Program correctness on finite fields*

László Csirmaz

Mathematical Institute of the Hungarian Academy of Sciences
Reáltanoda u. 13-15, Budapest, Hungary, H-1053

Abstract

An asserted program is presented whose correctness is provable by the Floyd-Hoare-Naur method in each finite field separately, which, however, admits no universal derivation, i.e. one which would work on all but finitely many finite fields of a given characteristic. Also, it is proved in general that if "executing a program twice with the same input, the outputs agree" is a provable property, then the output of the program is first order definable from the input.

Key words. Finite fields, pseudofinite fields, program correctness, Floyd-Hoare derivation.

1 Introduction

In the last 30 years extended research was devoted to clarify the exact power of different program verification methods. It has turned out that quite a number of the famous methods have nice model theoretic characterizations [4, 6, 8]. The characterizations, as a byproduct, give simple and universal methods for proving the unprovability of a partial correctness assertion by this or that method. The greatest effort, however, was put on the most profound method, the so called Naur-Floyd-Hoare method of intermediate assertions. The results are mainly negative, showing the strong incompleteness nature of Floyd's method.

In this paper by a *program* we mean a block diagram, regular, or while program of a given signature — whichever is closer to the reader's heart. However, recursive program schemes or programs with recursive procedures are excluded since they are untractable by our methods [5]. The assertions are simple first order formulas of the same signature, and the *partial correctness assertion* $\{\varphi_{\text{in}}\}p\{\varphi_{\text{out}}\}$ states that whenever the input variables of the program p satisfy the input assertion φ_{in} and the program halts, then the output variables of p will satisfy the output condition φ_{out}. We say that $\{\varphi_{\text{in}}\}p\{\varphi_{\text{out}}\}$ *admits a Floyd-Hoare derivation* in a structure (or model) $\mathbf{A}$ if it admits such a derivation (in sense of, e.g. [3]) in which all the oracle axioms used are elements of $\text{Th}(\mathbf{A})$, the set of sentences true in $\mathbf{A}$.

One of the rare completeness-type theorems about the Floyd-Hoare derivability is the following.

*A preliminary version of this paper appeared in the Proceedings of the Symposium on Logic in Computer Science, Cambridge, Mass. 1986, pp. 4–10

Theorem 1.1 *If an asserted program admits a Floyd-Hoare derivation in each model of a given theory T, then there exists a single "universal" derivation which works in all the models.*

While this theorem is an immediate consequence of the above-mentioned characterizations, for the convenience of the reader we shall give a simple and direct proof.

From both theoretical and practical points of view finite structures play an important role. Modeling computation in a finite structure is much less controversial than doing the same in infinite models. Moreover, on finite structures, partially correct and Floyd-Hoare provable asserted programs coincide. Thus in light of Theorem 1.1 the following question arises. Suppose that, instead of all models of T, we require derivability on its *finite* models only. Does it follow then that we have a universal derivation which works on all, or at least on infinitely many, of the finite models?

Using J. Ax's deep and nice result about the decidability of the theory of finite fields [1, 2], we prove that the answer is *no* if T is the theory of finite fields. More precisely, let p be the program with input variable x and output variable y which computes the parity of the multiplicative order of its input. That is, $y = 0$ if the smallest natural number $n \geq 1$ satisfying $x^n = 1$ is even, $y = 1$ if this n is odd, and the program diverges if for no $n \geq 1$ we have $x^n = 1$. Executing the program twice with the same input, our natural expectation is that the outputs are the same. If p' is the same program as p except that every variable is "primed," then this can be expressed by saying that the asserted program

$$\{x = x'\}\ p; p'\ \{y = y'\} \tag{1}$$

is partially correct, and consequently, Floyd-Hoare provable in each finite field.

