On Bounded Exploration and Bounded Nondeterminism

Yuri Gurevich and Tatiana Yavorskaya

January 2006

Abstract

This report consists of two separate parts, essentially two oversized footnotes to the article “Sequential Abstract State Machines Capture Sequential Algorithms” by Yuri Gurevich.

In Chapter I, Yuri Gurevich and Tatiana Yavorskaya present and study a more abstract version of the bounded exploration postulate.

In Chapter II, Tatiana Yavorskaya gives a complete form of the characterization, sketched in the original paper, of bounded-choice sequential algorithms.

Chapter 1

A more abstract bounded exploration postulate

Yuri Gurevich and Tatiana Yavorskaya

Abstract

In his article “Sequential Abstract State Machines Capture Sequential Algorithms”, Gurevich defines a sequential algorithm by means of three postulates: sequential time, abstract state, and bounded exploration postulates. Here we give another bounded exploration postulate such that (a) the new postulate is more abstract and is closer in spirit to the abstract state postulate than the original one, and (b) in the presence of the sequential time and abstract state postulates, the new bounded exploration postulate is equivalent to the original.

1.1 Introduction

According to [1], a sequential algorithm is any object A satisfying the sequential time postulate, the abstract state postulate and the bounded exploration postulate. We presume that the reader is familiar with [1]. But, for reader's convenience, we restate the three postulates.

Postulate 1 (Sequential Time)**.** A is associated with

- a nonempty set $\mathcal{S}(A)$ whose elements will be called *states* of A,
- a nonempty subset $\mathcal{I}(A)$ of $\mathcal{S}(A)$ whose elements will be called *initial states* of A, and
- a map $\tau_A : \mathcal{S}(A) \longrightarrow \mathcal{S}(A)$ that will be called the *one-step transformation* of A.

Remark 1. The original version of the postulate in [1] did not require that $\mathcal{S}$ and $\mathcal{I}$ be nonempty. The reasonable modification is due to [2].

Postulate 2 (Abstract State)**.**

- States of A are first-order structures.
- All states of A have the same vocabulary.
- The one-step transformation τ_A does not change the base set of any state.
- $\mathcal{S}(A)$ and $\mathcal{I}(A)$ are closed under isomorphisms. Further, any isomorphism from a state X to a state Y is also an isomorphism from $\tau_A(X)$ to $\tau_A(Y)$.

Postulate 3 (Bounded Exploration)**.** There exists a finite set T of terms in the vocabulary of A such that the update set $\Delta(A, X)$ of A at X coincides with the update set $\Delta(A, Y)$ of A at Y whenever states X, Y of A coincide over T.

The bounded exploration postulate is convincing but it contradicts the spirit of the abstract state postulate according to which a state is just a presentation of its isomorphism type so that only the isomorphism type of the state is important. In the bounded exploration postulate above, it is essential that the states X and Y are concrete. The purpose of this note is to give a more abstract form of the bounded exploration postulate that is in the spirit of the abstract state postulate and that is equivalent to the original bounded exploration postulate in the presence of the sequential time and abstract state postulates.

1.2 Some auxiliary definitions

Let Υ be a vocabulary as in [1]. It contains the logical names true, false, undef, the equality sign, and the standard propositional connectives. All other names in Υ are nonlogical. In any (first-order) Υ-structure, the values of true, false and undef are distinct logical elements; all other elements are nonlogical. Let X and Y be Υ-structures, and let T be a set of Υ-terms closed under subterms.

Definition 1. If f is a function symbol in Υ and t is an Υ-term, then f_X is the interpretation of f in X, $\mathsf{Val}_X(t)$ is the value of t in X, and $X \restriction T$ is the set $\{\mathsf{Val}_X(t) \;:\; t \in T\}$.

A binary relation can be viewed as a set of pairs.

Definition 2. Relation $\{(\mathsf{Val}(t, X), \mathsf{Val}(t, Y)) :\; t \in T\}$ is the *T-similarity relation* R^T between X and Y.

