenski는 Cn을 귀결 연산(consequence operation), 그리고 ⊢를 귀결 연산자(consequence operator)로 구분해 부르고 있는데,[16)] 양자는 (4)에 의해 상호 정의가 가능하므로 논리적 귀결에 관한 논의에 있어 어떤 것을 사용해도 무방할 것이다. 그렇지만 겐첸을 비롯한 대부분의 논리학자들은 귀결 연산자를 선호하는 반면에 귀결 연산은 주로 폴란드 출신의 논리학자들이 선호하고 있다. 이 글에서는 귀결 연산 Cn을 사용하는 쪽을 선택할 것이다.

벨납은 논리적 귀결 ⊢에 의해 어떤 연결어의 의미 ★를 $A \vdash A \bigstar B$, $A \bigstar B \vdash B$와 같이 규정할 때 ⊢가 이행성(transitivity)을 지닌다고 가정할 경우, 임의의 논리식 A와 B에 대해, $A \vdash B$와 같은 바람직하지 않은 결론이 나오는 것을 방지하기 위해 사전에 주어진 연역 가능성 즉 ⊢의 맥락에 의거하여 연결어의 정의를 제한할 것을 제의하고 있다.[17)] 예를 들어 사전에 주어진 연역 ⊢의 개념이 이행성을 지니지만 임의의 논리식 A와 B에 대해 $A \vdash B$인 것

---

15) '$A \subseteq Cn(X)$'는 $\{A\} \subseteq Cn(X)$를 의미한다. 이것은 곧 $A \in Cn(X)$를 의미하기도 한다. 이하에서 단일한 논리식으로 이루어진 집합과 논리식 자체를 구분하지 않고 이탤릭체 대문자로 표기할 것이다. 그것이 어떤 것을 의미하는가 하는 것은 문맥에 의해 구분하기로 한다. 위의 정의는 $Cn(X)=\{A: X \vdash A\}$와 마찬가지이다.

16) Wólenski(2002), 324쪽. 여기서 ⊢는 위의 (2)와 (3)에서 정의된 논리적 귀결 ⊦ 이나 ⊨을 의미한다고 생각할 필요가 없다. 그것의 의미는 뒤의 Cn에 관한 정의와 (4)에 의해 규정되는 것이며 그에 따라 ⊦ 이나 ⊨으로 해석해도 무방한 것으로 드러날 수도 있다.

17) Belnap(1962), 131쪽. 카르납은 *The Logical Syntax of Language*(Routledge and Kegan Paul, 1937), xv쪽에서 추론 규칙을 임의로 잡는다고 해도 그에 의해 기본적인 논리 기호(fundamental logical symbol)의 어떤 의미를 부여해야 하는가가 결정된다고 말하고 있는데 프라이어(A. N. Prior)가 든 위와 같은 연결어 ★에 대한 정의는 그러한 카르납의 견해가 명백히 잘못임을 보여준다.

은 아닐 경우 위의 ★와 같은 정의를 허용하지 말자는 것이다. 논리적 연결어와 같은 논리 상항의 의미를 논리적 귀결 관계에 의해 정의함에 있어 이러한 제한을 한다는 것은 곧 $\vdash$나 Cn의 개념이 사전에 주어지는 것으로 간주된다는 것을 의미한다. 그러한 개념들을 사전에 제시하는 한 가지 방법은 그것들에 관한 공리를 채택하는 것이다. 통상의 논리 체계의 공리가 논리 연결어들의 개념을 사전에 제시하고 그에 의해 논리적 귀결의 개념을 확립하는 순서를 취한다면 이러한 방법은 반대로 공리를 통해 논리적 귀결의 개념을 정의하고 그렇게 정의된 논리적 개념에 의거하여 논리 연결어의 의미를 정의하는 것에 해당한다.

언어 $\mathcal{L}$에서 $\wp(S)$에서 $\wp(S)$로의 함수 혹은 $\wp(S)$에서의 연산(operation), Cn이 주어져 있다고 하자. 즉 Cn: $\wp(S) \mapsto \wp(S)$라고 하자. 그 함수가 다음 공리를 만족할 때, 그것을 논리적 귀결 관계라고 한다.

S.	$X \subseteq CnX$	(Super-set forming)
M.	$X \subseteq Y \Rightarrow CnX \subseteq CnY$	(Monotone)
A.	$CnCnX \subseteq CnX$	(Auto-preserving)

S는 논리식들의 모든 집합이 자신으로부터 논리적으로 귀결됨을, 그리고 M은 어떤 집합으로부터 논리적으로 귀결되는 것은 모두 그것의 초집합(superset)으로부터 논리적으로 귀결됨을 말하고 있다. A는 어떤 논리식들로부터 귀결되는 논리식들의 집합으로부터 귀결되는 것은 그 논리식들의 집합 자체로부터 논리적으로 귀결된다는 것을 말하고 있다. 위의 세 조건은 논리적 귀결관계가 만족해야 할 기본 조건인 것으로 생각될 수 있는데 그것은 그런데 다음과 같은 다음 하나의 조건으로 압축할 수 있다.

C. $X \subseteq CnY \Rightarrow Y \cup CnX \subseteq CnY$[18)]

S, M, A로부터 C가 나온다는 것을 보이기 위해 $X \subseteq CnY$라고 가정하자. 우선 S에 의해 $Y \subseteq CnY$. 또한 M에 의해 $CnX \subseteq CnCnY$. 이것과 A에 의해 $CnX \subseteq CnY$. 따라서 $Y \cup CnX \subseteq CnY$. 반대로 C가 참이라고 하자. C로부터 곧장 $X \subseteq CnY \Rightarrow Y \subseteq CnY$ & $CnX \subseteq CnY$를 얻을 수 있다. X를 CnY로 잡으면 좌변이 참이 되므로 $Y \subseteq CnY$ & $CnCnY \subseteq CnY$를 얻는다. 따라서 곧장 S와 A를 얻는다. M을 유도하기 위해 $X \subseteq Y$라고 가정하자. S에 의해 $Y \subseteq CnY$이므로 $X \subseteq CnY$. 따라서 C에 의해 $CnX \subseteq CnY$. 그러므로 M도 성립한다.

S, M으로부터 $CnX \subseteq CnCnX$. 이것과 A를 결합하면 $CnX = CnCnX$. 그러므로 $CnX = Cn^{(n)}X$. 따라서 A를 다음과 같은 형태로 강화할 수 있다.

A*. $CnX = Cn^{(n)}X \qquad (n \geq 1)$

위의 S, M, A 이외에 관계 Cn이 만족해야 할 조건으로 타르스키는 다음을 추가하고 있다.

