2009 International Symposium on Computing, Communication, and Control (ISCCC 2009)
Proc .of CSIT vol.1 (2011) © (2011) IACSIT Press, Singapore

An Application of the Formal Method to Statistics

Hidetsune Kobayashi [1] and Yoko Ono [2 +]

[1] Nihon University

[2] Niigata University of International and Information Studies

Abstract. We applied a formal method to a theory on statistics, and described some problems of formalization with proposals for resolving them. We show that the proof of formal method makes clear the logical structure of the proof.

Keywords: formal method, automated reasoning system, Isabelle/HOL, contingency tables, descents

1. Introduction

Formal method is based on logics and mathematics, and it is used to describe, to develop, to check an information system. For example, it is applied to check the correctness of systems requiring high reliability such as a design of CPU circuit and a security protocol.

Since a formal proof allows no logical gap in inference steps and there is no room for error, those facts proved formally are highly trustworthy. Therefore if a formal expression is proved by a computer, correctness of, say a design of CPU circuit, is guaranteed provided that the formal expression represents the design precisely.

In this study, as an application to statistics, we formalized a theory of contingency tables based on Foulkes' proof [1] in Isabelle/HOL [4].

We have already formalized abstract mathematics [2][3], and according to the experience of formalization, we presented some problems concerning to formalization with proposals to resolve them. In section 2, we introduce formalized propositions to show how mathematical concepts are defined, how propositions are formalized and how a formal proof is written. In section 3 we discuss advantages of formal proof comparing with human proof. In section 4, we present some problems on formalization and give proposals to resolve them.

2. An Application of Formal Methods to Statistics

As a trial of formalization, we formalized a Foulkes' enumeration theorem of contingency tables which is used in statistics. Let $\mathbf{r} = (r_1, r_2, \ldots, r_n)$ and $\mathbf{c} = (c_1, c_2, \ldots, c_m)$ be vectors over natural numbers such that $\sum_{i=1}^{n} c_i = \sum_{j=1}^{m} r_j = N$ with a natural number N. A contingency table is an $m \times n$ matrix over natural numbers $T = (T_{ij})$ satisfying equations $\sum_{j=1}^{m} T_{ij} = c_i, \sum_{i=1}^{n} T_{ij} = r_j$. The number of contingency tables with fixed vectors c and r has a statistical meaning. Foulkes gave the following:

+ Corresponding author. Tel.: + 81-25-239-3111; fax: +81-25-239-3690.
E-mail address: onoyk@nuis.ac.jp.

Theorem (Foulkes) *Let* ***r*** *and* ***c*** *be compositions of N. The number of permutations p in N letters satisfying a descent condition D(**r**) and p−1 satisfying a descent condition D(**c**) coincides the number of contingency tables determined by* ***c*** *and* ***r****.*

The steps to formalize the above theorem are as follows:

(1) Formalize elementary properties and definitions for the theorem. Definitions of permutation, contingency table and permutation matrix and simple properties of these things are included in this category. For example, the following proposition concerning natural number interval is an elementary property.

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

A definition of a permutation f of N letters is formalized as;

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

(2) Formalize mathematical properties of permutation, matrix which are required to compose a formal proof of Foulkes' theorem. We present an example:

"[[permutation N f; permutation_matrix N f P; permutation_matrix N f Q]] ⟹ P = Q"

The meaning of this proposition is "let f be a permutation of N letters, and let P and Q be $\mathbb{N} \times \mathbb{N}$ matrices determined by f, then we have P = Q.

(3) Formalize Foulkes' algorithm deciding a permutation matrix from a contingency table T. The algorithm is written as:

(* Given a contingency table, we determine a P matrix block-wise. i, j block of P is given as follows.

In short, T to (i, j) block *)

"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 = (Σk = 1..(i − 1).(r k)) + (Σk = 1..(j − 1).(T i k)) + s ∧
y = (Σl = 1..(j − 1).(c l)) + (Σl = 1..(i − 1).(T l j)) + s)
then (P x y = 1) else (P x y = 0)))"

(4) Finally, we formalize a proof of Foulkes' theorem. Since a permutation matrix is determined by the algorithm, it is clear that the correspondence from contingency tables to permutation matrices is a one-to-one map. Therefore we have only to show that the correspondence is surjective. This is formalized as:

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"

3. Advantages of Formalization

The most important role of a formal method is to give a trustworthy system. In addition, there is another advantage of a formal method. That is it makes clear the logical structure of a formalized theory, because it has no logical gap. Almost any human proof is (partly) supported by human intuition and it has some logical gaps, and those gaps are easily filled up by some experts. However those gaps are fatal for some beginners and they cannot understand theorems having a proof written with logical gaps.

Although a formalized proof is much longer than a human proof, we can illustrate the logical structure of a formalized proof block by block and thus it makes the theorem easy to understand.

The logical structure of the proof to the proposition "lemma T_to_P_surjection" is described as:

Step 1.

We deny the conclusion, then we have the least im with

$\exists j \in \{Suc\ 0..n\}. \neg$ T_to_Perm_block im j T r c P

and the least jm with

$\neg$ T_to_Perm_block im jm T r c P

This means that there is the least im and the least jm with im fixed such that the (im, jm) block of P is not equal to the block decided by $T_{im\,jm}$ of the contingency table T. The original proposition is changed as:

$\wedge$ im jm.

