

THEOREMA を用いた自動定理証明について

田中寛人, 高橋正

HIROTO TANAKA, TADASHI TAKAHASHI *

甲南大学知能情報学部

DEPARTMENT OF INTELLIGENCE AND INFORMATICS, KONAN UNIVERSITY

1 はじめに

自動定理証明 (英: automated theorem proving, ATP) とは自動推論 (AR) の中でも最も成功している分野でありコンピュータプログラムによって数学的定理の証明を発見することである。ベースとなる論理によって、定理の妥当性を決定する問題は簡単なものから不可能なものまで様々である。THEOREMA は、オーストリアの RISC-Linz 研究所で開発された自動定理証明ソフトウェアであり、日本では利用者が少なく、その使用法に関して日本語の文献がほとんどない。

本稿では、2章で対象とする命題論理の枠組みを示し、3章で THEOREMA の使用方法を示す。そして、4章において2章で示した命題に対し、THEOREMA を使用する手順を日本語で示し、推論規則を用いた初等的な命題論理の証明を、THEOREMA を使用して示す。

2 割合に関する問題設定

2.1 命題論理における推論過程の記号化

本章では、[1] に基づき、割合に関する命題論理について述べる。

(1) 推論規則及び推論法則

x, y, z, a, b, c, d は 0 以上の値をとる変数とする。また、 $f(x)$ は、 $x = y, x > y, x < y$ のいずれかとする。この変数 x を含む $f(x)$ を式と呼ぶ。ここでは変数の演算を認め、その演算規則は推論規則に適用できるものとする。そこで、変数の演算に関する単位元、零元、反射律、対称律、推移律、推論規則、推論法則を以下の表に示す。

単位元	$x \times y = y \times x = x$ が成り立つとき、 y を単位元とする。また、 $y = 1$ と記す。			
零元	$x + y = y + x = x$ が成り立つとき、 y を零元とする。また、 $y = 0$ と記す。			
推移律	$\frac{x = y \quad y = z}{x = z}$	$\frac{x > y \quad y > z}{x > z}$	$\frac{x < y \quad y < z}{x < z}$	
	$\frac{x > y \quad y = z}{x > z}$	$\frac{x = y \quad y > z}{x > z}$	$\frac{x < y \quad y = z}{x < z}$	$\frac{x = y \quad y < z}{x < z}$

*takahasi@konan-u.ac.jp

推移律	$a * b = c$ のとき、 $\frac{f(a * b)}{f(c)} = \frac{f(c)}{f(a * b)}$ を認める。	
==	$\frac{a = b \quad c = d}{a * c = b * d}$	* : +, -, ×, ÷ のいずれか
>= 1	$\frac{a > b \quad c = d}{a * c > b * d}$	$\frac{a < b \quad c = d}{a * c < b * d}$ * : +, -, ×, ÷ のいずれか
	$\frac{a = b \quad c > d}{a * c > b * d}$	$\frac{a = b \quad c < d}{a * c < b * d}$ * : +, × のいずれか
	$\frac{a * c > b * d}{a * c > b * d}$	$\frac{a * c < b * d}{a * c < b * d}$
>= 2	$\frac{a = b \quad c > d}{a * c < b * d}$	$\frac{a = b \quad c < d}{a * c > b * d}$ * : -, ÷ のいずれか
>> 1	$\frac{a > b \quad c > d}{a * c > b * d}$	$\frac{a < b \quad c < d}{a * c < b * d}$ * : +, × のいずれか
>> 2	$\frac{a > b \quad c > d}{a * d > b * c}$	$\frac{a < b \quad c < d}{a * d < b * c}$ * : -, ÷ のいずれか
>> 2	$\frac{a > b \quad a < b}{b < a \quad b > a}$	

(上記の演算において、 $c \neq 0, d \neq 0$ の場合に ÷ を適応する。)

(2) くじを1本引くという試行に関する記号化「当たりくじ」と「はずれくじ」の入ったくじからくじを1本引くという試行における事象、この試行における根元事象の個数、確率に関する記号を以下の表に示す。なお、上述の変数 x は $n(X), n(Y), n(S), P(X), P(Y)$ のいずれか及びそれらから構成されるものとする。

