Formalized Search for FC-Families

Filip Marić, Bojan Vučković, Miodrag Živković

*Faculty of Mathematics,
University of Belgrade

FATPA Workshop,
2. 2. 2012.

Outline

About formal theorem proving

- Formalized mathematics and interactive theorem provers (proof assistants) have made great progress in recent years.
- Many classical mathematical theorems are formally proved.
- Intensive use in hardware and software verification.

proof-by-computation

- Most successful results in interactive theorem proving are for the problems that require proofs with much computational content.
- Highly complex proofs (and therefore often require justifications by formal means).
- Proofs combine classical mathematical statements with complex computing machinery (usually computer implementation of combinatorial algorithms).
- The corresponding paradigm is sometimes referred to as proof-by-evaluation or proof-by-computation.

Examples of proof-by-computation

- Four-Color Theorem.
- First conjectured in 1852 by Francis Guthrie.
- Century of work by many famous mathematicians (including De Morgan, Peirce, Hamilton, Cayley, Birkhoff, and Lebesgue) and many incorrect "proofs".

Examples of proof-by-computation

- Proved by Appel and Haken in 1976., using IBM 370 assembly language computer programs to carry out a gigantic case analysis (billion cases).
- The Appel and Haken proof attracted a fair amount of criticism.
- Computer programming is known to be error-prone, and difficult to relate precisely to the formal statement of a mathematical theorem.
- Attempts to simplify the proofs (e.g., Robertson et al.) — number of cases reduced, programs in C instead of assembly language.
- A doubts were removed only when Georges Gonthier employed proof assistant Coq reducing the whole proof to several basic logical principles.

Examples of proof-by-computation

- Kelpler's conjecture.
- In 1998. Thomas Hales announced the first (by now) accepted proof of Kepler's conjecture.
- It involves 3 distinct large computations.
- After 4 years of refereeing by a team of 12 referees, the referees declared that they were 99% certain of the correctness of the proof.
- Dissatisfied with this, Hales started the informal open-to-all collaborative flyspeck project to formalize the whole proof with a theorem prover.

Frankl's conjecture

Frankl's conjecture (Péter Frankl, 1979.)

For every non-trivial, finite, union-closed family of sets there is an element contained in at least half of the sets.

or dually

Frankl's conjecture

For every non-trivial, finite, intersection-closed family of sets there is an element contained in at most half of the sets.

Frankl's conjecture — example

Example

$$F = \{\{0\}, \{1\}, \{0,1\}, \{1,2\}, \{0,1,2\}\}$$

- F is union-closed.
- $|F| = 5$, $\#_F 0 = 3$, $\#_F 1 = 4$, $\#_F 2 = 2$

Frankl's conjecture — status

- Conjecture is still open.
- It is known to hold for:
  1. families of at most 36 sets (Lo Faro, 1994.),
  2. families of at most 40 sets? (Roberts, 1992., unpublished),
  3. families of sets such that their union has at most 11 elements (Bošnjak, Marković, 2008),
  4. families of sets such that their union has at most 12 elements (Vučković, Živković, 2011., computer assisted approach, unpublished).

Vučković's and Živković's proof

- Proof-by-computation.
- Sophisticated techniques (naive approach is doomed to fail — requires listing $2^{2^{12}} = 2^{4096}$ families).
- JAVA programs that perform combinatorial search.
- Programs are highly complex and optimized for efficiency.
- Abstract mathematics and concrete implementation tricks are not separated.
- How can this kind of proof be trusted?
- Newer versions of the programs generate proof traces that could be inspected by independent checkers.
- Ideal candidate for formalization!

FC-families

- An important technique in proving Frankl's condition are so called FC-families.

Definition

A family F is an FC-family if for all finite union closed families F' containing F one of the elements in $\bigcup F$ is contained in at least half of the sets of F' (so F' satisfies Frankl's condition).

In the sequel we will only consider proving that a family is an FC-family, and not the full Frankl's conjecture.