Theorem 1.2 *Let $q \geq 2$ be a prime number. There is no Floyd-Hoare derivation of the asserted program* (1) *which works on all but finitely many finite fields of characteristics q.*

The multiplicative group of a finite field is cyclic, therefore in fields with 2^n elements every nonzero element is of odd order. In these fields p always halts with $y = 1$, therefore we have

Corollary 1.3 *The asserted program $\{x \neq 0\}\, p\, \{y = 1\}$, while totally correct, admits no universal Floyd-Hoare derivation on finite fields of characteristics 2.*

The existence of such a program follows immediately from the fact that the set of asserted programs partially correct in every field is not recursive, while, by Ax's result, the set of those admitting universal Floyd-Hoare derivation in finite fields is recursive. What makes these theorem interesting is the simple structure of both the program and the assertions. If we require universal derivation not for all but for infinitely many finite fields only, the set of derivable asserted programs is no more recursive. However, we conjecture that the claim of Corollary 1.3 remains valid in this stronger sense, too, but we were unable to settle it.

Conjecture 1.4 *The asserted program $\{x \neq 0\}\, p\, \{y = 1\}$ does not admit Floyd-Hoare derivation which would work on infinitely many finite fields of characteristic 2.*

Finally, we investigate when programs like (1) admit universal Floyd-Hoare derivation. It turns out that in that case a first order formula, with the input variables as parameters, determines uniquely the output of p. The converse, as Corollary 1.3 shows, is not necessarily true. In general, we may require at least n matches among the outputs of k executions ($k \geq n$). If this is a provable property of the program p, then by the same method the existence of a first order formula $\varphi(x, y)$ can be shown so that for each x at most $k + 1 - n$ different y satisfy it, and the output of p is always among these y's ($k = n = 2$ above).

2 Prerequisites

2.1 Programs

Let t be any similarity type or signature. $F(t)$ denotes the set of first order formulas of type t. For a fixed theory $T \subseteq F(t)$, by a *program* (in T) we mean a first order formula $\varphi(x, y) \in F(t)$ with $n + n$ free variables so that

$$T \models \forall x \, \exists! y \, \varphi(x, y).$$

For simplicity, we let x, y, etc. denote tuples of variables. Thus we identify programs with definable functions; these functions describe a state transition (i.e. the result of executing a single computational step) rather than the input-output relation. We left it to the reader to verify that this notion of program covers that of block diagram, regular, and while programs. The only reason why we use this definition is that it simplifies significantly the formalism and makes the proofs more transparent.

We shall use the letter p to denote programs, and write $x \, p \, y$ for "y is the successor state of x," which is therefore a first order formula of type t with free variables x and y. Moreover we stipulate that the halting states of p are just the fixed points of the transition function, i.e. x is a halting state iff $x \, p \, x$. Naturally, these are not essential restriction since given any reasonable halting condition, it is a matter of routine to change the successor state function to one which obeys our rules.

Motivations, as well as a description how to translate, say while programs, into our kind of programs, can be found in [4]. It is also proved that the asserted program $\{\varphi_{\text{in}}\} \, p \, \{\varphi_{\text{out}}\}$ with $\varphi_{\text{in}}(x)$, $\varphi_{\text{out}}(x) \in F(t)$ admits a Floyd-Hoare derivation with oracle axioms from the theory $T \subset F(t)$, written as $T \vdash \{\varphi_{\text{in}}\} \, p \, \{\varphi_{\text{out}}\}$, iff there exists a formula $\Phi(x) \in F(t)$ so that

$$\begin{aligned} &T \vdash \varphi_{\text{in}}(x) \to \Phi(x) \\ &T \vdash \Phi(x) \wedge x \, p \, y \to \Phi(y) \\ &T \vdash \Phi(x) \wedge x \, p \, x \to \varphi_{\text{out}}(x). \end{aligned}$$

Thus $\{\varphi_{\text{in}}\} \, p \, \{\varphi_{\text{out}}\}$ is Floyd-Hoare derivable in the model $\mathbf{A}$ of T iff the oracle axioms are the sentences true in $\mathbf{A}$, i.e. iff $\text{Th}(\mathbf{A}) \vdash \{\varphi_{\text{in}}\} \, p \, \{\varphi_{\text{out}}\}$.

While we shall use these notions for reasoning about programs, we do not use them in examples. For that purpose we choose the more readable while program form.