Definition 3. The *T-similarity type* of X is the equivalence $\sim_X$ relation on T where

$$s \sim_X t \iff \mathsf{Val}(s, X) = \mathsf{Val}(t, X)$$

Structures X and Y are *T-similar* if they have the same T-similarity type.

If for $x \in X \restriction T$ there is a unique $y \in Y \restriction T$ such that $x \, R^T \, y$, then R^T is the graph of a function from $X \restriction T$ to $Y \restriction T$. It would be convenient to denote that function R^T as well. Thus the function R^T is defined if and only if the relation R^T is functional.

Lemma 1. *Suppose that X and Y are T-similar. Then the function R^T from $X \restriction T$ to $Y \restriction T$ is defined and bijective.*

Proof. First we show that relation R^T is functional, and so function R^T is defined. Let $x = \mathsf{Val}(s, X)$. By the definition of R^T, we have $x \, R^T \, \mathsf{Val}(s, Y)$.

If $x\, R^T\, \mathsf{Val}(t,Y)$ as well, then, by the definition of R^T, we have $\mathsf{Val}(s,X) = x = \mathsf{Val}(t,X)$. Then $\mathsf{Val}_Y(s) = \mathsf{Val}_Y(t)$ because X and Y are T-similar.

By symmetry, the inverse of relation R^T is functional as well. It follows that function R^T is bijective. □

Definition 4. A bijection F from $X \restriction T$ to $Y \restriction T$ is a T*-isomorphism* from X to Y if for every term $f(t_1,\dots,t_j)$ in T

$$F(f_X(\mathsf{Val}_X(t_1),\dots,\mathsf{Val}_X(t_j))) = f_Y(F(\mathsf{Val}_X(t_1)),\dots,F(\mathsf{Val}_X(t_j))).$$

Lemma 2. *Suppose that X and Y are T-similar. Then function R^T is defined, is bijective, and is a T-isomorphism from X to Y.*

Proof. By the previous lemma, R^T is defined and bijective. Let t be a term $f(t_1,\dots,t_j)$ in T. If $x_i = \mathsf{Val}_X(t_i)$ for $i = 1,\dots,j$ then

$$\begin{aligned} R^T(f_X(\mathsf{Val}_X(t_1),\dots,\mathsf{Val}_X(t_j))) = R^T(\mathsf{Val}(t,X)) = \\ = \mathsf{Val}(t,Y) = f_Y(R^T(x_1),\dots,R^T(x_j)) \end{aligned}$$

□

Remark 2. Suppose that X and Y are T-similar. One may be tempted to say that the T-isomorphism R^T from X to Y is a partial isomorphism from X to Y which means that

$$R^T(f_X(x_1,\dots,x_j)) = f_Y(R^T(x_1),\dots,R^T(x_j))$$

whenever every x_i and $f_X(x_1,\dots,x_j)$ are in the domain of R^T. But this is not necessarily true. For example, let the nonlogical part of Υ consist of two 0-ary functional symbols α, β and a unary functional symbol f. Set $T = \{\alpha,\beta\}$ and consider states X and Y with three nonlogical elements a, b, c such that $f_X(a) = f_Y(a) = b$, $f_X(b) = f_Y(b) = c$, $f_X(c) = f_Y(c) = a$, and

$$\begin{aligned} \alpha_X = a, \quad \beta_X = b, \\ \alpha_Y = a, \quad \beta_Y = c. \end{aligned}$$

The states X and Y are T-similar: in both cases the values of α,β are distinct. But R^T is not a partial isomorphism. We have $R^T(f_X(a)) = R^T(b) = R^T(\beta_X) = \beta_Y = c$ while $f_Y(R^T(a)) = f_Y(R^T(\alpha_X)) = f_Y(\alpha_Y) = b$. □

Lemma 3. *If function R^T from $X \restriction T$ to $Y \restriction T$ is defined and bijective then X and Y are T-similar.*

Proof. Let t_1, t_2 be terms in T. We need to prove that $t_1 \sim_X t_2$ if and only if $t_1 \sim_Y t_2$. By symmetry it suffices to prove the "only if" direction.