Examples of FC-families

- One-element family $\{a\}$ is an FC-family.
- Two-element family $\{a_0, a_1\}$ is an FC-family.
- Each family with three three-element sets whose union is contained in a five element set is an FC-family (e.g., $\{\{a_0, a_1, a_2\}, \{a_0, a_1, a_3\}, \{a_2, a_3, a_4\}\}$).
- ...

Technique — idea

Is a the Frankl's element?

$$\begin{array}{ccccc} \{\{a,b,c\}, & \{a,c,d\}, & \{b,c,d\}\} & & \\ 1 & 1 & 0 & = 2 \geq 3/2 \\ 1/2 & 1/2 & -1/2 & = 1/2 \geq 0 \end{array}$$

Is a or b the Frankl's element?

$$\begin{array}{ccccc} \{\{a,b,c\}, & \{a,c,d\}, & \{b,c,d\}\} & & \\ 2 & 1 & 1 & = 4 \geq 2 \cdot 3/2 \\ 1 & 0 & 0 & = 1 \geq 0 \end{array}$$

Frankl's condition — formal definition

$$\text{frankl } F \equiv \exists a.\ a \in \bigcup F \wedge 2 \cdot \#_F a \geq |F|$$

- Note that division is avoided in order to stay within integers — this is done throughout the formalization.

Weight functions

Weight functions — definition

A function $w : X \to \mathbb{N}$ is a weight function on X, denoted by $\mathrm{wf}_X\ w$, iff $\exists x \in X.\ w(x) > 0$.
Weight of a set A, denoted by $w(A)$, is the value $\sum_{x \in A} w(x)$.
Weight of a family F, denoted by $w(F)$, is the value $\sum_{A \in F} w(A)$.

Weight functions

Weight functions — example

- Let w be a function such that $w(a_0) = 1, w(a_1) = 2$, and $w(a_i) = 0$ for all other elements a_i, $i > 1$.
- w is clearly a weight function.
- $w(\{a_0, a_1, a_2\}) = 3$,
- $w(\{\{a_0, a_1\}, \{a_1, a_2\}, \{a_1\}\}) = 7$.

Frankl's characterization using weight functions

Lemma

frankl $F \iff \exists w.\ \mathrm{wf}_{(\bigcup F)}\ w \ \wedge\ 2 \cdot w(F) \ \geq\ w(\bigcup F) \cdot |F|$

Proof sketch

$\Rightarrow$: If F is Frankl's, then let w assign 1 to the element a that is contained in at least half of the sets and 0 to all other elements. Then, $w(F) = \#_F a$ and $w(\bigcup F) = 1$, and since $\#_F a \geq |F|/2$, the statement holds.

$\Leftarrow$: If F is not Frankl's, then for all a, it holds $\#_F a < |F|/2$. Then, $2 \cdot w(F) = 2 \cdot \Sigma_{a \in \bigcup F} \#_F a \cdot w(a) < |F| \cdot \Sigma_{a \in \bigcup F} w(a) = |F| \cdot w(\bigcup F)$.

Shares

A slightly more operative characterization is obtained by introducing set share concept, as it expresses how much does each member set contributes to a Family being Frankl's.

Share — definition

Let w be a weight function and X a set.
Share of a set A with respect to w and X, denoted by $\bar{w}_X(A)$, is the value $2 \cdot w(A) - w(X)$.
Share of a family F with respect to w and X, denoted by $\bar{w}_X(F)$, is the value $\sum_{A \in F} \bar{w}_X(A)$.