2.2 Finite fields

Let q be a natural number which is a power of some prime. It is well known that there is exactly one (up to isomorphism) finite field with q elements, we shall denote it by F_q. It is also a (finite) structure of type $t = \langle 0, 1, +, -, \cdot \rangle$. Let $\mathrm{Tp} \subset F(t)$ consist of all formulas valid in cofinitely many finite fields; this theory is obviously consistent. Models of Tp are the so-called *pseudofinite fields.* In particular, every nonprincipal ultraproduct of finite fields is pseudofinite, moreover every pseudofinite field is infinite.

The cardinality of a set A will be denoted by $|A|$, ω denotes the set of nonnegative natural numbers, and $\mathbb{Z}$ the set of integers.

The following facts about pseudofinite fields are from [1].

Theorem 2.1 *The theory of pseudofinite fields is recursively enumerable.*

Theorem 2.2 *Let F_1 and F_2 be saturated pseudofinite fields, both extensions of the field E which is algebraically closed both in F_1 and F_2. If $|F_1| = |F_2| > |E|$ then there is an isomorphism between F_1 and F_2 which leaves E fixed elementwise.*

Such an isomorphism will be called *E-isomorphism.*

For a set E, let $E[x_1, \ldots, x_n]$ denote the set of polynomials with variables x_1, …, x_n and coefficients from E. In particular, $\mathbb{Z}[x]$ is the set of all polynomials with the single variable x and integral coefficients. The following lemma is from [2], but it can also be verified directly.

Lemma 2.3 *Let F_1 and F_2 be extensions of E, moreover let $E_i \subseteq F_i$ be the relatively algebraic closure of E in F_i $(i = 1, 2)$. If*

$$\{f \in E[x] : f(x) = 0 \textit{ has a root in } F_1\} =$$
$$\{f \in E[x] : f(x) = 0 \textit{ has a root in } F_2\}$$

then there is an E-isomorphism between E_1 and E_2.

Next we state the corollary we need later.

Corollary 2.4 *Let F be uncountable saturated pseudofinite field, $\alpha \in F$, $E \subseteq F$ be the subfield generated by α. Let moreover β_1, $\beta_2 \in F$ be two elements with the following property:*

$$\{f \in \mathbb{Z}[x, y, z] : f(x, \alpha, \beta_1) = 0 \textit{ has a solution in } F\} =$$
$$\{f \in \mathbb{Z}[x, y, z] : f(x, \alpha, \beta_2) = 0 \textit{ has a solution in } F\}.$$

Then there is an E-automorphism π of F which maps β_1 to β_2.

Proof. Let E_1, E_2 be the subfields generated by α and β_1, α and β_2, respectively. Then by the assumption on polynomials, the natural mapping which extends $\pi(\beta_1) = \beta_2$, $\pi(\alpha) = \alpha$ is an E-isomorphism between E_1 and E_2; moreover for every $f \in E_1[x]$,

$$f(x) = 0 \text{ has a solution in } F$$

if and only if

$$\pi(f)(x) = 0 \text{ has a solution in } F.$$

Thus if E_i^* denotes the relatively algebraic closure of E_i in F, then, by Lemma 2.3, π extends to an isomorphism between E_1^* and E_2^*. Now E_1^* (and so E_2^* too) is countable, therefore by Theorem 2.2 any isomorphism between E_1^* and E_2^* extends to an automorphism of F. Especially this is true for π as was claimed. ■

3 Proofs

In this section we prove the theorems announced in the Introduction. First let t be a similarity type, $T \subset F(t)$ be a theory and p be a program in T. The asserted program $\{\varphi_{\rm in}\}\,p\,\{\varphi_{\rm out}\}$ admits a (universal) Floyd-Hoare derivation which works in all models of T just in case all the oracle axioms used are consequences of T, i.e. if $T \vdash \{\varphi_{\rm in}\}\,p\,\{\varphi_{\rm out}\}$. Theorem 1.1 can be reworded as follows.