Suppose $t_1 \sim_X t_2$. Then $\mathsf{Val}_Y(t_1) = R^T(\mathsf{Val}_X(t_1) = R^T(\mathsf{Val}_X(t_2)) = \mathsf{Val}_Y(t_2)$. Therefore $t_1 \sim_Y t_2$. □

Definition 5. An element a of state X is *T-accessible* if $\mathsf{Val}_X(t) = a$ for some $t \in T$. An update $(f, (a_1, \dots, a_j), a_0)$ is *T-accessible* if all a_i are T-accessible. A set of updates is *T-accessible* if every update in the set is T-accessible.

1.3 The new bounded exploration postulate

Postulate 4 (New Bounded Exploration Postulate)**.** There exists a finite set T of terms in the vocabulary of A, closed under subterms, such that

1. for every state X of A, $\Delta(A, X)$ is T-accessible, and
2. if states X and Y of A are T-similar, $f(t_1, \dots, t_j) \in T$, $a_i = \mathsf{Val}_X(t_i)$ and $b_i = \mathsf{Val}_Y(t_i)$ then

$$(f, (a_1, \dots, a_n), a_0) \in \Delta(A, X) \iff (f, (b_1, \dots, b_n), b_0) \in \Delta(A, Y). \quad \square$$

The original bounded exploration postulate did not require the accessibility of updates. The accessibility was derived [1].

Example 1. We give an example of a transition system A that satisfies the sequential time and abstract state postulates as well as the second part of the new bounded exploration postulate but where the updates are not accessible. The vocabulary Υ of A contains a nonlogical nullary function symbol f and no other nonlogical function symbols. Every state X of A consists of five distinct elements: three logical and two nonlogical elements; further, f_X is a nonlogical element. Every transition of A changes the value of f_X. Thus, if $\tau_A(X) = Y$, and the nonlogical elements of X are a and b, and $f_X = a$, then $f_Y = b$.

Clearly, A satisfies the sequential time and abstract state postulates. To check the second part of the new bounded exploration postulate, we can assume without loss of generality that T is the set $\{\mathsf{true}, \mathsf{false}, \mathsf{undef}, f\}$ of all Υ-term. Since the values of T-terms are distinct in every state of A, any two states are T-similar. Let X_1 and X_2 be any two states, with the nonlogical elements a_1, b_1, and a_2, b_2 respectively. Let $a_i = f_{X_i}$. We have $\Delta(A, X_i) = \{(f, b_i)\}$. Thus no $(f, \mathsf{Val}_{X_i}(t))$ belongs to $\Delta(A, X_i)$, and so

$$(f, \mathsf{Val}_{X_1}(t)) \in \Delta(A, X_1) \iff (f, \mathsf{Val}_{X_2}(t)) \in \Delta(A, X_2)$$

for every t. However, A fails the first part of the new bound exploration postulate as b_i is not T-accessible in X_i, and so $\Delta(A, X_i)$ is not T-accessible.

A does not satisfy the original bounded exploration postulate either. Indeed, let X be a state of A with nonlogical elements a, b where $f_X = a$, and let Y is obtained from X by replacing b with a fresh element c. Then X and Y coincide over T but

$$\Delta(A, X) = \{(f, b)\} \neq \{(f, c)\} = \Delta(A, Y) \quad \square$$

1.4 Theorems

We abbreviate "bounded exploration" to BE.

Theorem 1. *Let an object A satisfy the sequential time and abstract state postulates. Then A satisfies the new BE postulate if and only if it satisfies the original.*

Proof.

Only if We assume that A satisfies the new BE postulate with some BE witness T and we prove that it satisfies the original one with the same BE witness T. Suppose that the states X and Y of A coincide over T. We need to prove that $\Delta(A, X) = \Delta(A, Y)$.

Obviously X and Y are T-similar and R^T is the identity function from $X \restriction T$ onto $Y \restriction T$. By the new BE postulate, $\Delta(A, X) = \Delta(A, Y)$.

