Towards A New Type of Prover: On the Benefits of Discovering Sequences of “Related” Proofs

David M. Cerna

April 10^{th}, 2019

State of this work

- Disclaimer: This investigation is in a very early stage.
- Essentially, we have just started looking for promising ways to circumvent a fundamental issue concerning the instance generalization prover VIPER.
  - The instance proofs need to be "related" and/or "uniform".
  - For some proof sequences this comes naturally.
  - For most it is anything but natural.
- In this talk we
  - Introduce the method,
  - Discuss its capabilities, and
  - Discuss characterizations of relatedness.

Induction: The Difficultly of Generalization

- Inductive theorem proving: find a pattern which follows from the provided axioms and can be used to prove any instance of the goal statement.
- This patterns is usually referred to as the induction invariant.
- As many here will probably know, invariant discovery is in general undecidable.
- Their exists weak theories of arithmetic where this problem is actually decidable, i.e. Pressburger arithmetic and [Aravantinos *et al.*, 2013].

Existing Methods

- There are many different approaches to invariant discovery, we will only name a few:
  - Loop-Discovery Provers [Aravantinos *et al.*, 2011]
  - Lemma Generation and testing [Claessen *et al.*, 2013]
  - Rippling [Bundy *et al.*, 2005]
  - Superposition based methods [Cruanes, 2015]
  - Cycle discovery [Brotherston, 2012]
  - and Instance proof generalization [Pearson, 1995] [Eberhard and Hetzl, 2015]
- This last approach will be the focus of this talk.

Background: Gentzen's Sequent Calculus

- The sequent calculus applies inferences to objects referred to as sequents $\Delta \vdash \Pi$, where Δ and Π are multisets of well-formed formula. Chaining inferences forms proof trees.
- Semantically a sequent means *given* Δ *we may derive* Π*.*
- Note that, this interpretation implies that Δ is essentially a conjunction of formula and Π is a disjunction.
- The sequent calculus Inferences are as follows:

Axiom Inferences

$$\frac{}{A \vdash A}\,\mathrm{Ax}$$

Gentzen's Sequent Calculus

Structural Inferences

$$\frac{\Gamma \vdash \Delta}{D, \Gamma \vdash \Delta}\ \text{w:l} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, D}\ \text{w:r}$$

$$\frac{D, D, \Gamma \vdash \Delta}{D, \Gamma \vdash \Delta}\ \text{c:l} \qquad \frac{\Gamma \vdash \Delta, D, D}{\Gamma \vdash \Delta, D}\ \text{c:r}$$

$$\frac{\Gamma \vdash \Delta, C \qquad C, \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \text{cut}$$

Gentzen's Sequent Calculus

Logical Inferences

$$\frac{\Gamma \vdash \Delta, D}{\neg D, \Gamma \vdash \Delta}\ \neg\text{:l} \qquad \frac{D, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg D}\ \neg\text{:r} \qquad \frac{C, \Gamma \vdash \Delta}{C \wedge D, \Gamma \vdash \Delta}\ \wedge\text{:l}$$

$$\frac{D, \Gamma \vdash \Delta}{C \wedge D, \Gamma \vdash \Delta}\ \wedge\text{:l} \qquad \frac{\Gamma \vdash \Delta, C}{\Gamma \vdash \Delta, C \vee D}\ \vee\text{:r} \qquad \frac{\Gamma \vdash \Delta, D}{\Gamma \vdash \Delta, C \vee D}\ \vee\text{:r}$$

$$\frac{\Gamma \vdash \Delta, C \qquad \Gamma \vdash \Delta, D}{\Gamma \vdash \Delta, C \wedge D}\ \wedge\text{:r} \qquad \frac{C, \Gamma \vdash \Delta \qquad D, \Gamma \vdash \Delta}{C \vee D, \Gamma \vdash \Delta}\ \vee\text{:l}$$

$$\frac{C, \Gamma \vdash \Delta, D}{\Gamma \vdash \Delta, C \to D}\ \to\text{:r} \qquad \frac{\Gamma \vdash \Delta, C \qquad D, \Gamma \vdash \Delta}{C \to D, \Gamma \vdash \Delta}\ \to\text{:l}$$