Theorem 3.1 *Suppose that for each model* $\mathbf{A}$ *of* T *we have* $\mathrm{Th}(\mathbf{A}) \vdash \{\varphi_{\rm in}\}\,p\,\{\varphi_{\rm out}\}$. *Then* $T \vdash \{\varphi_{\rm in}\}\,p\,\{\varphi_{\rm out}\}$.

Proof. For a formula $\Phi(x) \in F(t)$ let us define the *closure* of Φ as

$$\mathrm{cl}(\Phi) = \forall x(\varphi_{\rm in}(x) \to \Phi(x)) \wedge \forall x \forall y(\Phi(x) \wedge x\,p\,y \to \Phi(y)) \wedge \forall x(\Phi(x) \wedge x\,p\,x \to \varphi_{\rm out}(x)).$$

This is a closed formula (i.e. has no free variables), and the hypothesis of the theorem says that in each model of T at least one element of the set $\Sigma = \{\mathrm{cl}(\Phi) : \Phi(x) \in F(t)\}$ is true. By the compactness theorem for the first order logic then there is a finite subset $\{\mathrm{cl}(\Phi_1), \ldots, \mathrm{cl}(\Phi_n)\}$ of Σ so that for each model $\mathbf{A}$ of T, $\mathbf{A} \models \mathrm{cl}(\Phi_i)$ for some $i \le n$. Now we claim that the formula

$$\Psi(x) = (\mathrm{cl}(\Phi_1) \wedge \Phi_1(x)) \vee \ldots \vee (\mathrm{cl}(\Phi_n) \wedge \Phi_n(x))$$

witnesses the derivation $T \vdash \{\varphi_{\rm in}\}\,p\,\{\varphi_{\rm out}\}$. To this end we have to show that the formulas

$$\varphi_{\rm in}(x) \to \Psi(x) \tag{2}$$

$$\Psi(x) \wedge x\,p\,y \to \Psi(y) \tag{3}$$

$$\Psi(x) \wedge x\,p\,x \to \varphi_{\rm out}(x) \tag{4}$$

are consequences of T, or, which is the same, are valid in every model $\mathbf{A}$ of T. Thus let $\mathbf{A}$ be such a model, then $\mathbf{A} \models \mathrm{cl}(\Phi_i)$ for some $i \le n$. Clearly, $\mathrm{cl}(\Phi_i) \wedge \varphi_{\rm in}(a) \vdash \Phi_i(a)$ for all $a \in A$. This proves (2) since if $\mathbf{A} \models \varphi_{\rm in}(a)$ for some $a \in A$ then $\mathbf{A} \models \mathrm{cl}(\Phi_i) \wedge \Phi_i(a)$, i.e. at least one disjunct of $\Psi(a)$ is true. Next suppose $\mathbf{A} \models \Psi(a)$. Then for some $j \le n$ we have $\mathbf{A} \models \mathrm{cl}(\Phi_j) \wedge \Phi_j(a)$, and

$$\mathrm{cl}(\Phi_j) \wedge \Phi_j(a) \vdash a\,p\,b \to \Phi_j(b),$$

this gives (3); finally

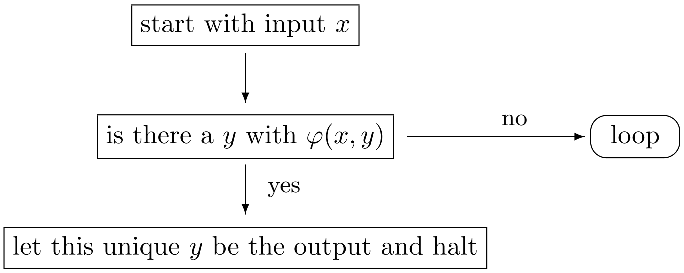
$$\mathrm{cl}(\Phi_j) \wedge \Phi_j(a) \vdash a\,p\,a \to \varphi_{\rm out}(a)$$

