第５回目

１１月３日「計算論理学」補足教材

証明の構成の手順について補足します

- 第１回レポート課題では特に「かつ」と「ならば」の場合の証明構成について問いを設定しました。これに関することを補足します。
- 否定については「ならば」の場合とほぼ同じです。（否定の導入規則と消去規則は、「ならば」の導入規則と消去規則の特別な場合であると注意したことからも分かると思います。）
- 「または」の消去規則は、数学で「場合分けの証明」と言われている形です。例題などで学習し、「または」の入っている練習問題に挑戦してみてください。
- １９ページの規則11(矛盾の消去）と規則12(二重否定の消去)については次回補足します。

先々週から先週にかけていくつかの例を出して、∧-I規則∧-E規則、→-I規則、→-E規則（それぞれ、「かつ」の導入規則、「かつ」の消去規則、「ならば」の導入規則、「ならば」の消去規則と呼ぶことにします）を組み合わせてできる「証明」について学習しました。基本教材には例えば次のような例があります。この例では、最後推論規則である「ならば」の導入規則の使用時に、二つの同じ内容の「前提」にカギカッコがつけられ、もはや前提ではないことになっています。

2. $(A \wedge (A \to B)) \to B$

$$\dfrac{\dfrac{\dfrac{[A \wedge (A \to B)]^1}{A}\ \wedge\text{-}E \quad \dfrac{[A \wedge (A \to B)]^1}{A \to B}\ \wedge\text{-}E}{B}\ \to\text{-}E}{(A \wedge (A \to B)) \to B}\ \to\text{-}I,1$$

Bであると結論する証明の部分までは常に「A でありかつ（AならばB）である」は前提となっています。しかし最後の「ならば」の導入規則によって、結論が「（ A でありかつ（AならばB）である）ならばBである」となったときには、新しき導入した「ならば」が条件文の形をしていて「ならば」の直前の条件となる内容に、これまでの証明の各段階で「前提」としていた部分が取り込まれるので、もはや前提は消してよいこととなり、カギカッコを付けて、前提ではなくなった位置を番号１で示しています。（「風が吹けば桶屋が儲かる」の証明も参照）

2. $(A \wedge (A \to B)) \to B$

$$\dfrac{\dfrac{\dfrac{[A \wedge (A \to B)]^1}{A}\ \wedge\text{-}E \quad \dfrac{[A \wedge (A \to B)]^1}{A \to B}\ \wedge\text{-}E}{B}\ \to\text{-}E}{(A \wedge (A \to B)) \to B}\ \to\text{-}I,1$$

では、例題２のように問題が与えられたとき、どのような手順で問題解決していけばよいでしょうか。（この例題のように、使ってよい前提論理してが問題文で示されていない場合は、前提なしの証明が求められていると理解してください。）
実際、この問題は単純なので、すぐにこの証明が頭に思い浮か人がいることと思います。それはそれで結構です。
第１回レポート課題でも証明を作り、どのような手順で作ったかも説明してもらう設問が出ています。

2. $(A \wedge (A \to B)) \to B$

$$\dfrac{\dfrac{\dfrac{[A \wedge (A \to B)]^1}{A}\ \wedge\text{-}E \quad \dfrac{[A \wedge (A \to B)]^1}{A \to B}\ \wedge\text{-}E}{B}\ \to\text{-}E}{(A \wedge (A \to B)) \to B}\ \to\text{-}I,1$$

論理式で与えられる問題の問題解決には、一般には、必ずしも「このやり方に従って証明を作ってください」というような定石は一般にはないのですが、命題論理に限ると定石があります（命題論理言語より表現力の豊かな論理言語の証明検索は、「自動定理証明」や「論理型プログラミング言語」の計算モデルとなります。）