くじを1本引くという試行における事象	X	当たりくじが出る事象
	X_A	くじAにおいて当たりくじが出る事象
	X_B	くじBにおいて当たりくじが出る事象
	Y	はずれくじが出る事象
	Y_A	くじAにおいてはずれくじが出る事象
	Y_B	くじBにおいてはずれくじが出る事象
	S	全事象
	S_A	くじAに関する全事象
	S_B	くじBに関する全事象
くじを1本引くという試行における根元事象の個数	$n(X)$	当たりくじの本数
	$n(X_A)$	くじAの当たりくじの本数
	$n(X_B)$	くじBの当たりくじの本数
	$n(Y)$	はずれくじの本数
	$n(Y_A)$	くじAのはずれくじの本数
	$n(Y_B)$	くじBのはずれくじの本数
	$n(S)$	くじの総数
	$n(S_A)$	くじAの総数
	$n(S_B)$	くじBの総数

くじを1本引くという試行における確率	$P(X)$	当たる確率
	$P(X_A)$	くじ A において当たる確率
	$P(X_B)$	くじ B において当たる確率
	$P(Y)$	はずれる確率
	$P(Y_A)$	くじ A においてはずれる確率
	$P(Y_B)$	くじ B においてはずれる確率
	$P(S)$	全事象の確率
	$P(S_A)$	くじ A に関する全事象の確率
	$P(S_B)$	くじ B に関する全事象の確率

3 THEOREMA

ここでは、Theorema 2.0: ファースト・ツアー [2] で示されている Theorema の使用方法について説明する。

Theorema 2.0 のユーザーインターフェース：

Theorema ノートブックは、数式テキストが書き込まれた Mathematica ノートブックである。Theorema ノートブックは Theorema 固有のスタイルシートのいずれかを使用していなければならない。

Theorema 2.0 コマンド：

グラフィカル・ユーザーインターフェースの一つで、ノートブックに書き込まれている素材との様々なインタラクションに対応している。

Theorema 2.0 における操作は、構造化した数式を含むドキュメントを Theorema ノートブックに書き込み、Theorema コマンドが対応するアクション（例えば、定理の証明）を使用して、ノートブックの様々な部分に対し、特定の演算を実施することを意味する。Theorema が行う基本的なアクティビティは、カレントセッションの設定、証明、計算、解決、そしてシステムのカスタマイズである。このようなアクティビティは Theorema コマンドの左端で選択できる。

アクティビティに応じて、コマンドの上部には、そのアクティビティの枠内で実施される関連アクションが表示される。上の図は、“Session”アクティビティのアクションとして、カレントセッションの構成 (“compose”)、カレントセッションの検証 (“inspect”)、そしてアーカイブ (“archives”) での作業があることが分かる。

セッションの設定：

新規ノートブックを作成するか、または既存のノートブックを開くときは、常に、“compose”アクション内のボタンを使用して作業しなければならない。ノートブックを開く、または作成するために、Mathematica の標準的手順を使用する事はできない。

セッションの構成：

Theorema により作業する際は常に、何らかの方法で数式を処理する。ある数式を Theorema に認識させるためには、この数式を Theorema 環境（定義、定理、補助定理など）の枠内で評価しなければならない。Theorema 環境の作成は、“compose”アクション内にあるボタンを使用して実行することが必要である。

次に、ボタンをどれか一つ押すと、選択されたタイプの環境用のテンプレートが挿入される。例えば、PROPOSITION のテンプレートを開くと、次のようになる（図 1）。

環境はセルスタイル（環境オープン、環境ヘッダなど）が異なるセルの集まりから構成されている。スタイルは単に目に見える外観だけでなく、環境の機能にとっても重要である。