shows that (4) is also true in $\mathbf{A}$. The theorem is proved. ∎

Not going into the special case of finite fields yet, we investigate when it is a provable property of a program that executing twice the results coincide. Let therefore $T \subset F(t)$ be a fixed theory, p be a program in T, and p' be a "primed" version of p. Suppose p and p' act on the disjoint set of variables x and x' respectively; our assumption says that the partial correctness assertion $\{x = x'\}\ p; p'\ \{x = x'\}$ is Floyd-Hoare derivable with oracle axioms from T. We say that the output of p is *definable* in T from its input if there exists a formula $\varphi(x, y) \in F(t)$ with $2n$ variables so that

$$T \vdash \varphi(x, y) \wedge \varphi(x, y') \to y = y'$$

i.e. for each x at most one y satisfies $\varphi(x,y)$, and whenever p starts with input value x and halts with output value y, they satisfy $\varphi(x,y)$. In other words, in arbitrary model of T, the input/output relation of the straight-line program


start with input x
is there a y with $\varphi(x,y)$
no
loop
yes
let this unique y be the output and halt

extends the input/output relation of p.

Theorem 3.2 *If we can prove (in T) that executing p twice the results coincide, then the output of p is definable.*

Proof. Suppose $T \vdash \{x = x'\}\ p; p'\ \{x = x'\}$, i.e. the following formulas are consequences of T for some $\Phi(x,x') \in F(t)$:

$$\Phi(x,x) \tag{5}$$

$$x\,p\,y \wedge \Phi(x,x') \rightarrow \Phi(y,x') \tag{6}$$

$$x\,p\,x \wedge x'\,p\,y' \wedge \Phi(x,x') \rightarrow \Phi(x,y') \tag{7}$$

$$x\,p\,x \wedge x'\,p\,x' \wedge \Phi(x,x') \rightarrow x = x'. \tag{8}$$

Let $\psi(x,y) = y\,p\,y \wedge \Phi(y,x)$, and $\varphi(x,y) = \psi(x,y) \wedge \forall y'(\psi(x,y') \rightarrow y' = y)$. Obviously, for each x at most one y can satisfy $\varphi(x,y)$. We claim that this φ defines the output of p. To check this, let $\mathbf{A}$ be a model of T, $a \in A^n$ be the input and $b \in A^n$ be the output of a run of p; we have to show $\mathbf{A} \models \varphi(a,b)$. Since b is a halting state, we have $\mathbf{A} \models b\,p\,b$, moreover by (5), $\mathbf{A} \models \Phi(a,a)$, and then by (6), $\mathbf{A} \models \Phi(a_i,a)$ for each state $a_i \in A^n$ occurring during the execution; in particular $\mathbf{A} \models \Phi(b,a)$. This proves $\mathbf{A} \models \psi(a,b)$, the first half what we wanted.

For the other half assume $\mathbf{A} \models \psi(a,b')$, i.e. $\mathbf{A} \models b'\,p\,b' \wedge \Phi(b',a)$. Then by (7) $\mathbf{A} \models \Phi(b',a_i)$ for each state a_i, from where $\mathbf{A} \models \Phi(b',b)$. Now applying (8) we get $b' = b$ as was required. ∎

In fact, the proof gives a bit more than stated. There is a useful generalization of the notion of program run, the so-called *relational run* [4, 7, 8]. Since we shall need this notion later, we give a sketchy definition. As above, we identify states with n-tuples of the ground set of the structure $\mathbf{A}$.

3.3 Definition The set of states $R \subset A^n$ with the distinguished state $a \in R$ constitutes a *relational run* for the program p if

(i) R is closed under p, i.e. $b \in R$ and $b\,p\,c$ implies $c \in R$;

(ii) R is *inductive*, i.e. for each formula $\Phi(x) \in F(t)$, if $\mathbf{A} \models \Phi(a)$ and for each $b,\ c \in R$, $\mathbf{A} \models b\,p\,c \wedge \Phi(b) \to \Phi(c)$ then Φ is true for all elements of R.

The distinguished state $a \in R$ is the *initial state*, and $b \in R$ is a *halting state* if $b\,p\,b$. A relational run may have more than one halting states, nevertheless partial correctness can also be defined in the obvious way. The significance of this notion is expressed by the following theorem of [4, 7]. Later we shall use the easy part only.