If We assume that A satisfies the original BE postulate with a BE witness T. Without loss of generality, T is closed under subterms. We prove that A satisfies the new postulate with that same bounded-exploration witness T. Thus we need to establish the following two claims:

1. for every state X of A, $\Delta(A, X)$ is T-accessible, and

2. if states X and Y of A are T-similar, $f(t_1, \dots, t_j) \in T$, $a_i = \mathsf{Val}_X(t_i)$ and $b_i = \mathsf{Val}_Y(t_i)$ then

$$(f, (a_1, \dots, a_j), a_0) \in \Delta(A, X) \iff (f, (b_1, \dots, b_j), b_0) \in \Delta(A, Y).$$

The first claim is proven in [1, Lemma 6.2]. To prove the second claim, suppose that X and Y are T-similar states of A, $f(t_1, \ldots, t_j) \in T$, $a_i = \mathsf{Val}_X(t_i)$, and $b_i = \mathsf{Val}_Y(t_i)$. By symmetry, it suffices to prove that $(f, (b_1, \ldots, b_j), b_0) \in \Delta(A, Y)$ if $(f, (a_1, \ldots, a_j), a_0) \in \Delta(A, X)$. Suppose that $(f, (a_1, \ldots, a_j), a_0) \in \Delta(A, X)$.

Case 1: $X \cap Y = \emptyset$. Consider a new state X' obtained from X by replacing $\mathsf{Val}_X(t)$ by $\mathsf{Val}_Y(t)$ for every $t \in T$. States X and X' are isomorphic. The desired isomorphism ξ coincides with R^T on $X \restriction T$ and is identity otherwise. Isomorphism ξ naturally lifts to locations, updates and sets of updates [1]. Accordingly

$$(f, (b_1, \ldots, b_j), b_0) = \xi((f, (a_1, \ldots, a_j), a_0)) \in \xi(\Delta(A, X)) = \Delta(A, X').$$

Now check, by induction on the depth of term t, that $\mathsf{Val}(t, X') = \mathsf{Val}(t, Y)$ for all $t \in T$. Therefore X' and Y coincide over T. By the old BE postulate, $\Delta(A, X') = \Delta(A, Y)$ and so $(f, (b_1, \ldots, b_j), b_0) \in \Delta(A, Y)$.

Case 2: $X \cap Y \neq \emptyset$. Let η be an isomorphism from X to a state X' of A such that $X' \cap Y = \emptyset$. Lifting η as above, we have $\mathsf{Val}_{X'}(t_i) = \eta a_i$ and

$$(f, (\eta a_1, \ldots, \eta a_j), \eta a_0) = \eta((f, (a_1, \ldots, a_j), a_0)) \in \eta(\Delta(A, X)) = \Delta(A, X')$$

It is clear that X' and Y are T-similar. We have Case 1 with X' playing the role of X and ηa_i playing the role of a_i. Thus $(f, (b_1, \ldots, b_j), b_0) \in \Delta(A, Y)$. □

We recall Lemma 6.11 (Main Lemma) in [1].

Main Lemma *For every sequential algorithm A of vocabulary Υ, there is an ASM program Π of vocabulary Υ such that $\Delta(A, X) = \Delta(A, \Pi)$ for all states X of A.*

We reprove Main Lemma using the new definition of sequential algorithms where the original BE postulate is replaced with the new BE postulate. The new proof is simpler than the original.

Proof. Consider an arbitrary state X of A, and let T be a bounded-exploration witness for A. By the new BE postulate, $\Delta(A, X)$ is accessible. For every update $u = (f, (a_1, \ldots, a_j), a_0) \in \Delta(A, X)$, fix terms t_i^u such that $\mathsf{Val}(t_i^u, X) = a_i$ and construct an update rule $f(t_1^u, \ldots, t_j^u) := t_0^u$ which we will call R^u. Let R_X be the do-in-parallel composition of the rules R^u.

For every state Y of A, T-similar to X, we have

$$\Delta(R_X, Y) = \Delta(A, Y)$$

Indeed, the similarity function R^T is a T-isomorphism from X to Y. Therefore

$$\Delta(R_X, Y) = R^T(\Delta(R_X, X)) = R^T(\Delta(A, X)) = \Delta(A, Y).$$

The first equality follows from the definition of R^T, the second one holds by the construction of R_X, the last one is a corollary of the new BE postulate.