Gentzen's Sequent Calculus

Quantifier Inferences

$$\frac{\Gamma \vdash \Delta, F(\alpha)}{\Gamma \vdash \Delta, \forall x F(x)} \forall{:}\mathrm{r} \qquad \frac{F(t), \Gamma \vdash \Delta}{\forall x F(x), \Gamma \vdash \Delta} \forall{:}\mathrm{l}$$

$$\frac{\Gamma \vdash \Delta, F(t)}{\Gamma \vdash \Delta, \exists x F(x)} \exists{:}\mathrm{r} \qquad \frac{F(\alpha), \Gamma \vdash \Delta}{\exists x F(x), \Gamma \vdash \Delta} \exists{:}\mathrm{l}$$

- Note that for $\exists : l$ and $\forall : r$ α may not occur in Γ or Δ. These rules are referred to as Strong quantification, i.e. require an eigenvariable, the other rules are referred to as Weak.

Gentzen's Sequent Calculus

Quantifier Inferences

$$\frac{\Gamma \vdash \Delta, F(\alpha)}{\Gamma \vdash \Delta, \forall x F(x)} \forall{:}\mathrm{r} \qquad \frac{F(t), \Gamma \vdash \Delta}{\forall x F(x), \Gamma \vdash \Delta} \forall{:}\mathrm{l}$$

$$\frac{\Gamma \vdash \Delta, F(t)}{\Gamma \vdash \Delta, \exists x F(x)} \exists{:}\mathrm{r} \qquad \frac{F(\alpha), \Gamma \vdash \Delta}{\exists x F(x), \Gamma \vdash \Delta} \exists{:}\mathrm{l}$$

- Note that for $\exists : l$ and $\forall : r$ α may not occur in Γ or Δ. These rules are referred to as Strong quantification, i.e. require an eigenvariable, the other rules are referred to as Weak.

Equational Axioms

$$\frac{}{\vdash x = x} \mathrm{Re} \qquad \frac{}{x_1 = y_1, \cdots, x_n = y_n, P(x_1, \cdots, x_n) \vdash P(y_1, \cdots, y_n)} P_=$$

$$\frac{}{x_1 = y_1, \cdots, x_n = y_n \vdash f(x_1, \cdots, x_n) = f(y_1, \cdots, y_n)} f_=$$

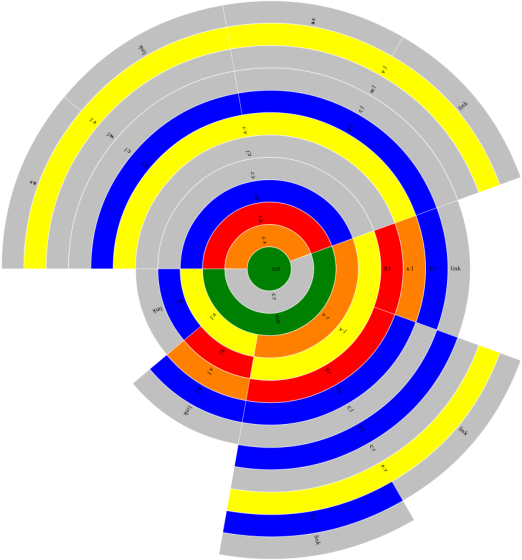
Example Sequent Proof with Cut

- Green sequents represent cuts.