Theorem 3.4 *The asserted program $\{\varphi_{\text{in}}\}\,p\,\{\varphi_{\text{out}}\}$ is Floyd-Hoare derivable from T if and only if in models of T every relational run of p is partially correct.*

The proof of Theorem 3.2 gives that under the same assumptions even relational runs have a unique, definable halting state.

Theorem 3.4 offers a universal method to establish unprovability of asserted programs: one has to find an incorrect relational run. This is exactly what we shall do in proving Theorem 1.2. So from now on let t denote the signature $\langle 0, 1, +, -, \cdot \rangle$ of fields. The program below computes the parity of the multiplicative order of its input x twice and puts the results into y and y', respectively.

$$\begin{array}{l} y := 1;\ y' := 1;\ z := x,\ z' := x; \\ \textbf{while } z \neq 1 \textbf{ do } z := x \cdot z;\ y := 1 - y \textbf{ od}; \\ \textbf{while } z' \neq 1 \textbf{ do } z' := x \cdot z';\ y' := 1 - y' \textbf{ od}. \end{array}$$

The states can be identified with the 5-tuples consisting of the contents of the variables x, y, z, y', and z'. Disregarding the initializations in the first line, in this case the state transition function is

$$p(x, y, z, y', z') = \begin{cases} (x, 1 - y, x \cdot z, y', z') & \text{if } z \neq 1, \\ (x, y, z, 1 - y', x \cdot z') & \text{if } z = 1 \text{ and } z' \neq 1, \\ (x, y, z, y', z') & \text{otherwise.} \end{cases}$$

This function is evidently formula-definable, and the halting states are just its fixed points. Thus p is a program as defined in Section 2.1. With this transcription we have avoided a lot of technical trouble arising otherwise. Now we prove Theorem 1.2 in the following form.

Theorem 3.5 *Let $q \geq 2$ be a prime number. The there is no Floyd-Hoare derivation of the asserted program*

$$\{y = 1 \wedge y' = 1 \wedge x = z \wedge x = z'\}\,p\,\{y = y'\} \tag{9}$$

which would work in all but finitely many finite fields of characteristic q.

Proof. All fields in this proof are of characteristic q. Suppose the claim of the theorem false, i.e that (9) admits a universal derivation. The in every ultraproduct of those finite fields the same proof works, and by Theorem 3.4 only partially correct relational runs can exist in the ultraproduct. So we prove the theorem by exhibiting an incorrect relational run in an appropriately chosen ultraproduct.

Without loss of generality we may assume that the continuum hypothesis holds. Indeed, if this were not the case then we can work in a generic extension which collapses the continuum to

ω_1. Since the theorem speaks about finite objects only, the statement is absolute, i.e. it holds in the generic extension if and only if it holds in the ground model.

As we remarked in Section 2.2, any nonprincipal ultraproduct of finite fields is pseudofinite. Let F be such an ultraproduct, it is well known that F has continuumly many elements, and since the continuum hypothesis holds, it is saturated. Pick $\alpha \in F$ transcendental over the prime field of F (only countably many algebraic elements are in F, consequently such an α exists), and consider the following sets of 5-tuples from F:

$$\begin{aligned} R_1 &= \{\langle \alpha, \mathrm{pt}(n), \alpha^n, 1, \alpha\rangle \,:\, n \in \omega \text{ and } n > 0\},\\ R_2 &= \{\langle \alpha, \mathrm{pt}(n), \alpha^{-n}, 1, \alpha\rangle \,:\, n \in \omega\} \cup \{\langle \alpha, 0, 1, \mathrm{pt}(n), \alpha^n\rangle \,:\, n \in \omega \text{ and } n > 0\},\\ R_3 &= \{\langle \alpha, 0, 1, 1 - \mathrm{pt}(n), \alpha^{-n}\rangle \,:\, n \in \omega\},\end{aligned}$$