Proposition

$$\bar{w}_X(F) = 2 \cdot w(F) - w(X) \cdot |F|$$

Share — example

Let w be a function such that $w(a_0) = 1, w(a_1) = 2$, and $w(a_i) = 0$ for all other elements a_i, $i > 1$.

$$\begin{aligned} \bar{w}_{\{a_0,a_1,a_2\}}(\{a_1, a_2\}) &= 2 \cdot w(\{a_1, a_2\}) - w(\{a_0, a_1, a_2\}) \\ &= 4 - 3 = 1. \end{aligned}$$

$$\begin{aligned} \bar{w}_{\{a_0,a_1,a_2\}}(\{\{a_0, a_1\}, \{a_1, a_2\}, \{a_1\}\}) &= \\ (2 \cdot 3 - 3) + (2 \cdot 2 - 3) + (2 \cdot 2 - 3) &= 5. \end{aligned}$$

Frankl's characterization using shares functions

Lemma

$$\text{frankl } F \iff \exists w.\ \text{wf}_{(\bigcup F)}\ w \ \wedge\ \bar{w}_{(\bigcup F)}(F) \geq 0$$

Proof idea — FC family

Now we consider the problem of proving that certain family is an FC-family. For example, let us analyze the proof of the following theorem.

Theorem

Each finite union-closed family containing $\{a_0, a_1\}$ is Frankl's.

Consider, e.g., the union-closed family F: $\{\{a_0, a_1\}, \{x_0\}, \{x_0, a_0\}, \{x_0, x_1\}, \{x_0, a_0, a_1\}, \{x_0, x_1, a_0\}, \{x_0, x_1, a_1\}, \{x_0, x_1, a_0, a_1\}\}$
How to show that it is Frankl's?

Reorganize:

$$
\begin{array}{lcl}
\{\} & - & \{\{a_0, a_1\}\} \\
\{x_0\} & - & \{\{x_0\}, \{x_0, a_0\}, \{x_0, a_0, a_1\}\} \\
\{x_1\} & - & \{\} \\
\{x_0, x_1\} & - & \{\{x_0, x_1\}, \{x_0, x_1, a_0\}, \{x_0, x_1, a_1\}, \{x_0, x_1, a_0, a_1\}\}
\end{array}
$$

Proof idea — FC family

Let w be a weight function assigning 1 to a_0 and a_1, and 0 to x_0 and x_0. Share of F (i.e., $\bar{w}_{(\bigcup F)}(F)$) is the sum of shares of all parts and is non-negative if all part shares are non-negative.

$\{\}$	–	$\{\{a_0, a_1\}\}$	–	2
$\{x_0\}$	–	$\{\{x_0\}, \{x_0, a_0\}, \{x_0, a_0, a_1\}\}$	–	0
$\{x_1\}$	–	$\{\}$	–	0
$\{x_0, x_1\}$	–	$\{\{x_0, x_1\}, \{x_0, x_1, a_0\}, \{x_0, x_1, a_1\}, \{x_0, x_1, a_0, a_1\}\}$	–	0

Proof idea — FC family

Let w be a weight function assigning 1 to a_0 and a_1, and 0 to x_0 and x_0. Share of F (i.e., $\bar{w}_{(\bigcup F)}(F)$) is the sum of shares of all parts and is non-negative if all part shares are non-negative.

Things do not change if the elements x_0 and x_1 are omitted (as their weight is 0).

$$
\begin{array}{lcll}
\{\} & - & \{\{a_0, a_1\}\} & - \; 2 \\
\{x_0\} & - & \{\{\}, \{a_0\}, \{a_0, a_1\}\} & - \; 0 \\
\{x_1\} & - & \{\} & - \; 0 \\
\{x_0, x_1\} & - & \{\{\}, \{a_0\}, \{a_1\}, \{a_0, a_1\}\} & - \; 0
\end{array}
$$

Proof idea — FC family

Notice that all four ,,parts" are:

- built of elements of the initial family $\{\{a_0, a_1\}\}$,
- union closed,
- closed for unions with the members of the initial family $\{\{a_0, a_1\}\}$ (although they need not contain these).

Various families F will give various ,,part" families, but these will always satisfy listed conditions.

Proof idea — FC Family

General proof strategy

To prove that an initial family is an FC-family, choose an appropriate weight function w, list all possible families satisfying listed conditions and show that all of them have non-negative shares (with respect to w).

Hypercubes

An S-hypercube with a base K, denoted by hc_K^S, is the family $\{A.\ K \subseteq A \wedge A \subseteq K \cup S\}$. Alternatively, a hypercube can be characterized by $\mathrm{hc}_K^S = \{K \cup A.\ A \in \mathrm{pow}\ S\}$.