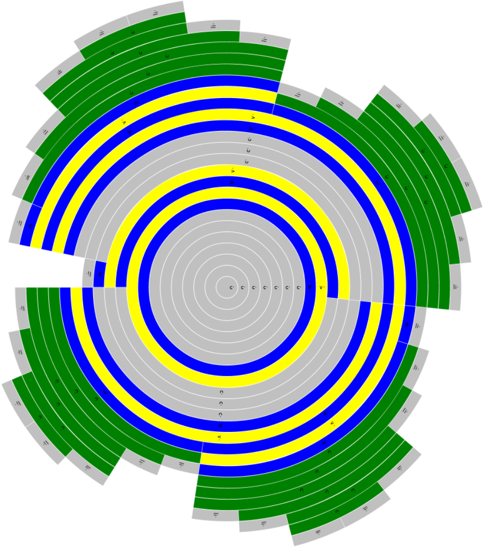
Example Sequent Proof without Cut

- Cannot eliminate atomic equational cuts.

Example Sequent Proof with Cut Sun Burst



Example Sequent Proof without Cut Sun Burst



Induction and the LK-calculus

- The theory of Peano arithmetic may by formalized as a theory extension of the LK-calculus with equality.
- Other than the axioms for successor, addition, and multiplication, one needs to add the following inference:

$$\frac{\Pi \vdash \Delta, \varphi(0) \qquad \Pi, \varphi(\alpha) \vdash \Delta, \varphi(s(\alpha))}{\Pi \vdash \Delta, \varphi(\beta)}\ \mathrm{IND}$$

- Alternatively one could consider adding the ω-rule which requires a proof of each instance of the main formula:

$$\frac{\Pi \vdash \Delta, \varphi(n) \qquad \forall n \in \mathbb{N}}{\Pi \vdash \Delta, \varphi(\beta)}\ \omega$$

- Without restrictions, the ω-rule is seemingly useless for practical cases.

Finitely describable sequences

- Fortunately, the primitive recursive ω-rule [J. Shoenfield 1959] is expressive enough to prove totality of all functions provably total in Peano arithmetic.
- Great a useful ω-rule, but how does one develop a finite description of a proof sequence?
- Maybe a little more specific, what can we do with $\varphi(0), \cdots, \varphi(n)$ for $n < \infty$?
- This is the topic of "Inductive theorem proving based on tree grammars" by S. Eberhard and S. Hetzl (2015).

Cut-freeness and the Herbrand Instances

- Not just any $\varphi(0), \cdots, \varphi(n)$ will do, we need the proofs to have particular properties.

Cut-freeness and the Herbrand Instances

- Not just any $\varphi(0), \cdots, \varphi(n)$ will do, we need the proofs to have particular properties.
- They should be proofs of the same statement.

Cut-freeness and the Herbrand Instances

- Not just any $\varphi(0), \cdots, \varphi(n)$ will do, we need the proofs to have particular properties.
- They should be proofs of the same statement.
- They should also be cut-free.
- Cut-free proofs, other than being Massive and being produced by theorem provers have particular properties.

Cut-freeness and the Herbrand Instances

- Not just any $\varphi(0), \cdots, \varphi(n)$ will do, we need the proofs to have particular properties.
- They should be proofs of the same statement.
- They should also be cut-free.
- Cut-free proofs, other than being Massive and being produced by theorem provers have particular properties.

Theorem (Mid-Sequent Theorem)

Let S be a sequent of prenex formulas then there exists a cut-free proof π of S s.t. π contains a sequent S' s.t.

- *S' is quantifier free.*
- *Every inference above S' is structural or propositional.*
- *Every inference below S' is structural or a quantifier inference.*

Cut-freeness and the Herbrand Instances

- Not just any $\varphi(0), \cdots, \varphi(n)$ will do, we need the proofs to have particular properties.
- They should be proofs of the same statement.
- They should also be cut-free.
- Cut-free proofs, other than being Massive and being produced by theorem provers have particular properties.

Theorem (Mid-Sequent Theorem)

Let S be a sequent of prenex formulas then there exists a cut-free proof π of S s.t. π contains a sequent S' s.t.