F. $CnX \subseteq \cup(Cn(Y)$: $Y \subseteq X$이고 Y는 유한하다$)$ (Finite)

F는 어떤 논리식이 어떤 집합으로부터 논리적으로 귀결된다면 그것의 유한한 부분집합으로부터 논리적으로 귀결된다는 것을 의미

18) Wójciski는 Wójciski(1988), 23쪽에서 S, M, A를 $X \subseteq CnCnX \subseteq CnX \subseteq Cn(X, Y)$ 하나로 압축하고 있으며 위의 관계가 성립하는 함수, 내지는 일항 연산(unary operation)을 폐포(閉包) 연산(closure operation)이라고 부르고 있다. 그러나 그것은 사실은 한 가지 관계가 아닌, 세 가지 관계 즉 $X \subseteq CnCnX$, $CnCnX \subseteq CnX$, $CnX \subseteq Cn(X, Y)$라고 해야 할 것이다.

한다. 즉 $A \in CnX \Rightarrow (EY \subseteq X)(Y$는 유한하고 $A \in CnY)$. M을 감안할 때 F에서 반대의 포함 관계도 성립함을 알 수 있으며 따라서 F를 다음과 같이 강화할 수 있다.

F*. $CnX = \cup(Cn(Y)$: $Y \subseteq X$이고 Y는 유한하다)

논리적 귀결관계 Cn에 대한 위의 공리 S, M, A 혹은 A*로부터 그 관계에 관한, 아래에서 유용하게 사용될 다음과 같은 몇 가지 정리를 얻을 수 있다.

정리 1. $Cn\varnothing \subseteq CnX$

정리 2. $Cn(X \cap Y) \subseteq CnX \cap CnY$

정리 3. $CnX \cup CnY \subseteq Cn(X, Y)$[19)]

정리 4. $X \subseteq CnY \Leftrightarrow CnX \subseteq CnY$

증명: 우측으로부터 좌측으로의 함축은 S에 의해 곧장 얻어진다. 또한 좌측으로부터 우측으로의 함축은 위의 C를 감안할 때 자명하다.□

정리 5. $CnX \subseteq CnY \Leftrightarrow (Z)(Cn(X, Z) \subseteq Cn(Y, Z))$

증명: 오른쪽에서 왼쪽으로의 함축은 자명하므로 왼쪽에서 오른쪽으로의 함축만을 증명하기로 하자. 이제 $CnX \subseteq CnY$라고 가정하

---

19) 정리 2와 3은 각각 다음과 같이 강화하는 것이 가능할 것이다. 아래에서 $\mathcal{R}$은 $\wp(S)$의 임의의 부분 집합을 가리킨다.

$Cn(\cap\{X: X \in \mathcal{R}\}) \subseteq \cap\{CnX: X \in \mathcal{R}\}$

$\cup\{CnX: X \in \mathcal{R}\} \subseteq Cn(\cup\{X: X \in \mathcal{R}\})$

자. S에 의해 $X \subseteq CnY$. 또한 논리적 귀결 관계의 단조성 M에 의해 $X \subseteq Cn(Y, Z)$. 마찬가지로 S와 M에 의해 $Z \subseteq Cn(Y, Z)$. 그러므로 $X \cup Z \subseteq Cn(Y, Z)$. 다시 M에 의해 $Cn(X, Z) \subseteq CnCn(Y, Z)$. A*에 의해 $Cn(X, Z) \subseteq Cn(Y, Z)$.□

정리 6. $Cn(CnX \cap CnY) = CnX \cap CnY$

증명: S에 의해 $CnX \cap CnY \subseteq Cn(CnX \cap CnY)$. 정리 2로부터 $Cn(CnX \cap CnY) \subseteq CnCnX \cap CnCnY$. 따라서 A*에 의해 $Cn(CnX \cap CnY) \subseteq CnX \cap CnY$. 그러므로 $Cn(CnX \cap CnY) = CnX \cap CnY$.□

정리 7. $X \subseteq CnY \Rightarrow (Y \subseteq CnZ \Rightarrow X \subseteq CnZ)$,

$X \subseteq CnY \Rightarrow (Z \subseteq CnX \Rightarrow Z \subseteq CnY)$

증명: 이 두 관계 가운데 앞의 것을 증명하는 것으로 충분하다. $X \subseteq CnY$이고 $Y \subseteq CnZ$라고 하자. 정리 4에 의해 $CnY \subseteq CnZ$. 따라서 $X \subseteq CnZ$.□

정리 8. $X \subseteq CnY \Leftrightarrow (Z)(Y \subseteq CnZ \Rightarrow X \subseteq CnZ) \Leftrightarrow (Z)(Z \subseteq CnX \Rightarrow Z \subseteq CnY)$

증명: 여기서도 앞의 쌍조건 관계만을 증명하기로 한다. 정리 7에 의해 $X \subseteq CnY \Rightarrow (Z)(Y \subseteq CnZ \Rightarrow X \subseteq CnZ)$. 반대 방향의 함축을 증명하기 위해 $(Z)(Y \subseteq CnZ \Rightarrow X \subseteq CnZ)$라고 하자. 보편 예화에 의해 $Y \subseteq CnY \Rightarrow X \subseteq CnY$. 이것과 S로부터 $X \subseteq CnY$.□

정리 9. $CnX = CnY \Leftrightarrow (Z)(Y \subseteq CnZ \Leftrightarrow X \subseteq CnZ) \Leftrightarrow (Z)(Z \subseteq CnX \Leftrightarrow Z \subseteq CnY)$

증명: 위의 정리 4와 정리 8에 의해 명백하다.□

정리 10. $Y \subseteq CnX$ & $X \subseteq CnY \Leftrightarrow CnX = CnY$

증명: 정리 4에 의해 $Y \subseteq CnX$ & $X \subseteq CnY \Leftrightarrow CnY \subseteq CnX$ & $CnX \subseteq CnY$.□

정리 7은 논리적 귀결관계가 이행적임을 보여주고 있다. 정리 8은 X가 집합 Y의 논리적 귀결인 것은 Y에 속한 모든 논리식을 이끌어낼 수 있는 모든 집합으로부터 X가 논리적으로 귀결되는 경우 또 오직 그 경우에 한한다는 것을, 혹은 X로부터 귀결되는 모든 논리식들이 Y로부터도 귀결되는 경우 또 오직 그 경우에 한한다는 것을 말해 주고 있다. 정리 9는 X의 논리적 귀결과 Y의 논리적 귀결이 동일한 것은 X가 귀결되는 모든 집합으로부터 Y도 귀결되며 또 그 역도 성립하는 경우 또 오직 그 경우에 한하며, 또한 X로부터 귀결되는 모든 집합이 Y로부터도 귀결되고 또한 그 역도 성립하는 경우 또 오직 그 경우에 한함을 말해주고 있다. 다시 말해 어떤 두 집합이 동일한 추론적 역할을 할 경우 또 오직 그 경우에 한해 그 두 집합의 논리적 귀결은 동일하다. 정리 10은 두 집합이 동일한 추론적 역할을 할 경우 또 오직 그 경우에 한해 그 가운데 하나로부터 다른 하나가 논리적으로 귀결됨을 말하고 있다.