Fix a maximal collection of states X_1,...,X_m of A where no two states are T-similar. Let φ_i be the term

$$\begin{aligned} & \bigwedge\{s = t \ : \ s, t \in T \ \wedge \ \mathsf{Val}(s, X_i) = \mathsf{Val}(t, X_i)\} \\ \wedge \quad & \bigwedge\{s \neq t \ : \ s, t \in T \ \wedge \ \mathsf{Val}(s, X_i) \neq \mathsf{Val}(t, X_i)\}. \end{aligned}$$

The desired Π is the following program:

```
do in-parallel
    if   $\varphi_1$   then   $R_{X_1}$
    if   $\varphi_2$   then   $R_{X_2}$
    ...
    if   $\varphi_m$   then   $R_{X_m}$
```

Now let Y be an arbitrary state of A. By the choice of states X_1,...,X_m, state Y is T-similar to some X_i, so that φ_i holds in Y and every other φ_j fails in Y; hence $\Delta(A, Y) = \Delta(R_{X_i}, Y)$. By the preceding equation, $\Delta(R_{X_i}, Y) = \Delta(A, Y)$. □

Chapter 2

Bounded-Choice Sequential Algorithms

Tatiana Yavorskaya

Abstract

In his article "Sequential Abstract State Machines Capture Sequential Algorithms", Yuri Gurevich characterized deterministic sequential algorithms as well as bounded-choice sequential algorithms (subsection 9.2) but the second characterization is incomplete in that the proofs are missing. Here we give a complete form of the characterization of bounded-choice sequential algorithms.

2.1 Introduction

In [1], Gurevich characterized deterministic sequential algorithms. More exactly he axiomatized deterministic sequential algorithms by means of three postulates and proved that every such algorithm is behaviorally equivalent to a deterministic sequential abstract state machine (and of course that every deterministic sequential abstract state machine satisfies the three postulates). In § 9.2, he sketched how to characterize bounded-choice sequential algorithms. Here we turn the sketch into a complete characterization. As usual, abstract state machines are called ASMs.

In Section 2.2, we modify the three postulate and thus axiomatize bounded-choice sequential algorithms. Then we define bounded-choice sequential ASMs, and we check that every such ASM satisfies the modified postulates.

In Section 2.3, we prove that every bounded-choice sequential algorithm is behaviorally equivalent to an appropriate bounded-choice sequential ASM.

Remark 1. In the case of bounded-choice sequential algorithms, the bounded-choice postulate plays the role that is played by the bounded-exploration postulate in [1]. The bounded-exploration postulate was reformed into a more abstract form in Chapter I. The bounded-choice postulate can be reformed in the same way though we do not do that here.

2.2 Definitions

The sequential time postulate of [1] needs to be relaxed.

Postulate 5 (Nondeterministic sequential time)**.** Every nondeterministic sequential algorithm A is associated with three objects:

- a nonempty set $\mathcal{S}(A)$, the set of states of A,
- a nonempty subset $\mathcal{I}(A)$ of $\mathcal{S}(A)$, the set of the initial states of A, and
- a relation $\tau_A \subseteq \mathcal{S}(A) \times \mathcal{S}(A)$, the *one-step transition relation* of A.

Remark 2. A is bounded-choice if, for every state X, there are finitely many states which are related to X by the transition relation τ_A. We do not formulate this restriction in the sequential time postulate, because we choose to separate concerns and to collect all the bounds in the Bounded Choice Postulate. □

The abstract state postulate of [1] needs to be adjusted to reflect the change in the sequential time postulate.

Postulate 6 (Nondeterministic abstract state)**.**

- States of A are first order structures; all states of A have the same vocabulary.
- If two states are τ_A-related then they have the same base sets, that is, if $(X, Y) \in \tau_A$ then $\mathsf{Dom}(X) = \mathsf{Dom}(Y)$.
- The set $\mathcal{S}(A)$ is closed under isomorphism. If α is an isomorphism from the state X onto X' then for every Y with $(X, Y) \in \tau_A$ there exists Y' such that $(X', Y') \in \tau_A$ and α is an isomorphism from Y onto Y'.