and, finally, let

$$R = R_1 \cup R_2 \cup R_3.$$

Here $\mathrm{pt}(n)$ is the parity of $n \in \omega$, i.e. $\mathrm{pt}(n) = 0$ if n is even, and $\mathrm{pt}(n) = 1$ if n is odd. Obviously, R consists of states for the program p, is closed under p, i.e. $b \in R$ implies $p(b) \in R$; and R has the only halting state $\langle \alpha, 0, 1, 1, 1\rangle$. Since the initial state $a = \langle \alpha, 1, \alpha, 1, \alpha\rangle \in R$ satisfies the input condition of (9), and the only halting state falsifies the output condition, we are home if R is a relational run. (i) of Definition 3.3 evidently holds, only the inductivity is questionable. So let $\Phi(x) \in F(t)$ be a formula with 5 free variables, and assume $F \models \Phi(a)$, moreover for each $b \in R$, $F \models \Phi(b) \to \Phi(p(b))$. Then Φ holds for the tuples in R_1 since they can be obtained from a applying p finitely many times. Next, R_2 consists of a single chain of states, so if Φ is not true for all elements in R_2 then it is false for a whole initial segment, i.e. for some $n_0 \in \omega$ we have

$$F \models \neg\Phi(\alpha, \mathrm{pt}(n), \alpha^{-n}, 1, \alpha) \quad \text{for all } n \geq n_0,$$

in particular

$$F \models \neg\Phi(\alpha, 0, \alpha^{-2n}, 1, \alpha) \quad \text{for all } n \geq n_0. \tag{10}$$

At the same time we know that

$$F \models \Phi(\alpha, 0, \alpha^{2n}, 1, \alpha) \quad \text{for all } n > 1. \tag{11}$$

Let E be the subfield of F generated by α. Since F is saturated, there exists an element $\beta \in F$ with the property that for any formula $\psi(x, y) \in F(t)$ and parameters p from E, if $F \models \psi(\alpha^{2n}, p)$ for all but finitely many $n \in \omega$ then $F \models \psi(\beta, p)$. In particular, by (10) and (11),

$$F \models \neg\Phi(\alpha, 0, \beta^{-1}, 1, \alpha),$$

and

$$F \models \Phi(\alpha, 0, \beta, 1, \alpha).$$

This is a contradiction if there is an E-automorphism of F interchanging β and β^{-1}, which would prove that Φ holds indeed for tuples in R_2.

Suppose for a moment that this automorphism exists. Then

$$F \models \Phi(\alpha, 0, 1, 0, \alpha^{2n}) \quad \text{for } n \geq 1 \tag{12}$$

since all of these 5-tuples are in R_2. Just as previously, if Φ does not hold for some element in R_3 then it is false for an initial segment, i.e. for some $n_1 \in \omega$,

$$F \models \neg\Phi(\alpha, 0, 1, 0, \alpha^{-2n+1}) \quad \text{for } n \geq n_1. \tag{13}$$

(12) and (13) gives $F \models \Phi(\alpha, 0, 1, 0, \beta)$ and $F \models \neg\Phi(\alpha, 0, 1, 0, \alpha\beta^{-1})$, which is impossible if there is an E-automorphism of F sending β to $\alpha\beta^{-1}$.

Summing up, R is the incorrect relational run we are looking for if there are E-automorphisms π_1 and π_2 of F with $\pi_1(\beta) = \beta^{-1}$ and $\pi_2(\beta) = \alpha\beta^{-1}$. So we have to choose the ultraproduct F and the element $\alpha \in F$ according to this requirement.

Since F is uncountable and saturated, we can apply Corollary 2.4 which says that these automorphisms exist if

$$\begin{aligned} &\{f \in \mathbb{Z}[x,y,z] \,:\, f(x,\alpha,\beta) = 0 \text{ has a solution in } F\} = \\ &\{f \in \mathbb{Z}[x,y,z] \,:\, f(x,\alpha,\beta^{-1}) = 0 \text{ has a solution in } F\} = \\ &\{f \in \mathbb{Z}[x,y,z] \,:\, f(x,\alpha,\alpha\beta^{-1}) = 0 \text{ has a solution in } F.\} \end{aligned} \tag{14}$$