위의 조건 S, M, A는 Cn을 언어 $\mathcal{L}$에서의 논리적 귀결관계로 만들기 위한 것이다.[20] 정인교(2003)에서 정인교는 논리적 귀결 관

20) 논리적 귀결 관계 Cn에 관한 공리를 처음으로 제시한 것은 타르스키였다. 타르스키는 Tarski(1956a)에서 Cn에 관한 공리로서 S와 A*, 그리고 F* 이외에 "S에 속하는 어떤 문장 A에 대해 $Cn(A)=S$"라는 것을 추가하고 있다. 그러나 Tarski(1956b)에서는 이 마지막 조건은 제외하고 S와 A*, 그리고 F*만을 제시하고 있다. S와 A*에 F*를 공리로 채택하면 M은 불필요한데 왜냐하면 M은 그 세 가지 공리로부터 정리로 도출할 수 있기 때문이다. 지금 $X \subseteq Y$라고 하고 $A \in Cn(X)$라고 하면 F*에 의해 X의 어떤 유한한 부분집합 X'에 대해 $A \in Cn(X')$. 그런데 $X \subseteq Y$이므로 X'는 또한

계에 의거하여 논리 상항의 의미를 규정하려는 과정에서 그가 말하는 "논리적 관계에 관한 기본적인 조건"으로서 다음과 같은 (R), (M), (T)을 제시하고 있는데, 그것은 Cn의 조건 S, M, A로부터 유도됨을 증명할 수 있다. 정인교는 논리적 귀결을 나타내는 기호로 '/'를 사용하고 있는 바, 그 대신에 여기서 사용하는 기호 '$\vdash$'를 사용하면 (R), (M), (T)를 다음과 같이 기술할 수 있다.[21)]

(R) $A \vdash A$ (모든 논리식은 자기 자신을 함축한다.)
(M) $X \vdash A \Rightarrow X, B \vdash A$ (논리적 귀결 관계는 단조적이다.)
(T) $X \vdash A$ & $Y, A \vdash B \Rightarrow X, Y \vdash B$ (논리적 귀결 관계는 이행적이다.)

위의 관계는 각기 Cn을 사용하여 다음과 같이 고쳐 쓸 수 있다.

(R1) $A \subseteq CnA$
(M1) $CnX \subseteq Cn(X, B)$
(T1) $A \subseteq CnX \Rightarrow (Cn(Y, A) \subseteq Cn(X, Y))$

(R1)과 (M1)은 S, M으로부터 곧장 유도된다. (T1)을 유도하기 위해 $A \subseteq CnX$라고 하자. 위의 정리 4에 의해 $CnA \subseteq CnX$. 여기에 다시 정리 5를 이용하면 $Cn(A, Y) \subseteq Cn(X, Y)$.[22)]

S 관계의 역관계도 성립하는 집합 T, 즉 $T = CnT$인 집합 T를 $\mathcal{L}$에서의 이론(theory)이라고 한다. A*는 모든 $X \subseteq S$에 대해 집합

Y의 유한한 부분집합이기도 하다. 따라서 F*에 의해 $A \in Cn(Y)$. 그러므로 $Cn(X) \subseteq Cn(Y)$.

21) 정인교(2003), 13-14쪽. 정인교는 그러나 (R), (M), (T)가 논리적 귀결 관계에 관한 공리라고 명시적으로 밝히고 있지는 않다.

22) (T1)에서 A는 단일한 논리식을 의미하지만 위에서 제시한 (T1)에 대한 증명은 A를 논리식들의 집합으로 바꾸어도 성립함을 보여준다.

CnX가 이론임을 말하고 있다. $Cn\varnothing$도 $\mathcal{L}$의 이론이다. 정리 1에 비추어 볼 때, 모든 이론 T에 대해 $Cn\varnothing \subseteq CnT=T$. 따라서 $Cn\varnothing$는 이론 가운데 가장 작은 이론이라고 할 수 있다.[23)]

만일 언어 $\mathcal{L}$이 다음을 만족하는 논리 연결어 $*$를 가지고 있다고 하자.

$$B \in Cn(X, A) \Leftrightarrow A * B \in CnX$$

위의 관계로부터 $B \in CnA \Leftrightarrow A * B \in Cn\varnothing$. 전제 A로부터 B가 논리적으로 귀결될 수 있는 것은 논리식 $A * B$가 집합 $Cn\varnothing$에 속하는 경우 또 오직 그 경우에 한한다. 이러한 의미에서 집합 $Cn\varnothing$는 언어 $\mathcal{L}$에서의 논리적 관계를 규정하고 있다고 말할 수 있다. 이 때문에 Woléski는 $Cn\varnothing$를 언어 $\mathcal{L}$의 논리로 정의하고 있다.[24)]

Ⅳ.

위에서 정의된 논리적 귀결 관계를 바탕으로 논리적 연결어를 정의하는 문제를 생각해 보자. 지금 정의하려는 논리 연결어를 n-항의 연결어인 c라고 하고 논리식 $c(A_1, ..., A_n)$를 K로 줄여 쓰기로 하자. 그 논리 연결어를 정의하는 도입 규칙은 A_1, ..., A_n들로부터 구성된 어떤 논리식들의 집합 X로부터 K가 귀결된다는 형태를 취할 것이다. 즉 다음과 같은 형태를 지닐 것이다.

I. $K \subseteq CnX$ ($c(A_1, ..., A_n) \in CnX$ 혹은 $X \vdash c(A_1, ..., A_n)$)

23) 즉, $Cn\varnothing = \cap\{T: CnT=T\}$.

24) Woléski(1995), 25쪽.