Proposition

1. $$\mathrm{pow}\ (K \cup S) = \bigcup_{K' \subseteq K} \mathrm{hc}_{K'}^S$$
2. If K_1 and K_2 are different and disjoint with S, then $\mathrm{hc}_{K_1}^S$ and $\mathrm{hc}_{K_2}^S$ are disjoint.

definition

A hyper-share of a family F with respect to the weight function w, the hypercube hc_K^S and the set X, denoted by $\bar{w}_{KX}^S(F)$, is the value $\sum_{A \in \mathrm{hc}_K^S \cap F} \bar{w}_X(A)$.

Lemma

Let $K \cup S = \bigcup F$ and $K \cap S = \emptyset$, and let w be a weight function on $\bigcup F$.

1. $$\bar{w}_{(\bigcup F)}(F) = \sum_{K' \subseteq K} \bar{w}_{K'(\bigcup F)}^S(F)$$
2. If $\forall K' \subseteq K.\ \bar{w}_{K'(\bigcup F)}^S(F) \geq 0$, then frankl F.

Definition

Projection of a family F onto a hypercube hc_K^S, denoted by $\mathrm{hc}_K^S \lfloor F \rfloor$, is the set $\{A - K.\ A \in \mathrm{hc}_K^S \cap F\}$.

Proposition

1. If uc F, then uc $(\mathrm{hc}_K^S \lfloor F \rfloor)$.
2. If uc F, $I \subseteq F$, $S = \bigcup I$, $K \cap S = \emptyset$, then uc_I $(\mathrm{hc}_K^S \lfloor F \rfloor)$.
3. If $\forall x \in K.\ w(x) = 0$, then $\bar{w}_{KX}^S(F) = \bar{w}_X(\mathrm{hc}_K^S \lfloor F \rfloor)$.

Definition

A set family F' is union closed for F, denoted by $\text{uc}_F\ F'$, iff

$$\text{uc}\ F' \wedge (\forall A \in F'.\ \forall B \in F.\ A \cup B \in F').$$

Union closed extensions of a set family F are families of sets that are created from elements of F and are union closed for F. Family of all union closed extensions is

$$\text{uce}\ F \equiv \{F'.\ F' \subseteq \text{pow} \bigcup F \wedge \text{uc}_F\ F'\}.$$

Theorem

Let F be a union closed family (i.e., uc F), and let F_c be its subfamily (i.e., $F_c \subseteq F$). Let w be a weight function on $\bigcup F$, and $\forall x \in \bigcup F \setminus \bigcup F_c.\ w(x) = 0$. If

$$\forall F' \in \text{uce } F_c.\ \bar{w}_{(\bigcup F_c)}(F') \geq 0,$$

then frankl F.

Search function

How to check that $\forall F' \in \text{uce } F_c.\ \bar{w}_{(\bigcup F_c)}(F') \geq 0$?

- Define a procedure ssn F w, such that if ssn F $w = \bot$, then $\forall F' \in \text{uce } F_c.\ \bar{w}_{(\bigcup F_c)}(F') \geq 0$.
- The heart of this procedure is a recursive function $\text{ssn}^{F,w,X}$ L F_t that will preform the systematic traversal of all union closed extensions of F, but with some pruning that speeds up the search.

Search function

Definition

$$
\begin{aligned}
\langle F \rangle &\equiv \{\bigcup F'.\ F' \in \mathsf{pow}\ F - \{\emptyset\}\} \\
\mathsf{ic}_I\ A\ F &\equiv F \cup \{A\} \cup \{A \cup B.\ B \in F\} \cup \{A \cup B.\ B \in I\} \\
\mathsf{ssn}^{F,w,X}\ [\,]\ F_t &\equiv \bar{w}_X(F_t) < 0 \\
\mathsf{ssn}^{F,w,X}\ (h\#t)\ F_t &\equiv \text{if } \bar{w}_X(F_t) + \sum_{A \in h\#t} \bar{w}_X(A) \geq 0 \text{ then } \bot \\
&\quad \text{else if } \mathsf{ssn}^{F,w,X}\ t\ F_t \text{ then } \top \\
&\quad \text{else if } h \in F_t \text{ then } \bot \\
&\quad \text{else } \mathsf{ssn}^{F,w,X}\ t\ (\mathsf{ic}_F\ h\ F_t)
\end{aligned}
$$