2. $(A \wedge (A \to B)) \to B$

$$\dfrac{\dfrac{\dfrac{[A \wedge (A \to B)]^1}{A}\ \wedge\text{-}E \quad \dfrac{[A \wedge (A \to B)]^1}{A \to B}\ \wedge\text{-}E}{B}\ \to\text{-}E}{(A \wedge (A \to B)) \to B}\ \to\text{-}I,1$$

（証明検索の定石）：「かつ」と「ならば」だけの場合。
１．もし使ってよい前提の論理式のリストが与えられていたら、それをストックしておく
２。証明が求められている論理式を下から上へ順にI－規則によって分解していく。　特に「ならば」のI－規則で分割時に、新しい前提を前提リストのストックに入れる。これ以上I-規則で分解できなくなるまで続ける。
３．使ってよい前提のストックの論理式とE-規則（消去規則（および必要に応じてI－規則も）を組み合わせて、２で途中までできている証明の前提を証明する。

2. $(A \wedge (A \to B)) \to B$

$$\dfrac{\dfrac{\dfrac{[A \wedge (A \to B)]^1}{A}\ \wedge\text{-}E \qquad \dfrac{[A \wedge (A \to B)]^1}{A \to B}\ \wedge\text{-}E}{B}\ \to\text{-}E}{(A \wedge (A \to B)) \to B}\ \to\text{-}I,1$$

下の例題２の論理式をゴール（目標）して証明検索を開始。。。
１．特に前提として使ってよい論理式は与えられていないので、前提のストックは空集合として証明検索を開始する。
２．まず→-I(「ならば」の導入規則）を下から上に向かって分解すると、新たな目標（サブゴール）はB（つまり、このBを証明することが次のステップの目標とする）。→Bの左辺の論理式が前提としてよいストックに入る。Bはこれ以上I-規則で分解できないので、３に移る。
3.この前提を今度は上から（主に）E-k規則で、現在の目標（サブゴール）Bに向かって証明することを目標に規則の組み合わせを工夫する。

2. $(A \wedge (A \to B)) \to B$

$$\dfrac{\dfrac{\dfrac{[A \wedge (A \to B)]^1}{A}\ \wedge\text{-}E \qquad \dfrac{[A \wedge (A \to B)]^1}{A \to B}\ \wedge\text{-}E}{B}\ \to\text{-}E}{(A \wedge (A \to B)) \to B}\ \to\text{-}I,1$$

この例題も同様の手順でできます。
1.使ってよい前提のストックは最初は空・
２・下から順番に３度→-I規則により分解しCに至る。それに伴い、３つの論理式が前提のストックに入る。Cはこれ以上I-規則で分解できないので、３に津済む。
３．前提として使ってよいことが分かっていいる前提のストックの３つの論理式の組み合わせを考えて、主にE-規則を使ってCを結論とすり証明をつくり、すでに２より、**Cが証明されれば求める例題３の論理式が証明できることが分かっているので、これにより問題は解決する。**

3. $(A \wedge B \to C) \to (A \to (B \to C))$

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]^2 \quad [B]^1}{A \wedge B}\,\wedge\!-I \qquad [A \wedge B \to C]^3}{C}\,\to\!-E}{B \to C}\,\to\!-I,1}{A \to (B \to C)}\,\to\!-I,2}{(A \wedge B \to C) \to (A \to (B \to C))}\,\to\!-I,3$$

「又は」と「否定」の規則を使ってみよう。
前回までの２回にわたって説明だけはしていた「又は」の規則も使った例題が以下です。手続き２で下から上に向かって分解し、Cに至る。ここで２つの前提が前提ストックに入るわけです。手続き３では是前提ストックをうまく使って、上から下のCを結論とする証明を作ります。￥または」の入った前提があるときには、その「又は」をなる初めに（下の方で）消去規則を適用し、場合分けの証明）下では、A前提としてCを証明する場合とBを前提にしてCを証明する場合に持ち込みます。どちらもCに至る証明となれば、AとBという暫定的前提はカッコを付けて、それら２つの前提なしに「AまたはB]のどちらなのかわからなくても、「AまたはB」からCだと結論し、手続き２の部分と接続し、全体の証明が完結します。