- *S' is quantifier free.*
- *Every inference above S' is structural or propositional.*
- *Every inference below S' is structural or a quantifier inference.*

- What if we limit S to a sequent only containing weak quantification.

Cut-freeness and the Herbrand Instances

- No strong quantification means no eigenvariables and thus all terms are existential witnesses.
- Collecting those witnesses gives us Herbrand's Theorem

Cut-freeness and the Herbrand Instances

- No strong quantification means no eigenvariables and thus all terms are existential witnesses.
- Collecting those witnesses gives us Herbrand's Theorem

Theorem (Herbrand's Theorem)

Let S be a sequent of the form $\forall\bar{x}\varphi(\bar{x}) \vdash \exists\bar{x}\psi(\bar{x})$. S is valid if and only if there exists a sequence of term vectors $\bar{t}_1, \cdots, \bar{t}_n$ s.t.

$$\bigwedge_{i=0}^{k} \varphi(\bar{t}_i) \vdash \bigvee_{i=0}^{k} \psi(\bar{t}_i)$$

is valid.

Cut-freeness and the Herbrand Instances

- No strong quantification means no eigenvariables and thus all terms are existential witnesses.
- Collecting those witnesses gives us Herbrand's Theorem

Theorem (Herbrand's Theorem)

Let S be a sequent of the form $\forall \bar{x} \varphi(\bar{x}) \vdash \exists \bar{x} \psi(\bar{x})$. S is valid if and only if there exists a sequence of term vectors $\bar{t}_1, \cdots, \bar{t}_n$ s.t.

$$\bigwedge_{i=0}^{k} \varphi(\bar{t}_i) \vdash \bigvee_{i=0}^{k} \psi(\bar{t}_i)$$

is valid.

- Cut-free (weakly quantified end sequent) $\Longrightarrow$ *weak* mid-sequent $\Longrightarrow$ Herbrand instances.

Using First-Order Instance Proofs

- Let $\varphi(\beta)$ be quantifier-free, Δ only contains weakly quantified formula, and $\Delta \vdash \varphi(\beta)$ the main sequent of a sound application of the ω-rule.
- Furthermore, each of the instance proofs $\varphi(n)$ for $n \in \mathbb{N}$ is provable without induction.
- We can ask a first-order theorem prover for a proof π_n of $\varphi(n)$.
- Each π_n is cut-free (atomic cuts don't count) and thus the Herbrand instances H_n may be extracted.
- At this point we can build a tree grammar G_n whose language is precisely H_n.
- Notice that G_n is <u>specific</u> to a particular π_n.

Building induction proofs from a sequence of Grammars

- This goes beyond the scope of this talk.
- For details please see "Inductive theorem proving based on tree grammars"
- Essentially, a schematic tree grammar for a particular type of induction proof may be built from the instances...

Building induction proofs from a sequence of Grammars

- This goes beyond the scope of this talk.
- For details please see "Inductive theorem proving based on tree grammars"
- Essentially, a schematic tree grammar for a particular type of induction proof may be built from the instances... The right instances.
- Now comes the issues with the method.

When Any Proof is Not Enough

- Consider the problem

$$ADD, \forall x(x + 0 = 0 + x) \vdash \forall x(x + (x + x) = (x + x) + x)$$

- While simple Heuristics are enough to prove this statement, algorithmic ATP approaches tend to have a very difficult time with this simple problem, i.e [Aravantinos *et al.*, 2013].
- The tree grammar method discussed above manages to find the invariant

$$y + (x + x) = (x + x) + y$$

  Congrats!