I가 $c(A_1, ..., A_n)$의 온전한 정의가 되기 위해서는 I에 의해 $c(A_1, ..., A_n)$가 한 가지로 고정되어야 한다. 그러나 I는 K가 CnX 집합의 한 원소라는 것을 이야기하는 데 불과하며 따라서 $c(A_1, ..., A_n)$를 한 가지로 고정하기 위해서는 K가 CnX의 원소 가운데 특히 어떤 원소인가를 밝히지 않으면 안 된다. 벨납도 연역 가능성에 의거하여 연결어의 의미를 정의하는 과정에서 요구되는 조건으로서 유일성(uniqueness)을 들고 있다.[25] 그가 말하는 유일성이란 연역 가능성에 의거한 정의가 단 한가지의 논리적 역할만을 허용해야 하며 서로 다른 두 가지 추론적 역할을 하는 연결어를 만들어 내서는 안 된다는 것이다. 이러한 요구를 충족하기 위해 논리식의 집합 간의 관계에 의거하여 $c(A_1, ..., A_n)$를 고정하는 방법을 생각할 수도 있다. 지금 $\mathscr{P}(S)$의 원소 X와 Y 사이에 $X \in CnY$가 성립할 때, 즉 $(C \in X)(Y \vdash C)$일 때 $X \leq Y$로 나타내기로 하자. 지금 $\mathscr{P}(S)$의 부분 집합 R의 모든 원소에 대해 $\leq$ 관계에 있는 논리식을 집합 R의 상계(upper bound)라고 하고, R의 상계들의 집합을 U_R로 나타내자. 즉,

(5) $U_R = \{Z \in S: (Y)(Y \in R \Rightarrow Y \leq Z)\}$

(5)에서 R은 $\mathscr{P}(S)$의 원소이지만 R이 단순히 S의 부분 집합인 경우에도 그 상계를 (5)와 같은 방식으로 정의할 수 있을 것이다. 예를 들어 논리식 G가 CnX의 상계(upper bound) 집합 U_{CnX}의 원소라는 것을 다음과 같이 정의할 수 있다.

(6) $G \in U_{CnX} \Leftrightarrow (C)(C \leq X \Rightarrow C \leq G) \Leftrightarrow CnX \subseteq CnG.$

만일 CnX에 속하면서도 U_{CnX}에 속한 원소, 즉 $CnX \cap U_{CnX}$의

25) Belnap(1962), 133쪽.

원소가 단 하나 존재한다면 그것을 $c(A_1, ..., A_n)$로 정의할 때 유일성 조건은 충족된다. 이것은 $K=c(A_1, ..., A_n)$를 다음에 의해 정의하는 것과 마찬가지이다.

I*. $K \subseteq CnX$ & $(C)(C \in CnX \Rightarrow C \in CnK)$ 혹은 $K \subseteq CnX$ & $CnX \subseteq CnK$.

I*는 K가 X로부터 귀결되는 논리식 가운데 가장 강한 논리식임을 말하고 있다. 그런데 I*에서 K가 CnX의 상계에 속한다는 것을 말하는 $(C)(C \in CnX \Rightarrow C \in CnK)$ 혹은 $CnX \subseteq CnK$는 위의 정리 4에 비추어 볼 때 다음과 동치이다.

E. $X \subseteq CnK$ (혹은 $X \in Cnc(A_1, ..., A_n)$)

그러므로 I*는 I&E와 동치가 된다.

I&E. $K \subseteq CnX$ & $X \subseteq CnK$

그런데 정리 10에 의해 I&E는 다음과 동치이다.

Eq. $CnK=CnX$ (혹은 $Cnc(A_1, ..., A_n)=CnX$)

E는 $K=c(A_1, ..., A_n)$의 제거 규칙이 취하는 형식이다. 위의 논의는 $CnX \cap U_{CnX}$의 원소가 단 하나 존재한다면 강화된 도입 규칙 I* 대신에 도입 규칙과 제거 규칙을 합친 I&E에 의해서도 동일한 연결어를 정의할 수 있음을 보여 준다. 그런데 I&E은 바로 포퍼가 취하는 방향이다. 또한 그러한 정의는 Eq와 같은 정의와 대등하다. 위의 정리 9에 비추어 볼 때, 논리 연결어 $K=c(A_1, ..., A_n)$를 위의 I*나 I&E와 같이 정의한다는 것은 $K=c(A_1, ..., A_n)$를 X

와 동일한 추론적 역할을 하는 논리식으로 정의한다는 말과 다름없다.

위에서 우리는

(7) $X \subseteq CnK \Leftrightarrow (C)(C \in CnX \Rightarrow C \in CnK)$

임을 보았다. $X \subseteq CnK$는 즉 K가 CnX 집합의 상계에 속함을 말하고 있다. 마찬가지로 위의 정리 8에 의해 다음을 쉽게 증명할 수 있다.

(8) $K \subseteq CnX \Leftrightarrow (Z)(X \subseteq CnZ \Rightarrow K \subseteq CnZ)$

(8)의 우변은 K가 X를 논리적 귀결로 갖는 모든 집합의 논리적 귀결임을 말하고 있다. (8)에 비추어 연결어 $K=c(A_1, \ldots, A_n)$에 대한 다음과 같은 정의도 정의 I* 및 I&E와 동치임을 알 수 있다.

E*. $X \subseteq CnK$ & $(Z)(X \subseteq CnZ \Rightarrow K \subseteq CnZ)$

E*는 K가 X를 논리적 귀결로 갖는 논리식 가운데 제일 약한 논리식임을 말하고 있다.

위의 논의는 논리 연결어 $K=c(A_1, \ldots, A_n)$를 I*와 같은 도입규칙을 통해 정의한다는 것은 E*와 같은 제거 규칙을 통해 정의한다는 것과 맞먹으며 그것들은 또한 통상적인 도입규칙 I와 E의 결합과 동치라는 것을 보여준다. 그러한 정의의 가장 간단한 예로서 X가 집합 $(A, B)=\{A, B\}$인 경우를 생각할 수 있다. 여기서 A와 B는 개별 논리식이다. 그러한 X로부터 도입되는 2항의 논리 연결어를 '$\wedge$'라고 쓸 때 $K=c(A_1, \ldots, A_n)$는 '$(A \wedge B)$'와 같은 형태가 될 것이다. 이 경우 I는 논리 상항 $\wedge$를 정의하기 위한 도입 규칙

으로서 다음과 같은 형태를 지니게 될 것이다.

I∧. $(A \wedge B) \subseteq Cn(A, B)$

I∧는 $(A \wedge B)$가 A와 B로 이루어진 전제로부터 귀결되는 논리식이라는 것에 의해 ∧의 의미를 정해주고 있다. 반면에 다음의 정의는 A와 B가 모두 $(A \wedge B)$로부터 귀결되는 결론이라는 것에 의해 $(A \wedge B)$의 의미를 규정하려는 제거 규칙이라고 말할 수 있다.

E∧. $(A, B) \subseteq Cn(A \wedge B)$[26]

E∧는 A와 B가 동시에 그로부터 귀결되는 논리식이 $(A \wedge B)$임을 말하고 있다. 그러나 위의 정의는 I∧와 E∧는 그 어느 것도 $(A \wedge B)$의 의미를 일의적으로 고정시키지 못한다. 따라서 일의성을 확보하기 위해서는 다음 세 가지 정의와 같은 것을 선택해야 한다.