Now α was chosen to be transcendental over the prime field of F, which means that the different powers of α are different Any polynomial $f \in E[x]$ has only finitely many roots, therefore $f(\alpha^{2n}) \neq 0$ for all but finitely many n. By the definition of $\beta \in F$ this means that $f(\beta) \neq 0$, i.e. β is transcendental over E, and then for every $f \in \mathbb{Z}[y,z]$,

$$f(\alpha,\beta) = 0 \text{ iff } f(\alpha,\beta^{-1}) = 0 \text{ iff } f(\alpha,\alpha\beta^{-1}) = 0 \text{ iff } f \text{ is identically zero.}$$

Therefore the sets in (14) share the same members in $\mathbb{Z}[y,z]$, and their equality follows immediately if $f(x,\alpha,\beta) = 0$ is solvable for every other polynomial in $\mathbb{Z}[x,y,z]$. This latter is a consequence of the (apparently weaker) condition that $g(x,\alpha) = 0$ is solvable (in F) for every $g \in \mathbb{Z}[x,y] \setminus \mathbb{Z}[y]$. Indeed, fixing $f \in \mathbb{Z}[x,y,z] \setminus \mathbb{Z}[y,z]$,

$$F \models \exists x\, f(x,\alpha,\alpha^n) = 0 \quad \text{for every } n \geq 1,$$

and then, by the definition of $\beta \in F$, we have

$$F \models \exists x\, f(x,\alpha,\beta) = 0,$$

i.e. $f(x,\alpha,\beta) = 0$ is solvable.

To finish the proof, we have to find a nonprincipal ultraproduct F of those finite fields of characteristic q in which the universal Floyd-Hoare derivation of our asserted program works, and an element $\alpha \in F$ so that

(i) α is transcendental over the prime field of F;

(ii) for each $g \in \mathbb{Z}[x,y] \setminus \mathbb{Z}[y]$, $g(x,\alpha) = 0$ is solvable in F.

Let $\{f_j \,:\, j \in \omega\}$ enumerate the nontrivial polynomials in $\mathbb{Z}[y]$, and $\{g_j \,:\, j \in \omega\}$ enumerate the set $\mathbb{Z}[x,y] \setminus \mathbb{Z}[y]$. For $i \in \omega$ define the finite field F_i of characteristic q and element $\alpha_i \in F_i$ as follows. First find a field F'_i and an element $\alpha_i \in F'_i$ such that α_i is not the solution of any

$f_j(y) = 0$ with $j \leq i$. Second, take an algebraic extension F_i of F_i' in which all $g_j(x, \alpha_i) = 0$ has a solution for $j \leq i$. Since by assumption only finitely many of the fields do not respect our derivation, we may choose F_i to be one in which the Floyd-Hoare derivation works. In any nontrivial ultraproduct F of these F_i's the element $\alpha \in F$ with coordinates $\alpha_i \in F_i$ satisfies (i) and (ii), so the theorem is proved. ∎

References

[1] J. Ax, Solving Diophantine problems modulo every prime, *Annals of Mathematics*, Vol 85(1967) pp. 161-183

[2] J. Ax, The elementary theory of finite fields, *Annals of Mathematics*, Vol 88(1968) pp. 239-271

[3] J. W. de Bakker, em Mathematical theory of program correctness (1980), Prentice-Hall, London

[4] L. Csirmaz, Programs and program verification in a general setting, *Theoretical Computer Science*, Vol 16(1981) pp. 199-210

[5] A. J. Kfoury, Definability by programs in first-order structures, *Theoretical Computer Science*, Vol 25(1983) pp. 1-66

[6] I. Németi, Nonstandard dynamic logic *Proceedings of Logics of Programs*, Lecture Notes in Computer Science, Vol 131(1982), D. Kozen ed. Springer-Verlag, Berlin, pp. 311-348

[7] I. Sain, A simple proof for the completeness of Floyd's method *Theoretical Computer Science*, Vol 35(1985) pp.345-348

[8] I. Sain, The reasoning power s of Burstall's and Pnueli's program verification methods, *Proceedings of Logics of Programs*, Lecture Notes in Computer Science, Vol 193(1982), Springer-Verlag, Berlin, pp. 302-319