  > "Tree grammars for induction on inductive data types modulo equational theories" by G. Ebner and S. Hetzl

- Now, let us try

$$ADD, MUL, \forall x(x * 0 = 0 * x) \vdash \forall x(x * (x * x) = (x * x) * x)$$

When Any Proof is Not Enough

- Consider the problem

$$ADD, \forall x(x + 0 = 0 + x) \vdash \forall x(x + (x + x) = (x + x) + x)$$

- While simple Heuristics are enough to prove this statement, algorithmic ATP approaches tend to have a very difficult time with this simple problem, i.e [Aravantinos *et al.*, 2013].
- The tree grammar method discussed above manages to find the invariant

$$y + (x + x) = (x + x) + y$$

  Congrats!

  > "Tree grammars for induction on inductive data types modulo equational theories" by G. Ebner and S. Hetzl

- Now, let us try failure, why?

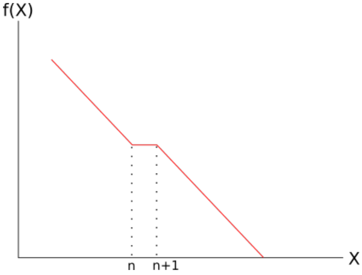
$$ADD, MUL, \forall x(x * 0 = 0 * x) \vdash \forall x(x * (x * x) = (x * x) * x)$$

Example two: The 1-Strict Monotone Assertion (1-SMA)

- A total monotonically decreasing (increasing) function $f : \mathbb{N} \to B$, $B \subseteq \mathbb{Q}$, is said to be be k-strict monotone decreasing (increasing) if there exists at least k values in A s.t. $f(a) = f(a+1)$ for $a \in A$.

Assertion (1-SMA)

Every total monotonically decreasing function $f : \mathbb{N} \to \mathbb{N}$ is at least 1-strict monotone decreasing.



- Combinatorially this statement encodes:

  Number of objects in all ascending runs in the identity permutation of n ordered objects.

1-SMA Formalized and Solved

- We formalize 1-SMA as an unsat inductive definition F:

  $$\forall n(\forall x(f(g(x)) = n \vee f(x) < n \wedge \forall x(f(x) = n \vee f(x) < n) \wedge \hat{Q}(n))$$

  where $\hat{Q}$ is defined as follows:

  $$\begin{array}{rcl} \hat{Q}(0) & \Rightarrow & \neg f(a) < 0 \ \wedge \ \forall x(\neg f(x) = 0 \vee \neg f(g(x)) = 0) \\ \hat{Q}(s(n)) & \Rightarrow & \forall x(\neg f(x) = s(n) \vee \neg f(g(x)) = s(n)) \ \wedge \\ & & \forall x(\neg f(x) < s(n) \vee f(x) = n \vee f(x) < n) \ \wedge \\ & & \forall x(\neg f(g(x)) < s(n) \vee f(g(x)) = n \vee f(x) < n) \\ & & \wedge \ \hat{Q}(n)) \end{array}$$

- Viper, an implementation of the tree grammar prover, took ($\sim$ 5 hours), but manage to find the following invariant.

  $$(F\{n \leftarrow x\} \rightarrow (f(g(a)) = 0 \vee f(a) = 0 \vee \hat{Q}(0)))\wedge$$
  $$\neg(\hat{Q}(s(x)) \wedge \hat{Q}(x) \wedge F\{n \leftarrow s(x)\})$$

When There is More Than One Way to Prove π_n

- For each successful example there are only a few ways to construct π_n.
- In truth there is only one proof modulo structural changes.
- This is not the case for the multiplication case.
- Two Instance proofs π_n and π_{n+1} may use the ADD theory and MUL theory in different ways.
- An even more important example as well as more problematic is the Non-Injectivity Assertion:

Non-Injectivity Assertion

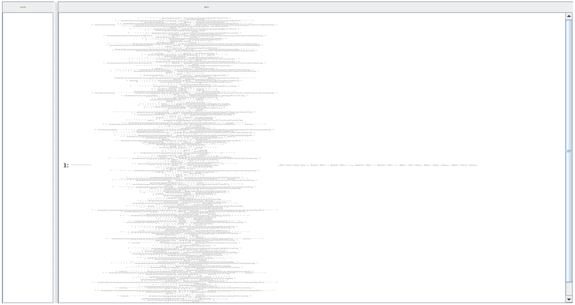
- The formula F(n) is defined as follows:

$$\forall x \left(\bigvee_{i=0}^{n} f(x) = i \right) \wedge \left(\bigwedge_{i}^{n} \forall x \forall y \neg \left(s(x) \leq y \wedge f(x) = i \wedge f(y) = i \right) \right)$$

$$\wedge \forall x \forall y \forall z \left(max(x, y) \leq z \rightarrow (x \leq z \wedge y \leq z) \right) \wedge \forall x (x \leq x)$$

- Note that $\vdash \forall n \neg F(n)$ is provable in arithmetic.
- but there are many ways to prove $F(\alpha) \vdash$ for $\alpha \in \mathbb{N}$

SPASS Herbrand Instances $F(2)$



- These Herbrand instances where found using SPASS.
- If we compare this to the Herbrand instances found by cut-elimination for $F(1)$ an issue arises.

Cut-elimination Herbrand Instances $F(1)$

$$
\exists p\ \exists q \quad
\begin{array}{c}
\langle\ \mathrm{max}(z, z)\ ,\ \mathrm{max}(g(\mathrm{max}(z, g(\mathrm{max}(z, z)))), g(\mathrm{max}(z, z)))\ \rangle \\
\langle\ \mathrm{max}(z, g(\mathrm{max}(g(\mathrm{max}(z, z)), z)))\ ,\ \mathrm{max}(g(\mathrm{max}(z, g(\mathrm{max}(g(\mathrm{max}(z, z)), z)))), g(\mathrm{max}(g(\mathrm{max}(z, z)), z)))\ \rangle \\
\langle\ \mathrm{max}(z, z)\ ,\ \mathrm{max}(z, g(\mathrm{max}(z, z)))\ \rangle \\
\langle\ \mathrm{max}(z, z)\ ,\ \mathrm{max}(g(\mathrm{max}(z, z)), z)\ \rangle \\
\langle\ \mathrm{max}(z, g(\mathrm{max}(z, z)))\ ,\ \mathrm{max}(g(\mathrm{max}(z, g(\mathrm{max}(z, z)))), g(\mathrm{max}(z, z)))\ \rangle \\
\langle\ \mathrm{max}(g(\mathrm{max}(z, z)), z)\ ,\ \mathrm{max}(z, g(\mathrm{max}(g(\mathrm{max}(z, z)), z)))\ \rangle \\
\langle\ \mathrm{max}(g(\mathrm{max}(z, z)), z)\ ,\ \mathrm{max}(g(\mathrm{max}(z, g(\mathrm{max}(g(\mathrm{max}(z, z)), z)))), g(\mathrm{max}(g(\mathrm{max}(z, z)), z)))\ \rangle
\end{array}
\quad (\ \mathrm{LE}(p, \mathrm{c}
$$

- If you look closely (and know the problem) you will see that it is just counting natural numbers.
- It is not clear how counting natural number results in the instances for $F(2)$.

SPASS Herbrand Instances $F(1)$

$$\textbf{1:}\ \forall A_0\, \forall B\, \forall C \begin{array}{c} \langle\, g^2(U)\ ,\ g(U)\ ,\ \mathrm{max}(g^2(U), g(U))\,\rangle \\ \langle\, g(U)\ ,\ g^2(U)\ ,\ \mathrm{max}(g(U), g^2(U))\,\rangle \\ \langle\, g(U)\ ,\ g(U)\ ,\ \mathrm{max}(g(U), g(U))\,\rangle \end{array} (\,\neg\, \mathrm{LEQ}(\mathrm{max}(A_0, B), C) \lor \mathrm{LEQ}(B, C)\,)$$

