数式処理 *Bulletin of JSSAC (2009)*
Vol. 16, No. 2, pp. 32 - 36

特集記事＜統計科学と数式処理＞

Isabelle/HOLによる分割表の数え上げ

小野 陽子*

新潟国際情報大学 情報文化学部

1 はじめに

形式的手法とは，数学における論理学・集合論・代数学などを基盤としたシステムの記述手法や検証手法などの総称であり，近年 CPU 回路設計の検証やソフトウェア開発に用いられる．形式的手法は，大別すると仕様を厳密に記述することを主目的にする仕様記述と，モデルを検証することを主目的にする形式検証とに分類される．これらの共通部分に定理証明系は位置する．この定理証明系を用いて示された証明には，論理の隙間も,「よって自明」という数学のテキストで見慣れた表現も存在しない．また，形式的手法で表現されていることから，自然言語表現が内包する表現上の曖昧さと冗長さが回避されている．

定理証明系は，限定されたドメインを対象として完全自動証明を行うもの（例：Boyer-Moore 証明系 [1]）と，ドメインを限定せずにルール摘要の形で形式証明をユーザ自身が書くことを助ける定理証明支援系（例：Coq [4], HOL [3], Isabelle/HOL [8]）の２つに大別される．

本稿では，定理支援証明系のひとつである Isabelle/HOL による定理証明の簡単な例と，Isabelle/HOL の統計への適用として，2 元分割表の数え上げに関する Foulkes theorem の証明を紹介する．更に，Isabelle/HOL を利用した定理証明の現状での問題点と自動証明システム構築への展望を記す.

2 Isabelle/HOL による証明

Isabelle/HOL は高階論理を基盤とした定理証明支援系であり，証明は対話型で行われる．Proof General と呼ばれる Emacs 上でのインターフェースを利用して記述する．対話型であるため，ユーザが証明する命題を入力すると，命題が証明すべきゴールとして返される．ユーザは apply コマンドにより，ゴールを証明する方法を指定する．実行後サブゴールが返されたなら，再びユーザは証明方法を指定する．示すべきサブゴールがなくなれば証明終了である．

1. 証明する事柄の明示化
   証明したい命題を次のように記述する．このとき，後の命題証明にて利用することを考慮し，命題に名前をつけておくことが望ましい.

*onoyk@nuis.ac.jp

lemma 名前 :"[|仮定|] ⟹ 結論"

2. 規則の適用
   1 を実行すると，記述した命題が証明のゴールとして提示されるので，apply を用いて規則を適用する．

   apply(適用したい規則を記述)

3. 証明終了
   規則を適用して得られたサブゴールに対し，新たに規則の適用を繰り返した結果，示すべきサブゴールがなくなると，No subgoals!と表示される．これは命題の証明が終了することを意味する．証明の最後は done と記述する．これで一つの命題が完結する．

例として,「A が B の部分集合であり，B が C の部分集合であるならば，A は C の部分集合である」という命題を形式化し，example という名前をつけたものを下に記す.

```
lemma example:"[|A ⊆ B; B ⊆ C|] ==> A ⊆ C"
        apply (rule subsetI)
        apply (frule_tac c = x in subsetD[ of A B ])
        apply assumption
        apply (frule_tac c = x in subsetD[ of B C ])
        apply assumption+
done
```

3 Isabelle/HOL を用いた分割表の数え上げ

分割表とは 2 以上の変数間の関係を分析するために記録されたものである．例えば，喫煙とある病気の発病の関係性を分析したいものとする．対象者に対し，喫煙しているかどうか，発病しているかどうかを調査し，その人数を 2×2 行列にまとめた上，各行と列ごとの和（周辺和）を求め，検定などの統計的手法を用いて関係性を分析することが可能である．n 変数の分割表は n 元分割表と呼ばれるが，統計解析で利用される分割表は 2 元，3 元が多い．
周辺和を固定した 2 元分割表の数え上げに，Diaconis and Gangolli [2] で述べられている Foulkes の定理がある．

定理 **1 (Foulkes)**
変数 A のカテゴリ数 m，変数 B のカテゴリ数 n である 2 元分割表を扱う．各行の和を$\boldsymbol{r}$= $\{r_1, r_2, \ldots, r_m\}$, 各列の和を$\boldsymbol{c}$= $\{c_1, c_2, \ldots, c_n\}$ とし，行と列の総和を $N = \sum_{i=1}^{m} r_i = \sum_{j=1}^{n} c_j$ で表す．周辺和$\boldsymbol{r}$, $\boldsymbol{c}$ を持つ分割表全体を Σ_{rc} とする．N 元の置換全体 S_N において，$D(\pi) \subseteq D(\boldsymbol{r})$, $D(\pi^{-1}) \subseteq D(\boldsymbol{c})$ をみたす置換 π の数は Σ_{rc} のサイズ（個数）と一致する．ただし，$D(\pi)$ は π の descent 集合を表し，$D(\pi) = \{i : \pi(i) > \pi(i+1)\}$ である．