図 1: 環境用のテンプレート

(a) 環境ヘッダは環境のタイプ、例えば、"Proposition"を示し、省略記号 (...) は、その環境に名前を付けることができることを示す。後述するように、名前はオリエンテーションで役立つ。ヘッダの内容は全て書式が自由で、構文上の規則もない。また、環境のタイプは自由に選択できる。ボタンはあくまで提案を示しているにすぎない。希望する環境がここに列挙されていなければ、? ボタンを使用する。環境のタイプは、どんな意味も示していない。単なる見た目の問題である。数式を定義環境で書いても、定理環境で書いても、ノートブックを読む人にとっては違いがあるかもしれないが、システムにとって違いはない。

(b) 数式自体は、この数式専用の 1 セルの中に収まっている。数式セルのセルスタイルは"FormalTextInputFormula" でなければならない。1 つの環境に複数の数式セルがあっても構わない。

(c) 全ての数式には、後でその数式を参照できるようにラベルが付けられていなければならない。数式ラベルは、そのセルのセルタグから割り振られ、"???" は、ラベルがまだついていないことを示している。Mathematica の標準メニューを使用してセルタグを編集し、"???" を希望のラベルと置き換える。ラベルを気にしないのであれば、"???" のままにしておいても構わない。新しいラベルが自動的に作成される。

(d) 環境のグループ化は重要である。環境ヘッダはエンドマーカまでを一つのグループとする。環境の始まりには環境オープンセル、終わりには環境クローズセルがある。これらを削除してはならない。

(e) 環境を削除するには、"×"をクリックする。Mathematica 自体の「切り取り」や「削除」を使用しない。ここでカーソルを数式セルの"■"の所に置き、数式を入力する。"compose"アクション内のボタンまたは個々のキーボードショートカットを使用する。Theorema 言語で書かれた言語構成体の大半に、式テンプレートをノートブックに貼り付けるためのボタンが提供されている。ボタンは言語カテゴリ (Logic, arithmetic, Sets など) に応じて分類され、各ボタンにはツールチップ (吹き出し) があり、その式の各部分とともに、さらに最も大事な、それぞれの式のキーボードショートカットを説明している。どのボタンにも対応するショートカットがある。原則的に、数式も Theorema ノートブックに貼り付け可能である。ただし、決められた通りに書かれていない数式を Mathematica に貼り付けると、構文的に不適切な式になってしまう危険性がある。そのため階層的数式入力が必要である。数式を Theorema に認識させるために、この数式を評価しなければならない (Mathematica 入力と同様)。数式が評価されると、Cell Tags で指定されたラベルが、数式ラベルとして右端に配置される。個々の数式の削除用として、数式セルにも"×"がある。環境の場合と同様、Mathematica 自体の「切り取り」や「削除」を使用できない。

証明：

数式を Theorema に入力すれば、この数式を証明する準備は完了する。「証明」とは、あるステートメントがある仮定の下で真であるかを論理的に立証する作業である。検討しているステートメントを証明目標、仮定をナレッジ (ベース) と呼ぶ。ここで、 $\{(P \vee Q) \wedge (P \Rightarrow Q)\} \Leftrightarrow Q$ を証明したいとする。例えば次のようになる (図 2)。

まず、"Prove"アクティビティに移る。ここで、何かを証明するために必要なアクションとしては、目標の選択、ナレッジベースの選択、組込み演算の宣言、プルーバ (証明器) の設定、証明タスクのシステムへ

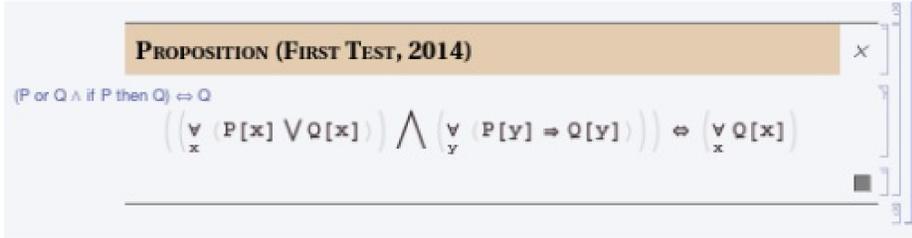


図 2: 命題

の実行依頼、そして最後に、得られた証明の検証からなることが分かる。上部のアクションは左から右に実行しなければならない。これにより証明プロセスを進めることができる。