I*∧. $(A \wedge B) \subseteq Cn(A, B)$ & $(C)(C \in Cn(A, B) \Rightarrow C \in Cn(A \wedge B))$ 혹은 $(A \wedge B) \subseteq Cn(A, B)$ & $Cn(A, B) \subseteq Cn(A \wedge B)$.

I∧&E∧. $(A \wedge B) \subseteq Cn(A, B)$ & $(A, B) \subseteq Cn(A \wedge B)$

E*∧. $(A, B) \subseteq Cn(A \wedge B)$ & $(Z)((A, B) \subseteq CnZ \Rightarrow (A \wedge B) \subseteq CnZ)$

선언의 연결어 ∨은 $CnA \cap CnB$를 전제로 하는, 즉 X가 $CnA \cap CnB$인 도입 규칙에 의해 정의되는 논리 상항으로 이해할 수 있다. 그런데 위의 정리 6에 의해 $Cn(CnA \cap CnB) = CnA \cap CnB$이므로 ∨를 정의하는 도입 규칙은 보다 단순한 다음과 같은 형태가 될 것이다.

26) 정의 I∧와 E∧는 통상적인 기호 ⊢를 사용해서는 각각 $A, B \vdash (A \wedge B)$ 그리고 $A \wedge B \vdash A$, $A \wedge B \vdash B$와 같이 쓸 수 있을 것이다.

I∨. $(A \vee B) \subseteq (CnA \cap CnB)$

I∨는 물론 다음으로 바꾸어 쓸 수 있다.

$$(A \vee B) \subseteq CnA \ \& \ (A \vee B) \subseteq CnB$$

I∨는 $A \vee B$가 A의 논리적 귀결이기도 하고 동시에 B의 귀결이기도 하다는 것에 의해 ∨의 의미를 정의하고 있다. 반면에 연결어 ∨에 관한 제거 규칙은 다음과 같은 형태가 될 것이다.

E∨. $(CnA \cap CnB) \subseteq Cn(A \vee B)$

E∨는 A로부터 귀결되고 동시에 B로부터 귀결되는 모든 논리식들은 $A \vee B$로부터 귀결됨을 말하고 있다. 연결어 ∨의 경우에도 그 의미를 일의적으로 고정하기 위해서는 위의 정의들을 다음과 같은 세 가지 정의와 같은 것으로 강화해야 한다.

I*∨. $(A \vee B) \subseteq (CnA \cap CnB) \ \& \ (C)(C \subseteq (CnA \cap CnB) \Rightarrow C \subseteq Cn(A \vee B))$

I∨&E∨. $(A \vee B) \subseteq (CnA \cap CnB) \ \& \ (CnA \cap CnB) \subseteq Cn(A \vee B)$

E*∨. $(CnA \cap CnB) \subseteq Cn(A \vee B) \ \& \ (Z)(CnA \cap CnB) \subseteq CnZ \Rightarrow A \vee B \subseteq CnZ)$

위의 논의에 비추어 볼 때, I*∧, I∧&E∧, E*∧는 아래의 Eq∧와 동치이며, I*∨, I∨&E∨, E*∨는 각각 아래의 Eq∧, Eq∨와 동치이다.

Eq∧. $Cn(A \wedge B) = Cn(A, B)$

Eq∨. $Cn(A \vee B) = (CnA \cap CnB)$

정인교가 정인교(2003)에서 제시하고 있는 연결어 ∧과 ∨에 관한 도입 규칙과 제거 규칙은 그곳에서 사용된 기호를 여기서 사용하는 기호로 적절하게 바꾸면 다음과 같이 쓸 수 있다.

(I∧) $A, B \vdash (A \wedge B)$ & $(C)(A, B \vdash C \Rightarrow (A \wedge B) \vdash C)$
(E∧) $(A \wedge B) \vdash A$ & $(A \wedge B) \vdash B$ & $(Z)((Z \vdash A$ & $Z \vdash B) \Rightarrow Z \vdash (A \wedge B))$
(I∨) $A \vdash (A \vee B)$ & $B \vdash (A \vee B)$ & $(C)((A \vdash C$ & $B \vdash C) \Rightarrow (A \vee B) \vdash C)$
(E∨) $(C)((A \vdash C$ & $B \vdash C) \Rightarrow (A \vee B) \vdash C)$ & $(Z)((C)((A \vdash C$ & $B \vdash C) \Rightarrow Z \vdash C) \Rightarrow Z \vdash (A \vee B))$[27]

그런데 귀결 연산자 ⊢ 대신에 귀결 연산 *Cn*을 이용하여 다시 쓰면 위의 (I∧), (E∧), (I∨), (E∨)는 각각 I*∧, E*∧, I*∨, E*∨와 동일한 표현이 된다. 그리고 우리가 본 것처럼 앞의 두 식은 I∧&E∧와, 그리고 뒤의 두 식은 I∨&E∨와 동치이다. 그러므로 (I∧)⇔(E∧)⇔I∧&E∧⇔Eq∧, 그리고 (I∨)⇔(E∨)⇔I∨&E∨⇔Eq∨라고 말할 수 있다. 다시 말해 연산자 ∧와 ∨에 관한 정인교의 정의는 사실은 각각 포퍼가 제안한 정의 I∧&E∧, I∨&E∨와 맞먹는다. 즉 연결어 ∧와 ∨에 관한 한, 포퍼식의 정의, 그리고 정당화주의적 정의, 그리고 실용주의적 정의는 모두 대등하다.

위에서 본 것처럼 위의 결론은 위의 I나 E와 같은 형태의 도입 규칙 혹은 대입 규칙을 통해 정의되는 모든 연결어에 타당하게 성립한다. 그러나 조건의 연결어 →나 부정의 연결어 ¬의 도입 규칙은 그와 같은 형태가 아니다. 연결어 →의 도입 규칙은 통상 다음과 같은 형태를 지닌다.

27) 정인교(2003), 14-15쪽 참조.

I→. $B \subseteq Cn(X, A) \Rightarrow (A \rightarrow B) \subseteq CnX$

I→는 A를 포함한 전제에서 귀결된 결론을 후건으로 하고 A를 전건으로 한 조건문이 그 전제에서 A를 제외한 전제로부터 귀결됨을 말하고 있다. →의 제거 규칙은 보통 도입 규칙처럼 조건문이 아닌 $B \subseteq Cn(A \rightarrow B, A)$와 같은 형태로 제시된다. 그런데 다음 관계를 생각할 때 도입 규칙의 역의 형태로 제시할 수가 있다.