以降，置換 π または置換行列 P が分割表 $T \in \Sigma_{rc}$ に対して descent 条件をみたすとは，π が $D(\pi) \subseteq D(\boldsymbol{r})$ かつ $D(\pi^{-1}) \subseteq D(\boldsymbol{c})$ の 2 条件をみたすことと定義する．ただし，$D(\boldsymbol{r}) =$

表 1: 分割表 $\boldsymbol{r}$=$\boldsymbol{c}$={1,1,2}

1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	1
0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	1
0	0	2	0	0	2	0	1	1	1	0	1	0	1	1	1	0	1	1	1	0

π:1234	**2**134	1**3**24	**3**124	2**3**14	**3**214	3**4**12
π^{-1}:1234	**2**134	1**3**24	2**3**14	**3**124	**3**214	3**4**12

$D(r_1, r_2, \ldots, r_m) = \{d_1, d_2, \ldots, d_{m-1}\} = \{r_1, r_1+r_2, \ldots, r_1+r_2+\ldots+r_{m-1}\}$, $D(\boldsymbol{c}) = D(c_1, c_2, \ldots, c_n) = \{d_1, d_2, \ldots, d_{n-1}\} = \{c_1, c_1 + c_2, \ldots, c_1 + c_2 + \ldots + c_{n-1}\}$.

例として，周辺和が$\boldsymbol{r}$=$\boldsymbol{c}$={1,1,2}の 2 元分割表の数え上げを考慮する．$N = 4$, $D(1, 1, 2) = \{1, 2\}$ より，$D(\pi) \subseteq \{1, 2\}$, $D(\pi^{-1}) \subseteq \{1, 2\}$ となることから，(1,2,3,4) に対する π/π^{-1} の組み合わせは表 1 に記すように，7 通りである．なお，π/π^{-1} の太字は descent を示す.

Foulkes の定理を Isabelle/HOL を用いて証明するための鍵は，分割表 T に対し，descent 条件をみたす置換行列 P を作成し，T の集合から P の集合への写像が全単射であることを示すことである [9]．しかし，Diaconis and Gangolli では数行で記述されているこの証明も，Isabelle/HOL では多くの準備を要する．Foulkes の定理の形式化と形式的手法を用いた証明を行う手順は数千行に渡るので，以下では形式化の例をいくつか示す.

1. 基本的な定義と定理の準備
   置換や行列の定義を行うために必要とされる基本事象を証明し，準備をしておく必要がある．Isabelle/HOL には数論，集合論，群論といった形式的手法の基盤となる数学に関する知識データベースが存在するが，完備されているとは言えない．また，ユーザの証明方針によっては，補題として準備をしておくことで，冗長な証明を回避することができることがある．準備しておく命題の例として，自然数の区間に関する命題が挙げられる．この命題を下に記す.

   "[|a < y; y ≤ a + b|] ==> ∃ s ∈ {1. . b}. y = a + s"

2. 置換，行列などの定理証明に直接必要とされる事柄の準備
   分割表 T から置換行列 P を作成するためには，行列，置換，置換行列の定義が必要である．しかしながらこれらの定義は Isabelle/HOL に用意されていないため，ユーザが準備をしなければならない．N 文字の置換 f を次のように表現する．ここでの extensional, bij_to は写像に関して定義されたものである.

   "permutation N f == f ∈ extensional{1. . N} ∧ bij_to f {1. . N}{1. . N} "

   さらに，permutation を用いて，f を N 文字の置換とし，P,Q を f により定められる $N \times N$ 行列としたとき，$P = Q$ であるとする命題を形式的に表現する:

   "[|permutation N f; permutation_matrix N f P; permutation_matrix N f Q|] ==> P = Q"

3. 分割表 T から置換行列 P を作成（T の集合から P の集合への写像の定義にあたる）
   分割表から descent 条件をみたす置換行列をブロックごとに作成することを次のように表現する．

   "T_to_Perm_block i j T r c P ==
   (∀ x ∈ (horizontal_strip i r). ∀ y ∈ (vertical_strip j c).
   (if (∃ s ∈ {1. .(T i j)}.
   x = (Σ k = 1. . (i-1). (r k))+Σ k = 1. .(j-1).(T i k))+ s ∧
   y = (Σ l = 1. .(j-1).(c l))+Σ l = 1. .(i-1). (T l j))+s)
   then (P x y = 1) else (P x y = 0)))"