証明目標の選択：

”goal”アクションで示されるように、証明目標を選択するには、証明の対象となる数式が含まれる、Theorema ノートブック内のセルを選択するだけである。コマンドは即座に選択された数式を表示する。

選択箇所を別の数式に動かすと、選択される証明目標も変わる。”OK, next ...”ボタンを押すと、証明目標が確定する。その後、ノートブック内で何を選択しても証明目標は変わらない。目標を変更するには、別の数式セルを選択し、”OK, next ...”をもう一度押すことになる。目標を確定すると、自動的に”knowledge”アクションに移る。

ナレッジベースの選択：

証明器が使用するナレッジベースの作成に、ナレッジブラウザが役立つ。マウスをクリックするだけで、ナレッジベースに収める数式のグループを選択したり、選択解除することができる。”OK, next ...”ボタンを押すと、利用できる組込み演算を選択する次のアクションへ進む。もしくは、上部のアクションメニューから、次のアクションを選ぶこともできる。

利用可能な組込み演算の選択：

Theorema の組込み演算が開始するのは、数式を簡略化するために、計算が適用されるときである。組込みブラウザのレイアウトと機能は、上記”knowledge”アクションの所で示したナレッジブラウザと似ている。チェックボックスを使って、演算または演算のグループ全体を、Theorema システムにおけるその組込み関数と関連付けることができる。証明時、新しい数式が生成されるとすぐに、計算により数式の簡略化を適用する。デフォルトでは、証明時に組込み関数が与えられている演算は、ほとんどない。Theorema ノートブックで上位レベルの計算を実行するとき、デフォルト値は異なる。下記の例では、組込み演算は機能していない。デフォルト値をそのままにしておくことができる。”OK, next ...”ボタンをクリックすると、証明器を設定する次のアクションへ進む。または、上部のアクションメニューから次のアクションを選ぶこともできる。

証明器の設定：

これは”Prove”アクティビティの中心となるアクションで、その目的は、次の証明で使用する証明器を設定することである。基本的に、Theorema における証明器とは、論理的証明規則（推論規則）と、その規則の適用方法との両方の集まりである。利用できる規則の集まりはメニューから選択可能である。”Proof Rules Setup”セクションにおいて、全ての規則が推論規則ブラウザで階層的にグループ分けされて表示される。この階層は規則の実現時に定義され、ユーザーが変更することはできない。このブラウザのレイアウトと機能は、上記”knowledge”アクションで示されるナレッジブラウザと似ている。基本的に、マウスをク

リックするだけで、個々の規則や規則のグループ全体を有効にしたり、無効にしたりすることができる。

次に、証明方法の選択を行い、最後に、証明器の探索の深さを調整できる。分岐やステップ、数式を減らすことで、成功した証明を簡略化できる。また、証明プロセスの一部は、インタラクティブな証明が案内がされ、証明出力を調整できる。“OK, next ...”ボタンをクリックすると、証明タスクを実行依頼する次のアクションに進む。

証明タスクの実行依頼：

“submit”アクションでは、“Prove”アクティビティで選択された全設定の要約が表示される。何か修正したい場合、上部のアクションメニューを使用して、個々のアクションに戻る。問題なければ、“Prove!”ボタンを押す。ここで、Theorema は証明の生成を実行する。

得られた証明の検証：

証明が生成されていくにつれ、“inspect”アクションでは、証明探索ツリーが成長しながら表示される。また、ツリーがある程度大きく成長すると、その一部が表示される。

証明が終了すると、Theorema コマンドは、最終的な証明ツリーを表示し、証明情報が Theorema ノートブック内の、目標を含む環境のすぐ下書き込まれる。

チェックが表示されたものは、証明の成功を示している（？が表示されたものは、証明の失敗を示している）。証明した式のラベルが “Proof of” の後ろに書かれ、さらに、その証明を行った回数の番号が # の後ろに書かれ：Show proof \times （前述の記号）が続く。