Let L be a list with no repeated elements such that its set is $\{A.\ A \in \mathsf{pow}\ \bigcup F \wedge \bar{w}_X(A) < 0\}$.

$$\mathsf{ssn}\ F\ w \equiv \mathsf{ssn}^{\langle F \rangle, w, (\bigcup F)}\ L\ \emptyset$$

Search function — correctness

Lemma

If

1. $\mathrm{ssn}^{F,w,X}\ L\ F_t = \bot$,
2. *for all elements* A *in* L *it holds that* $\bar{w}_X(A) < 0$,
3. *for all* $A \in F' - F_t$, *if* $\bar{w}_X(A) < 0$, *then* A *is in* L,
4. $F_t \subseteq F'$,
5. $\mathrm{uc}_F\ F'$,

then $\bar{w}_X(F') \geq 0$.

Lemma

If $\mathrm{ssn}\ F\ w = \bot$ *and* $F' \in \mathrm{uce}\ F$ *then* $\bar{w}_{(\bigcup F)}(F') \geq 0$.

- The formal correctness proofs are given.
- These imply that the search function is (in some sense) sound.
- The search function is also (in some sense) complete.

Search function — optimizations

- Many optimizations to the basic ssn F w definition are introduced. For example:
  - How to represent sets and families of sets so that the program becomes efficiently executable?
  - Without loss of generality assume dealing only with sets of natural numbers.
  - Encode sets of natural numbers by natural numbers (e.g., $\{0, 2, 3\}$ can be encoded by $2^0 + 2^2 + 2^3 = 13$). Computing unions (that is very frequent operation) then reduces to bitwise disjunction.
  - Avoid repeating same calculations by using memoization techniques.
- The function is refined 5 times, introducing optimization one by one, until a final version is obtained.
- Each version is shown to be equivalent with the previous one.

Symmetries

- Proofs of several theorems contain plenty symmetric cases.
- For example:

Theorem

Each family with three three-element sets whose union is contained in a five element set is an FC-family.

Consider families $\{\{a_0, a_1, a_2\}, \{a_0, a_1, a_3\}, \{a_2, a_3, a_4\}\}$ and $\{\{a_0, a_1, a_2\}, \{a_1, a_3, a_4\}, \{a_2, a_3, a_4\}\}$. These cases are symmetric since there is a permutation $(a_0, a_1, a_2, a_3, a_4) \mapsto (a_3, a_4, a_1, a_2, a_0)$ mapping one to another.

Avoiding symmetries

Definition

A family is *nkm*-family if it has m members, each with k elements, and its union is an n element set.

- Symmetries are avoided by a function that finds all nonequivalent *nkm*-families (for a given n, k, and m).
- This function is verified (if the families returned by this function are Frankl's then all non-returned *nkm*-families are also Frankl's).

Summary

- Using the demonstrated technique, it has been shown that the following families are FC-families:
  1. $\{\{a\}\}$
  2. $\{\{a, b\}\}$
  3. All 533-families.
  4. All 634-families.
  5. All 734-families.
- Total proof checking time is around 15 minutes, most of which is devoted in computation (evaluating ssn w F function).

Current work

- In this talk, I only covered results on proving FC-families.
- Currently, the case 12 of Frankl's conjecture is formalized (FC-families are important step since they allow pruning a huge amount of search space).
- Similar (but no the same) techniques used in proofs.
- High computation time, but (hopefully) still manageable.

Conclusions

- Formalization filled many gaps present in previous proofs.
- Proofs were not wrong (as they usually are not), but were imprecise.
- A big contribution of the formalization is the separation between abstract mathematical and computational content.