4. 置換行列 P から分割表 T を作成
   descent 条件をみたす置換行列から分割表を作成することを次のように構成する．

   "P_to_Table N m n r c P T == (∀ i ∈ {1..m}. ∀ j ∈ {1..n}.
   T i j = (Σ l=(srow i r)..(lrow i r). (Σ k=(scol j c)..(lcol j c). (P l k))))"

5. T から P の写像が全単射であることの証明
   上述の準備がなされた後に，定理の鍵である全単射の証明に入ることができる．下には全射に関する命題のみを記す．

   lemma T_to_P_surjection:
   "[| n_matrix m n T; permutation N f; permutation_matrix N f P;
   composition_n N m r; composition_n N n c; P_to_Table N m n r c P T;
   Comp_to_D m r dr; Comp_to_D n c dc;
   descents N f ∈ dr‘{1. .(m-1)}; descents N (f^{p}_{N}) ∈ dc‘{1. .(n-1)}|]
   ==> ∀ i ∈ {1. .m}. ∀ j ∈ {1. . n}. T_to_Perm_block i j T r c P"

4 形式的手法を用いた定理証明の課題

形式的手法を用いた定理証明の多くは，形式的手法の基盤である論理学・代数学・群論などに関するものが多く，形式化表現を行いやすい命題であると言える．しかし，本稿で紹介した統計学への利用には準備の多さと定義の複雑化が見られた．このことから，現状での問題点と今後の課題を示す．

1. 知識データベースの構築
   Isabelle/HOL の知識データベースは日々更新されており，ユーザグループにより証明された命題が審査を経た上で共通知識として利用されている [6]．しかし，それらの多くは基礎的な問題に留まっており，ブルバキ・中山の定理といった，大学で数学を専門に学ぶ学生が目にするような命題証明は，一部ユーザからの支援に頼っている状況にある [5]．証明したい命題のために準備しなければならない命題の方が多いようでは，何のための定理支援であるのかがわからず，証明の道筋を見失うことになる．知識データベースの充足は最優先課題であると思われる．

2. 形式化の互換性
   異なる定理証明支援系には互換性がなく，知識の共有がなされていない．したがって，ユーザは選択したシステムが有するデータベースが所有しない知識を，自ら命題を形式化した上で証明しなければならない．これは定理証明支援系の発展を妨げるものと思われる．命題証明成功過程に関して，互換性のあるデータベースが構築されることで，本来の目的である証明支援の簡易化と高速化が期待される．
3. 自動証明システムの構築
   定理証明支援系の問題のひとつに，言語としての成熟度の低さが挙げられる．現状では，*Mathematica* などの数式処理システムのように，ユーザが詳細を意識することなく関数を利用することで目的を達成するには程遠い．また，Isabelle/HOL に限らず，定理証明支援系は対話的に証明を行うため，サブゴールをユーザが解釈し，規則を適用する必要がある．ユーザの数学的知識と利用するシステムの理解度により，証明の仕方は大きく異なる．
   知識データベースが充足されただけでは，人間の有する数学知識をなぞらえただけに過ぎない．ユーザの能力に依存した証明支援だけではなく，知識データベースを活用した自動証明システムの構築が待たれている．SQL を利用した自動証明システム構築への取り組みがなされているとの報告もある [7]．これらのシステムの完成により，新しい数学命題とその証明がなされることが期待される．

参 考 文 献

[1] Boyer, R.S. and Moore, J.S. Computational Logic, 1980, Academic Press.
[2] Diaconis, P. and Gangolli, A.: Rectangular Arrays with Fixed Margins, Discrete Probability and Algorithms (D.Aldous et al., eds.), 1994, pp.15?41, Springer.
[3] Gordon, M.J.C. and Melham, T.F.: Introduction to HOL: A Theorem Proving Environment for Higher Order Logic, 1993, Cambridge University Press.
[4] INRIA: The Coq Proof Assistant, http://coq.inria.fr/.
[5] Kobayashi, H.: *Fundamental Properties of Valuation Theory and Hensel's Lemma*, The Archive of Formal Proofs, 2007, http://afp.sourceforge.net/entries/Valuation.shtml.
[6] Kobayashi, H., Chen, L. and Murao, H.: *Groups, Rings and Modules, The Archive of Formal Proofs*, 2004, http:// afp.sourceforge.net/entries/Group-Ring-Module.shtml.
[7] Kobayashi, H.and Group You Santo: Formalization of abstract algebra in Isabelle/HOL, http://www.formalg.com/.
[8] Nipkow, T., Paulson, C.L. and Wenzel, M.: Isabelle/HOL: A Proof Assistant for Higher-Order Logic, 2002, Springer.
[9] Ono,Y. and Kobayashi,H.: *Comparison a human proof with a proof in Isabelle*, Proc. of the 2nd Workshop on Programming Languages for Mechanized Mathematical Systems, 2008, pp.29-40, PLMMS.