(9) $B \subseteq Cn(A \rightarrow B, A) \Leftrightarrow (X)((A \rightarrow B) \subseteq CnX \Rightarrow B \subseteq Cn(X, A))$

(9)는 $B \subseteq Cn(A \rightarrow B, A)$이 I→의 역과 동치라는 것을 말하고 있다. (9)에서 오른쪽에서 왼쪽으로의 함축은 오른쪽에서 X 대신에 $(A \rightarrow B)$를 대입함으로써 곧장 증명된다. 왼쪽에서 오른쪽으로의 함축을 증명하기 위해 $B \subseteq Cn(A \rightarrow B, A)$이고 $(A \rightarrow B) \subseteq CnX$라고 가정하자. 위의 정리 4에 의해 $Cn(A \rightarrow B) \subseteq CnX$. 따라서 정리 5에 의해 $Cn(A \rightarrow B, A) \subseteq Cn(X, A)$. 그러므로 $B \subseteq Cn(X, A)$.

(9)에 비추어 →의 제거 규칙을 다음과 같은 형태로 제시할 수 있다.

E→. $(A \rightarrow B) \subseteq CnX \Rightarrow B \subseteq Cn(X, A)$

타르스키는 나중에는 제외했지만 Tarski(1956a)에서 Cn에 관한 공리로서 S와 A*, 그리고 F* 이외에 다음과 같은 공리를 채택하고 있다.

Tar. S에 속하는 어떤 논리식 A에 대해 $Cn(A)=S$[28)]

28) Tarski(1956a), 31쪽.

그런데 정인교는 명시적으로 밝히고 있지는 않지만 부정의 연결어를 정의하는 추론 규칙을 제시하기 위해 사실상 타르스키의 위의 공리를 받아들이고 있다. 그리고 공리 Tar를 충족시키는 논리식을 $\bot$로 표기하고 부정의 연결어 $\neg A$를 $(A\rightarrow\bot)$로 정의할 것을 제안하고 있다.[29] 이러한 제안은 $\rightarrow$를 정의하는 도입 규칙, 제거 규칙을 이용하여 $\neg$를 정의하는 도입 규칙, 제거 규칙을 제시할 수 있는 장점을 지닌다. 이제 I$\rightarrow$와 E$\rightarrow$에서 B 대신에 $\bot$를 대입함으로써 다음을 얻을 수 있다.

(10) $\bot\subseteq Cn(X, A) \Rightarrow \neg A\subseteq CnX$
$\neg A\subseteq CnX \Rightarrow \bot\subseteq Cn(X, A)$

그런데 다음이 성립함을 쉽게 알 수 있다.

(11) $\bot\subseteq CnX \Leftrightarrow CnX=S$

오른쪽에서 왼쪽으로의 함축은 자명하다. 왼쪽에서 오른쪽으로의 함축을 증명하기 위해 $\bot\subseteq CnX$라고 하자. 위의 정리 4에 의해 $Cn(\bot)\subseteq CnX$. 이것과 $Cn(\bot)=S$로부터 곧장 $CnX=S$라는 결론을 얻는다.

(11)에 비추어 (10)에서 $\bot\subseteq Cn(X, A)$ 대신에 $Cn(X, A)=S$를 바꾸어 넣음으로써 $\neg$를 정의하는 도입 규칙과 대입 규칙을 다음과 같이 $\bot$를 사용하지 않는 형태로 제시할 수 있다. 그와 같이 $\neg$의 규칙을 제시할 경우 타르스키의 공리 Tar는 필요하지 않게 된다.

I$\neg$. $Cn(X, A)=S \Rightarrow \neg A\subseteq CnX$

29) 정인교(2003), 15쪽.

E¬. $\neg A \subseteq CnX \Rightarrow Cn(X, A)=S$

위의 (9)에 비추어 볼 때, E¬는 물론 $Cn(A, \neg A)=S$와 동치이다.

정인교는 →과 ¬의 의미를 정의하는 도입 규칙과 제거 규칙을 제시하고 있는데 그것을 우리가 사용하는 기호를 사용하여 다음과 같이 기술할 수 있다.

(I→) $(X)(X, A \vdash B \Rightarrow X \vdash A \rightarrow B)$ & $(C)((X)(X, A \vdash B \Rightarrow X \vdash C) \Rightarrow A \rightarrow B \vdash C)$

(E→) $(A \rightarrow B), A \vdash B$ & $(X)(X, A \vdash B \Rightarrow X \vdash A \rightarrow B)$

(I¬) $(X)(X, A \vdash \perp \Rightarrow X \vdash \neg A)$ & $(C)((X)(X, A \vdash \perp \Rightarrow X \vdash C) \Rightarrow \neg A \vdash C)$

(E¬) $\neg A, A \vdash B$ & $(X)(X, A \vdash \perp \Rightarrow X \vdash \neg A)$[30]

(I→)는 $(A \rightarrow B)$가 A와 더불어 B를 귀결하는 논리식들로부터 귀결되는 논리식 가운데 가장 강한 논리식임을, 그리고 (E→)는 A와 더불어 B를 귀결하는 논리식 가운데 가장 약한 논리식임을 말하고 있다. (E→)는 귀결 연산자 대신 귀결 연산 Cn을 사용해서 다음과 같이 고쳐 쓸 수 있다.

$$B \subseteq Cn(A \rightarrow B, B) \ \& \ (B \subseteq Cn(X, A) \Rightarrow (A \rightarrow B) \subseteq CnX)$$

위의 (9)에 비추어 볼 때 우리는 (E→)가 다음과 동치임을 알 수 있다.

(E→)* $((A \rightarrow B) \subseteq CnX \Rightarrow B \subseteq Cn(X, A))$ & $(B \subseteq Cn(X, A) \Rightarrow$

30) 정인교(2003), 15-16쪽 참조. (E→) 변항 X는 $\wp(S)$의 임의의 원소를 가리킨다. 정인교의 본래 (E→)의 정의에서는 S의 임의의 원소를 가리키는 변항 C를 사용하고 있지만 X를 사용해도 무방하다. 그렇게 할 경우 (E→)에서 &의 오른쪽 연언지(conjunct)가 (I→)의 오른쪽 연언지와 동일하다는 것이 보다 명시적으로 드러난다.

$(A \to B) \subseteq CnX)$

그런데 (E→)*는 E→&I→와 동일하다. 이것은 정인교의 정의 (E→)가 E→&I→와 동치임을 말해 준다. 연결어 ㄱ의 의미를 규정하는 정의 (Eㄱ)도 (E→)를 기초하여 얻어진 것이므로 Eㄱ&Iㄱ와 동치라는 결론을 내릴 수 있다. 다시 말해 조건의 연결어 →와 ㄱ의 의미를 각각 제거 규칙 (E→)와 (Eㄱ)에 의해 정의한다는 것은 위의 도입 규칙 E→와 I→ 및 Eㄱ와 Iㄱ에 의해 정의한다는 것과 다름없다.