$$(A \to C) \wedge (B \to C) \to (A \vee B \to C)$$

$$\dfrac{\dfrac{\dfrac{[A \vee B]^2 \quad \dfrac{[A]^1 \quad \dfrac{[(A \to C) \wedge (B \to C)]^3}{A \to C} \wedge\text{-}E}{C} \to\text{-}E \quad \dfrac{[B]^1 \quad \dfrac{[(A \to C) \wedge (B \to C)]^3}{B \to C} \wedge\text{-}E}{C} \to\text{-}E}{C} \vee\text{-}E,1}{(A \vee B \to C)} \to\text{-}I,2}{(A \to C) \wedge (B \to C) \to (A \vee B \to C)} \to\text{-}I,3$$

基本教材17-18ページの規則1から10を使って、どんな証明が作れるか

- このことを試してみてください。

出題している第１回レポート課題にある証明作成問題は、規則１から規則５を使えば解答できるものだけです。

今回の土足教材で、証明を作成する定石的Strategy(手順・方略・戦略）を紹介しました。レポート課題の証明作成の設問に対しては、今回紹介した手順に従わなくても従ってもどちらでも結構です。自分なりの解き方を説明していただければよいです。

論理型プログラミング・定理自動証明とのかかわりについての補足ング t

- 本授業では直接扱いませんが、プロローグをはじめとする「宣言型」プログラム言語があります。それらの根本的な考え方は、

１．前提として使ってよい論理式（それに相当する表現）の集合をProgramとみる。

２．求めようとする論理式（の形）をQueryとみる。

３．あるStrategyに従って証明を作ること（または証明検索）の過程をプログラムの実行とみる。

ということです。証明が完成するとQueryに対して付加的情報が得られる仕組みになっています。

証明検索と認知・認識プロセス

- Wこの授業で採用している論理推論規則のシステムは、「自然演繹体系」Natural Deductionというすステムです。Hilbert学派のGentzenが提案した命題論理に対する最もエレガントな体系の一つです。
- 多これまでに学んだ推論規則1-10の形からも分かるように、各論理接続子に対して、導入規則と消去規則がそれぞれの論理接続子ごとに与えられています。
- このことから次のような「**証明の正規化**」が可能となります。
- 「証明の正規化」は本授業の計算モデルの核心をなすものです
- 。

(参考) 自然演繹体系の基本定理： 正規化定理 (Proof-Normalization)

証明の中で次のような形で現れる論理式 $A \wedge B$, $A \to B$, $\neg A$ を**極大論理式 (maximal 論理式)** と呼ぶ. 即ち極大論理式とは, (自然演繹の) 証明中に導入規則を用いて現われ, すぐその後に消去規則で消えてしまうような, いわば不用な論理式である.

$$\dfrac{\dfrac{A \quad B}{A \wedge B}\ \wedge\text{-}I}{A}\ \wedge\text{-}E$$

$$\dfrac{\dfrac{A \quad B}{A \wedge B}\ \wedge\text{-}I}{B}\ \wedge\text{-}E$$

$$\dfrac{\dfrac{A}{A \vee B}\ \vee\text{-}I \qquad \begin{matrix}[A] & [B] \\ \vdots & \vdots \\ C & C\end{matrix}}{C}\ \vee\text{-}E$$