[[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 $\subseteq$ dr `{Suc 0..m − Suc 0};
descents N ($f^P{}_N$) $\subseteq$ dc `{Suc 0..n − Suc 0}; im $\in$ {Suc 0..m};
$\forall$ x $\in$ {Suc 0..im − Suc 0}. $\forall$ j $\in$ {Suc 0..n}. T_to_Perm_block x j T r c P;
jm $\in$ {Suc 0..n}; $\neg$ T_to_Perm_block im jm T r c P;
$\forall$ x $\in$ {Suc 0..jm − Suc 0}. T_to_Perm_block im x T r c P; P_matrix N P;
contingency_table m n N r c T]]
$\Longrightarrow$ False

Step 2.

If $T_{im\,jm} = 0$ then P is uniquely determined since the entries of the (im, jm) block of P are all zero. Therefore in the case " $T_{im\,jm} = 0$", T to Perm block im jm T r c P is true, hence we have contradiction. The proposition we have to prove is the above proposition with " $T_{im\,jm} \neq 0$" added in the premises. Unfolding the definition of T to Perm block, we have ¬ T_to_Perm_block im jm T r c P rewritten as

$\exists$ x $\in$ horizontal_strip im r.
$\exists$ y $\in$ vertical_strip jm c.
$\neg$ (if $\exists$ s $\in$ {1..T im jm}.
x = setsum r {1..im − 1} + setsum (T im) {1..jm − 1} + s $\wedge$
y = setsum c {1..jm − 1} + (Σ l = 1..im − 1. T l jm) + s
then $P^{\prime}$ x y = 1 else $P^{\prime}$ x y = 0)

Step 3.

Now, we consider the case

$\exists$ s $\in$ {1..T im jm}.
x = setsum r {1..im − 1} + setsum (T im) {1..jm − 1} + s $\wedge$
y = setsum c {1..jm − 1} + (Σ l = 1..im − 1. T l jm) + s

We can prove this case applying lemma T_to_P_surjection01 which is proved earlier.

Step 4.

The case

$\neg$ ($\exists$ s $\in$ {1..T im jm}.
x = setsum r {1..im − 1} + setsum (T im) {1..jm − 1} + s $\wedge$
y = setsum c {1..jm − 1} + (Σ l = 1..im − 1. T l jm) + s)

is left. By logical computation, the final goal is written as

[[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 $\subseteq$ dr `{1..m − 1};
descents N ($f^P{}_N$) $\subseteq$ dc `{1..n − 1}; im $\in$ {1..m};

```
∀x ∈ {1..im − 1}.∀j ∈ {1..n}.T_to_Perm_block x j T r c P; jm ∈ {1..n};
∀x ∈ {1..jm − 1}.T_to_Perm_block im x T r c P; P_matrix N P;
contingency_table m n N r c T; 0 < T im jm;
x ∈ horizontal_strip im r; y ∈ vertical_strip jm c; 0 < P x y;
∀s ∈ {1..T im jm}.
x = setsum r {1..im − 1} + setsum (T im) {1..jm − 1} + s −→
y ≠ setsum c {1..jm − 1} + (Σl = 1..im − 1.T l jm) + s]]
⟹ False
```

In this case, we have

```
"∃s ∈ {1..T im jm}.
P (setsum r {1..im − 1} + setsum (T im) {1..jm − 1} + s)
(setsum c {1..jm − 1} + (Σl = 1..im − 1.T l jm) + s) = 0"
```

and this gives contradiction by virtue of the conditions that the permutation f satisfies.

Thus we see the logical structure of the formal proof. Actual proof is long but it is not difficult to check the detail of the proof.

4. Problems Concerning with Formalization

Through formalization of contingency tables and abstract mathematics, we see following four items are important problems.

(1) knowledge database

At present, since we do not have enough formalization of elementary properties, we have to formalize the basic elementary facts. If we have enough elementary properties already formalized, then it will be easy to start writing formalized proofs.

(2) compatibility of formalization

There are many formalizing languages. Therefore if we have no automatic translator between them, we cannot use knowledge database written in other formalizing language. Thus we see that we need automatic translator between formalizing languages.

(3) systematize formalization

In the knowledge database for formalization, knowledge should be accumulated systematically. For example, a systematic arrangement of formalized knowledge in mathematics will be, elementary properties of numbers, elementary properties of sets and functions and then group theory. Hence it will be a good way to give a skeleton of formalization and then fill formalization to each theory in the skeleton.

(4) automated reasoning

At present, formalized proofs are written by man power. But we see some steps of formal proofs are quite simple and we can expect such steps are proved by automated reasoning. That is we have to develop an automated reasoning system to save human labor.

5. References

[1] Diaconis, P. and Gangolli, A., Rectangular Arrays with Fixed Margins, Discrete Probability and Algorithms (D. Aldous et al., eds.), pp.15–41, Springer, New York, 1994.

[2] Kobayashi, H., Chen, L. and Murao, H., Groups, Rings and Modules, *The Archive of Formal Proofs*, http:// afp.sourceforge.net/entries/Group-Ring-Module.shtml, 2004.

[3] Kobayashi, H., Fundamental Properties of Valuation Theory and Hensel's Lemma, *The Archive of Formal Proofs*, http:// afp.sourceforge.net/entries/ Valuation.shtml, 2007.

[4] Nipkow, T., Paulson, C.L. and Wenzel, M. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, Springer, 2002.