정의 (I→)를 Cn을 사용해서 다음과 같이 고쳐 쓸 수 있다.

$$(B \subseteq Cn(X,\ A) \Rightarrow (A \to B) \subseteq CnX)\ \&\ (C)((X)(B \subseteq Cn(X,\ A) \Rightarrow C \subseteq CnX) \Rightarrow C \subseteq Cn(A \to B))$$

따라서 (I→)는 Cn을 사용한 다음과 같은 표현과 대등하다.

$$(\mathrm{I}\to)^* \quad \mathrm{I}\to\ \&\ (C)((X)(B \subseteq Cn(X,\ A) \Rightarrow C \subseteq CnX) \Rightarrow C \subseteq Cn(A \to B))$$

그러므로 E→($\Leftrightarrow B \subseteq Cn(A \to B,\ A)$)가 $(A \to B)$가 A와 더불어 B를 귀결하는 논리식들로부터 귀결되는 논리식 가운데 가장 강한 논리식임을 말하는 $(C)((X)(B \subseteq Cn(X,\ A) \Rightarrow C \subseteq CnX) \Rightarrow C \subseteq Cn(A \to B))$와 동치이기만 하면 (I→)와 (E→)는 동치가 된다. 그런데 I→와 후자의 공식은 다음과 각각 다음과 같이 고쳐 쓸 수 있다.

(12) $(A \to B) \in \cap\{CnX\colon\ B \in Cn(X,\ A)\}$

(13) $(C)(C \in \cap\{CnX\colon\ B \in Cn(X,\ A)\} \Rightarrow C \leq (A \to B))$ 혹은 $\cap\{CnX\colon\ B \in Cn(X,\ A)\} \subseteq Cn(A \to B)$

(13)은 $(A \to B)$가 집합 $\cap\{CnX: B \in Cn(X, A)\}$의 상계임을, 그리고 (12), 즉 I→는 $(A \to B)$가 그 집합에 속함을 말하고 있다. 이제 $B \in Cn(A \to B, A)$ (⇔E→)가 참이면 명백히 $\cap\{CnX: B \in Cn(X, A)\} \subseteq Cn(A \to B)$가 성립한다. 따라서 E→ ⇒ $(C)((X)(B \subseteq Cn(X, A) \Rightarrow C \subseteq CnX) \Rightarrow C \subseteq Cn(A \to B))$. 그러므로 (E→) ⇒ (I→). 즉 (E→)는 (I→)를 함축한다. ¬의 경우에도 마찬가지 논의가 성립할 것이므로 (E¬) ⇒ (I¬), 즉 (E¬)는 (I¬)를 함축한다. 그러나 정인교도 논의하고 있듯이 $(C)((X)(B \subseteq Cn(X, A) \Rightarrow C \subseteq CnX) \Rightarrow C \subseteq Cn(A \to B))$ 혹은 $\cap\{CnX: B \in Cn(X, A)\} \subseteq Cn(A \to B)$가 E→ 혹은 $B \in Cn(A \to B, A)$를 함축하는지는 불명확하다.

Ⅳ.

겐첸은 “도입 규칙들이 말하자면 관련된 기호들의 ‘정의’에 해당하며, 제거 규칙들은 궁극적으로 그러한 정의의 귀결에 불과한 것”이라는 수수께끼 같은 말을 남기고 있다.[31] 프라위츠(Dag Prawitz)와 같은 논리학자들이 겐첸의 주장을 살리려고 노력하고 있지만 위에서 본 것처럼 논리 연결어 $K=c(A_1, ..., A_n)$를 $K \subseteq CnX$ 혹은 $K \in CnX$와 같은 형태의 도입 규칙에 의해 정의할 경우, K를 유일한 것으로 고정하기 위해서 별도의 장치를 추가해야 한다. 정인교의 경우 그 장치는 K를 집합 CnX의 원소 가운데 ≤ 관계에 대해 상계인 것으로 잡는다는 것이다. 만일 어떤 논리식 G가 그러한 상계라면 CnX의 모든 원소가 그로부터 귀결되어야 한다. 즉 $CnX \subseteq CnG$이다. 그러나 그러한 G가 존재하는지 또한 존재하더라도 집

---

31) Gentzen(1969), 80쪽.

합 CnX에서 존재하는지는 문제이다. 이러한 문제는 $K\subseteq CnX$인 K가 바로 그러한 G 가운데 하나라고 규정하면 해소된다. 즉 $CnX \subseteq CnK$라고 규정하는 것이다. 위에서 본 것처럼 정인교는 도입 규칙 (I∧)이나 (I∨)에서 연접지로 이러한 규정을 도입하고 있다. 그리고 정리 4에 의해 $CnX\subseteq CnK$는 $X\subseteq CnK$와 동치이며 이것은 통상적인 제거 규칙이다. 그러므로 ∧와 ∨에 관한 한 정인교의 도입 규칙은 (그리고 제거 규칙도) $K\subseteq CnX\&X\subseteq CnK$와 같은 형태라고 생각해야 한다. 그런데 정리 10에 의해 $(K\subseteq CnX\&X\subseteq CnK) \Leftrightarrow (CnK=CnX)$이므로 연결어 ∧와 ∨를 정의하는 정인교의 도입 규칙과 대입 규칙은 모두 다음과 같은 형태라고 생각할 수 있다.

(14) $CnK=CnX$ 혹은 $(C)(K\vdash C \Leftrightarrow X\vdash C)$[32]

구체적으로 정인교의 정의 (I∧)와 (E∧), 그리고 (I∨)와 (E∨)는 다음과 같은 형태의 정의와 동치인 것으로 생각할 수 있다.

Eq∧. $Cn(A\wedge B)=Cn(A,\ B)$
Eq∨. $Cn(A\vee B)=Cn(Cn(A)\cap Cn(B))=Cn(A)\cap Cn(B)$

그렇지만 (14)와 같이 연결어 $K=c(A_1,\ \ldots.,\ A_n)$를 정의함으로써 K가 CnK의 원소이면서도 상계임을 확보할 수는 있지만 그러나 일의성을 확보하는 데까지 성공할 수는 없다.[33] 왜냐하면 K와 동

32) $CnK=CnX$이면 $(C)(K\vdash C \Leftrightarrow X\vdash C)$일 뿐만 아니라 모든 집합 Z에 대해 $K\subseteq CnZ\Leftrightarrow X\subseteq CnZ$이다. 다시 말해 K로부터 귀결되는 모든 논리식이 X로부터 귀결되고 그 역도 성립할 뿐만 아니라 K가 귀결되는 모든 집합으로부터 X(의 모든 원소)도 귀결되고 그 역도 성립한다. 이러한 의미에서 X와 K는 동일한 추론적 역할을 하며, 그러한 의미에서 X와 K는 동치이다.
33) 정인교도 이러한 사실을 지적하고 있다. 정인교(2003), 19-20쪽 참조.