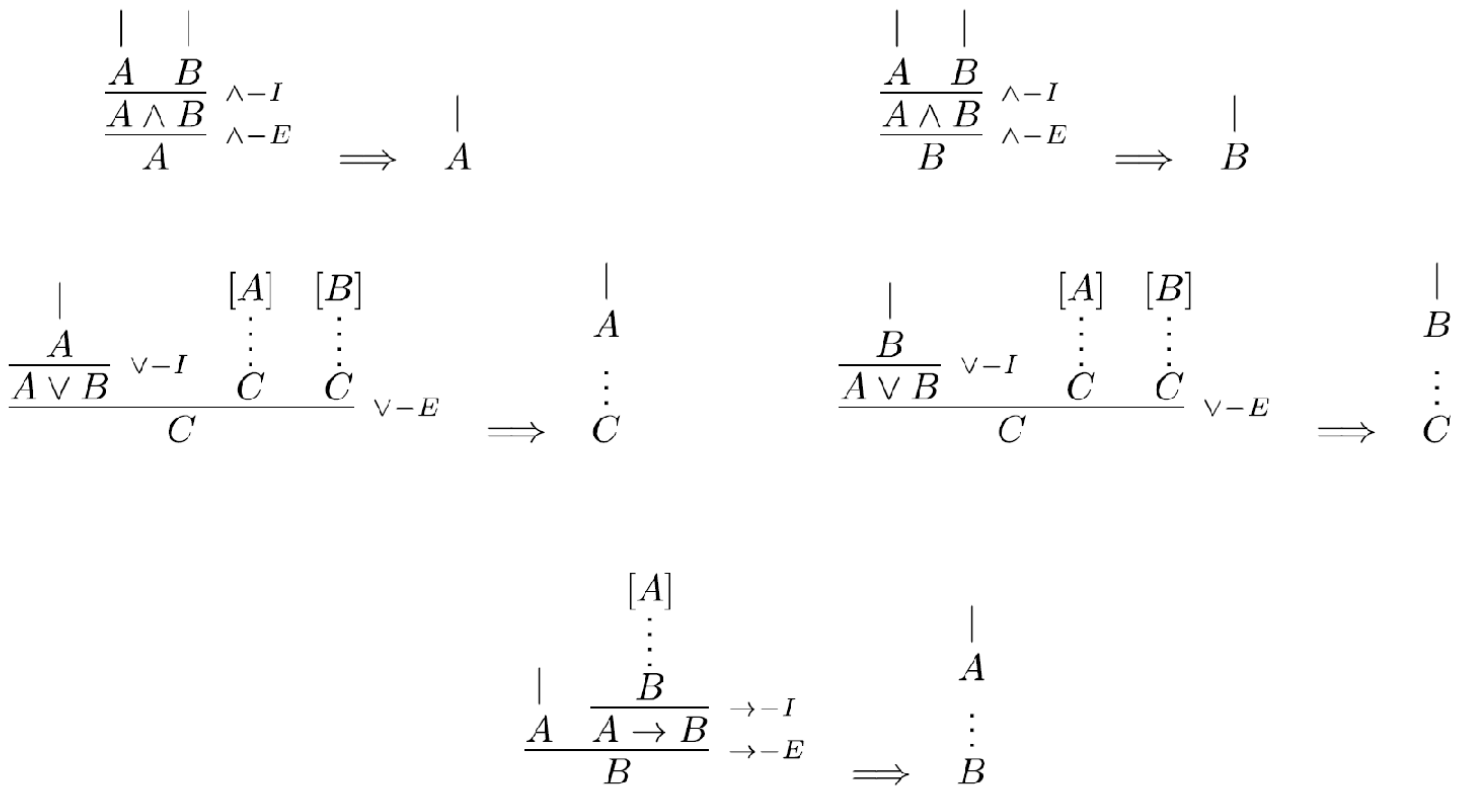
$$\dfrac{\dfrac{B}{A \vee B}\ \vee\text{-}I \qquad \begin{matrix}[A] & [B] \\ \vdots & \vdots \\ C & C\end{matrix}}{C}\ \vee\text{-}E$$

$$\dfrac{A \quad \dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \to B} \; \to\!-I}{B} \; \to\!-E$$

$$\dfrac{A \quad \dfrac{\begin{array}{c}[A]\\ \vdots \\ \bot\end{array}}{\neg A} \; \neg\!-I}{F} \; \neg\!-E$$