証明情報の詳細表示には、左端のオープナーを使用して、セルグループをオープンすることができる。ここで表示される情報はノートブックに保存されていない。ノートブックの関連ディレクトリにある外部ファイルから読み出されている。従って証明を削除したい場合は、Mathematica 自体の「切り取り」や「削除」を使用せずに、右端にある “ \times ” をクリックする。これは Theorema が関連ディレクトリをクリーンアップできるようにするためである。

4 THEOREMA を用いた自動証明

このセクションにおいては、[1] における命題論理を Theorema を用いて証明した例を示す。ここでもう一度、必要な記号のみを以下に示す。

X_a : くじ A において当たりくじが出る事象

X_b : くじ B において当たりくじが出る事象

S_a : くじ A に関する全事象

S_b : くじ B に関する全事象

$n(X_a)$: くじ A の当たりくじの本数

$n(X_b)$: くじ B の当たりくじの本数

$n(S_a)$: くじ A の総数

$n(S_b)$: くじ B の総数

〔設定〕 くじ引きで、「当たりくじの数」と「はずれくじの数」を合わせたくじの数を「全体のくじの数」とする。くじ引きで、「当たりくじ」が出やすいことを「当たりやすい」と呼ぶ。「当たりくじ」と「はずれくじ」の入った 2 つのくじ A, B がある。このとき、以下の問題を考える。

問題：

くじ A とくじ B の「全体のくじの数」は同じとする。くじ A は、くじ B より「当たりくじ」が多く入っているとするとする。（くじ B は、くじ A より「はずれくじ」が多く入っているとするとする。）くじを 1 本ずつ引くとき、くじ A とくじ B では、どちらが「当たりやすい」かを考える。

この問題を、Theorema を用いて命題論理として証明する手順は、以下のようになる。まずはじめに、Theorema コマンドで、Session の NOTEBOOK で新しいファイルを開き、環境用テンプレートから定理と定義を入れる。このとき、符号などの記号は、Session 中にあるものを使うか、Mathematica のアシスタントパレットを使用する。キーボードから直接打ち込むと、エラーにつながる可能性がある。環境用テンプレートの Definition で定義を設定する。くじは自然数であるから任意の自然数とし、定義となる対称律や、推移律などを設定する。環境用テンプレートの Theorem で定理をいれる。環境用テンプレートは、どれを選択しても問題はないため、定理や定義などに Example や、Lemma などを使用しても結果は、同じである。ここでの Theorem は、証明したい事であるとする。すべて数式を入力し終えたら、Theorema に認識させるために、それらの数式を評価しなければならない。評価をしたら証明の準備は、終わりである。次に、Theorema コマンドの Prove に移り、証明プロセスを進める。証明対象が含まれるセルを選択し、ok,next を押し、Proof の前にチェックがついた状態になる。これによって、証明が終了する (図 3)。

DEFINITION (1)

$$(((n(S_a)) \in \mathbb{N}) \wedge ((n(X_a)) \in \mathbb{N}) \wedge ((n(S_b)) \in \mathbb{N}) \wedge ((n(X_b)) \in \mathbb{N})) \quad (\text{必須2}) \times$$

$$(((n(X_b)) \div (n(S_b))) < ((n(X_a)) \div (n(S_a)))) \Rightarrow P \quad (\text{必須3}) \times$$

THEOREM (1)

$$(((n(S_a)) = (n(S_b))) \wedge ((n(X_b)) < (n(X_a)))) \Rightarrow P \quad (\text{証明}) \times$$

☑ Proof of (60) #3: Show proof

図 3: 証明

5 考察

Theorema 2.0 を用いて、初等数学 (確率概念の基礎) に関する命題論理の証明を試みたが、未だに、その機能が明らかでない部分がある。現状では、マニュアル (Theorema FirstTour) [2] を見るだけでは厳しい。

参考文献

- [1] Takeshi Sakai and Tadashi Takahashi. Logical analysis of ratio inference by children. *Scientiae Mathematicae Japonicae*, Online, e-2015-20, pp. 1-28, 2015.
- [2] <https://www.risc.jku.at/research/theorema/software/documentation/tutorial/FirstTour.html>