$$\textbf{2:}\ \forall A_0\, \forall B\, \forall C \begin{array}{c} \langle\, g^2(U)\ ,\ g(U)\ ,\ \mathrm{max}(g^2(U), g(U))\,\rangle \\ \langle\, g(U)\ ,\ g^2(U)\ ,\ \mathrm{max}(g(U), g^2(U))\,\rangle \\ \langle\, g(U)\ ,\ g(U)\ ,\ \mathrm{max}(g(U), g(U))\,\rangle \end{array} (\,\neg\, \mathrm{LEQ}(\mathrm{max}(A_0, B), C) \lor \mathrm{LEQ}(A_0, C)\,)$$

$$\textbf{3:}\ \forall A \begin{array}{c} \langle\, g(U)\,\rangle \\ \langle\, \mathrm{max}(g(U), g(U))\,\rangle \\ \langle\, U\,\rangle \\ \langle\, \mathrm{max}(g^2(U), g(U))\,\rangle \\ \langle\, \mathrm{max}(g(U), g^2(U))\,\rangle \end{array} (\, E(f(A), s(0)) \lor E(f(A), 0)\,)$$

$$\textbf{4:}\ \forall A \begin{array}{c} \langle\, g(U)\,\rangle \\ \langle\, \mathrm{max}(g(U), g(U))\,\rangle \\ \langle\, \mathrm{max}(g(U), g^2(U))\,\rangle \\ \langle\, \mathrm{max}(g^2(U), g(U))\,\rangle \end{array} \mathrm{LEQ}(A, A)$$

$$\textbf{5:}\ \forall B_1\, \forall A_2 \begin{array}{c} \langle\, U\ ,\ \mathrm{max}(g^2(U), g(U))\,\rangle \\ \langle\, U\ ,\ g(U)\,\rangle \\ \langle\, U\ ,\ \mathrm{max}(g(U), g(U))\,\rangle \\ \langle\, g(U)\ ,\ \mathrm{max}(g(U), g^2(U))\,\rangle \end{array} (\,(\,\neg\, \mathrm{LEQ}(g(B_1), A_2) \lor \neg\, E(f(B_1), s(0))\,) \lor \neg\, E(f(A_2), s(0))\,)$$

$$\textbf{6:}\ \forall B_0\, \forall A_1 \begin{array}{c} \langle\, U\ ,\ g(U)\,\rangle \\ \langle\, U\ ,\ \mathrm{max}(g(U), g(U))\,\rangle \\ \langle\, g(U)\ ,\ \mathrm{max}(g^2(U), g(U))\,\rangle \\ \langle\, U\ ,\ \mathrm{max}(g(U), g^2(U))\,\rangle \end{array} (\,(\,\neg\, \mathrm{LEQ}(g(B_0), A_1) \lor \neg\, E(f(B_0), 0)\,) \lor \neg\, E(f(A_1), 0)\,)$$

- Even simpler...

The relationship between π_n and π_{n+1}

- Our example instance sets for $F(1)$ and $F(2)$ illustrate that the various proofs are not related.
- Thus, if we give the proofs to Viper the chance it will find an invariant is around 0.
- Can we develop a prover which generates sequences of proofs which are "Uniform".
- What do we mean by "uniform" anyway, What is "relatedness".
- Mathematically, are we trying to find proofs which use a particular trick and/or method.

Proposal: Can Modern Machine Learning Help?

- This is not a question about theorem proving, rather it is a "mathematical understanding"?
- Can we get the Theorem prover to understand what it ought to look for while constructing π_{n+1} using the proofs produced for π_n and below?
- We know the prover can prove π_{n+1}, but can it prove it in the right way!
- As mentioned earlier, this work is in its infancy.
  - A) I believe modern machine learning method may help solve the "uniformity" problem.
  - B) I don't know how they might help, maybe you do?
  - C) If interested and think you might have an idea, I would love to discuss it.
  - D) Currently looking for collaboration for a proposal I am developing.

Thank you for your time.