証明の中に現れる極大論理式の部分を部分的に書き換えて証明の複雑さを小さくする書き換え規則：簡約規則（ラムダ計算論でベータ規則と呼ぶ）

Reduction rules



証明の中に現れている極大論理式を簡約則で簡約する例

$$\dfrac{\dfrac{\dfrac{\dfrac{[A]^2 \quad [B]^1}{A \wedge B}\wedge I}{A}\wedge E}{B \to A}\to I,\,1}{A \to (B \to A)}\to I,\,2$$

ここで $A \wedge B$ が極大論理式になっています。簡約規則を適用すると、

$$\dfrac{\dfrac{[A]^1}{B \to A}\to I}{A \to (B \to A)}\to I,\,1$$

注意

- 前ページで、書き換え後の証明で「ならば」の導入推論規則では、もともと「ならば」の前提がないのに「ならば」を導入しています。これも「ならば」の導入規則の正しい適用例と考えてください。

例えば、Aが「関数F(x)の値は常に正数である」と証明された時、

Bとして「Fの定義域を100までとする」としたとしても、

「Fの定義域が100までとすれば、関数F(x)の値は常に正数である」が論理的に正しいと考えられます。

（時間に余裕のある人は、このようなことが成り立たない日常的な文があるかどうか考えてみてください。）

ロジックから計算論へ(これまでの補足もしながら、この話題にも入ります。）論理式＝タイプ、証明＝プログラム、証明の正規化＝プログラムの実行

第4章　タイプ付ラムダ計算 (Typed λ-calculus)

計算機科学や情報科学の基本的計算モデルの重要な枠組の１つとして**ラムダ計算体系**と呼ばれる計算モデルの形式体系がある．特にプログラム言語等で重要な**タイプ** (**型**とも呼ばれる) の概念が組み込まれたラムダ計算体系は**タイプ付 (型付き) ラムダ計算体系**と呼ばれる．

本節では第 2 章で導入した自然演繹体系による命題論理と対応させながらタイプ付ラムダ計算体系を導入する．ラムダ計算言語における項 (これは**ラムダ項**と呼ばれる) はアルゴリズムを表現する表現力を持ち，特にタイプ付関数型プログラミング言語のプログラムの抽象的 (数学的) 表現と考えることができる．本節では，この (タイプ付) ラムダ項は命題論理の (自然演繹の) 証明構造と同一視することができ，又，タイプ (型) は命題論理の論理式と同一視することができることを明らかにする．又，ラムダ項で表されるアルゴリズム (プログラム) の計算過程 (実行過程) は，命題論理の証明論 (文章論) の節で述べた証明の正規化過程と同一視することができることも明らかにする．

上級者へ

- 基本教材にある「弱正規化定理」の証明を学習してください。
- 教材で言及している「強正規化定理」の証明にチャレンジしてみてください。
- より簡単な設定で「強正規化定理」を証明したければ、使われる論理結合子の種類を限定して考えてみてください。

型理論とコンピュータ理論の誕生の経緯

- 型や階層構造を考慮に入れないカントールの素朴集合論に矛盾が発見された（１９０２年：ラッセルノパラドクス）。数理科学が集合論内で定式化できると考えられていたので大事件であった。

ラッセルノパラドクス[“自分自身をメンバーとして含まないものすべてからなる集合”を集合Rと呼ぶこととする。この時、RがRのメンバーとすると矛盾し、メンバーでないとしても矛盾する。]

このパラドクスの解決策としてラッセルは型付き論理を提唱した。

一方、チャーチはこのラッセルの体系に計算論的要素を導入しようとして論理と計算論の融合理論へ向かった。チャーチとチューリング（及びポスト）がほぼ同時に「すべての計算ができる計算機（の論理ー数理）理論モデル」を完成させた（1936年）（チャーチの型なしラムダ計算、チューリングマシン誕生）。授業でも紹介します。