치인 모든 논리식 L이, 즉 $CnK=CnL$인 모든 L이 (14)를 충족하기 때문이다. 예를 들어 $Cn(A\wedge B)=Cn(A, B)$뿐만 아니라 $Cn(A\wedge(A\rightarrow B))=Cn(A, B)$도 성립한다. 그러므로 $CnK=CnX$와 같은 형태의 정의는 연결어를 일의적으로 정의하는 데 실패한다.[34]

그러나 연결어 →나 ¬의 의미를 정의하는 도입 규칙은 $c(A_1, ..., A_n)\subseteq CnX$와 같은 형태로 주어지지 않는다. 그것은 →의 경우 위의 (12)와 같은 형식을 지닌다.

(12) $(A\rightarrow B)\in\cap\{CnX: B\in Cn(X, A)\}$

(12)는 단순히 $(A\rightarrow B)$가 집합 $\cap\{CnX: B\in Cn(X, A)\}$의 원소임을 말하고 있을 따름이다. (13)은 논리식 $(A\rightarrow B)$가 단순히 집합 $\cap\{CnX: B\in Cn(X, A)\}$의 원소일 뿐만 아니라 그것의 상계임을 말하고 있다. 위에서 본 것처럼 그러한 정의를 추가한다고 해도 $(A\rightarrow B)$를 고정시킬 수 있는가 하는 것은 문제가 된다. 그러나 그에 앞서 $\cap\{CnX: B\in Cn(X, A)\}$ 집합에서 관계 ≤에 관한 상계가 존재하는지도 불분명하다. 만일 존재한다면 $(A\rightarrow B)$가 그러한 상계 중에 하나라는 것을 말하는 (13)과 같은 조건을 추가하는 것이 의미가 있을 것이다. 그러나 제거 규칙 (E→)는 통상적인 제거 규칙과 도입 규칙을 결합한 형태인 것으로 파악된다. 따라서 →를 정의하는 제거 규칙을 다음과 같이 쓸 수 있다.

(E→)⇔(E→&I→) $(A\rightarrow B)\subseteq CnX \Leftrightarrow B\subseteq Cn(X, A)$

마찬가지로

34) 그러나 동일한 추론적 역할을 하기만 하면 유일성이 확보된다고 생각하는 벨납의 입장에서는 (14)와 같은 정의도 무방할 것이다.

$(E\neg)\Leftrightarrow(E\neg\&I\neg)$ $\neg A\subseteq CnX \Leftrightarrow Cn(X, A)=S$

만일 연결어 →와 ¬의 경우에 정인교의 도입 규칙과 제거 규칙이 서로 동치가 아닌데 제거 규칙의 경우에는 연결어 ∧나 ∨의 경우와 마찬가지로 통상적인 도입 규칙과 제거 규칙의 결합으로 볼 수 있다면 연결어 →와 ¬의 경우에도 제거 규칙을 정의로 채택하는 것이 바람직 할지 모른다. 그렇다면 모든 연결어의 개념에 대한 정인교의 정의는 사실은 포퍼가 제안한 제거 규칙과 결합 규칙의 결합에 의해 주어지는 것으로 볼 수 있을 것이며 그 결과 포퍼식의 정의가 갖는 장점과 아울러 모든 결함도 같이 안게 될 것이다. 그리고 정인교가 제시한 도입 규칙과 제거 규칙, 그리고 포퍼식의 정의가 모두 대등하다면 그 가운데 특정한 정의만을 자체적으로 정당하다고 볼 수 없을 것이다. 즉 그 모든 정의가 자체적으로 정당하거나 혹은 자체적으로 정당하지 않은 정의라고 해야 할 것이다.

참고문헌

정인교(2003), "자체적으로 정당한 규칙과 논리상항의 의미", 『논리연구』 제6집 제2호, 2003.

Belnap, N.(1962), "Tonk, Plonk and Plink", *Analysis* 22, 130-34쪽

Dosen, K.(1997), "Logical Consequence: A Turn in Style", M. L. Dalla et al.(eds.), *Logic and Scientific Methods*, 289-311쪽

Gentzen, G.(1969), "Investigations into Logical Deduction", M. Szabo(ed.), *The Collected Papers of Gerhard Gentzen*, North-Holland Pub. Co.

Jacquette, D.(ed.)(2002), *A Companion to Philosophical Logic*, Blackwell Publishers

Pogorzelski, W. A. & Wojtylak, P.(2000), "*Cn*-Definitions of Propositional Connectives", *Studia Logica* 67, 1-26쪽

Pollock, J. L.(1990), *Technical Method in Philosophy*, Westview Press

Schroeder-Heister, P.(1984), "Popper's Theory of Deductive Inference and the Concept of Logical Constant", *History and Philosophy of Logic* 5, 79-110쪽

Sundholm, B. G.(2002), "Varieties of Consequence", Jacquette(2002), 241-55쪽

Tarski, A.(1956), *Logic, Semantics, Meta-mathematics*, Hackett Publishing Co.

Tarski, A.(1956a), "On Some Fundamental Concepts of

Metamathematics", Tarski(1956), 30-37쪽
Tarski, A.(1956b), "Fundamental Concepts of of the Methodology of the Deductive Sciences", Tarski(1956), 60-109쪽
Tarski, A.(1956c), "On the Concept of Logical Consequence", Tarski(1956), 409-420쪽
Wójcicski, R.(1988), *Theory of Logical Calculi*, Kluwer Academic Publishers
Wólenski, J.(19995), "Logic and Mathematics", Depauli-Schimanovich 외 편집(1995), *The Foundational Debate: Complexity and Constructivity in Mathematics and Physics*, Kluwer Academic Publishers, 197-210쪽 게재
Wólenski, J.(2002), "Metatheory of Logics and the Characterization Problem", Jacquette(2002), 307-18쪽
Zucker, J. I. & Tragesser, R. S.(1978), "The Adequacy Problem for Inferential Logic", *Journal of Philosophical Logic* 7, 501-16쪽

중앙대학교 철학과
Email: leejk@cau.ac.kr

Logical Cosequence and the Meaning of the Logical Constants

Lee, Jong Kwon

Chung In-Kyo, discussing how the logical constants can be defined in terms of logical consequence in one his recent articles, distinguished three definitions. One is Popper's approach which utilizes both introduction rules and elimination rules, the second the justificationists' definition which uses only introduction rules; and lastly, the pragmatists' approach which involves only elimination rules. In this essay, by making use of an axiomatic system for logical consequence first formulated by Tarski, I will show that in cases of conjunction and disjunction the three definitions are equivalent, and that in cases of connectives for conditional and negation, the pragmatists' definition is equivalent to Popper's one.

【Key Words】 logical consequence, proof theory, logical constant, Inkyo Chung, Tarski, Popper