We use the definition of a location and an update from [1]. For the sake of readability we omit parentheses in the notation for locations and updates and use simplified notation. Thus $(f, a_1, \ldots, a_n)$ is a location such that f is a functional symbol of arity n and all a_i are elements of the state. $(f, a_1, \ldots, a_n; a_0)$ is the update of that location with an element a_0.

As in [1], for two arbitrary states X and Y by $Y - X$ we denote the set of updates which, being applied to X, gives Y, so that $Y = X + (Y - X)$ in the notation of [1]. Note that $Y - X$ is uniquely defined for all states X and Y with the same base sets. If states Y_1 and Y_2 are different then $Y_1 - X$ and $Y_2 - X$ are different as well.

Definition 1. For a nondeterministic algorithm A we define for every state X the *update family* $\mathcal{F}(A; X) = \{Y - X \mid (X, Y) \in \tau_A\}$. Note that $\mathcal{F}(A; X)$ is a set consisting of sets of updates. □

Postulate 7 (Bounded Choice)**.** There exists a finite set of terms T such that for every two states X and Y if $\mathsf{Val}(t, X) = \mathsf{Val}(t, Y)$ for every term $t \in T$ then $\mathcal{F}(A; X) = \mathcal{F}(A; Y)$.

Definition 2. Programs (or rules) for bounded-choice ASM's are defined by induction.

- Update rules
  $\mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_\mathsf{n}) := \mathsf{t}_0$
  Here f is a functional symbol of arity n and all t_i are terms.

- Conditional rules
  $\mathsf{if}\ b\ \mathsf{then}\ R1\ \mathsf{else}\ R2$
  Here b is a boolean term and $R1$, $R2$ are rules.

- Parallel rules
  $\mathsf{do\text{-}in\text{-}parallel}\quad R_1, \ldots, R_n$
  Here all R_i are rules.

- Choose-among rules
  $\mathsf{choose\text{-}among}\quad R_1, \ldots, R_n$
  Here all R_i are rules. □

Definition 3. With any program π and state X we associate an update family $\mathcal{F}(\pi; X)$ in the following way.

- If R is an update rule $f(t_1, \ldots, t_n) := t_0$ then

$$\mathcal{F}(R; X) = \{\{(f, \mathsf{Val}(t_1), \ldots, \mathsf{Val}(t_n); \mathsf{Val}(t_0))\}\}.$$

- If R is a conditional rule $\mathsf{if}\ b\ \mathsf{then}\ R1\ \mathsf{else}\ R2$ then

$$\mathcal{F}(R; X) = \begin{cases} \mathcal{F}(R_1; X) & \text{if } \mathsf{Val}(b) = \mathsf{true}; \\ \mathcal{F}(R_2; X) & \text{if } \mathsf{Val}(b) = \mathsf{false}. \end{cases}$$

- If R is a parallel rule $\mathsf{do\text{-}in\text{-}parallel}\ R_1, \ldots, R_n$ then

$$\mathcal{F}(R; X) = \{\cup_{i=1}^n \Delta_i \mid \Delta_i \in \mathcal{F}(R_i; X) \text{ and } \cup_{i=1}^n \Delta_i \text{ does not clash}\}.$$

- If R is a rule $\mathsf{choose\ among\ rules}\ R_1, \ldots, R_n$ then

$$\mathcal{F}(R; X) = \bigcup_i \mathcal{F}(R_i; X).$$

Bounded-choice ASMs are defined exactly as sequential ASMs are defined in [1], except that programs are as above rather than as in [1].

Theorem 1. *Every bounded-choice ASM is a bounded-choice algorithm.*

Proof. Obvious. □

2.3 Emulation of algorithms by ASM's

Theorem 2. *For every bounded-choice algorithm A there exists a bounded choice ASM with the same states and initial states and with a program π such that $\mathcal{F}(A;X) = \mathcal{F}(\pi;X)$ for every state X.*

Proof. Suppose that $A = (\mathcal{S}(A), \mathcal{I}(A), \tau)$ is an algorithm in the vocabulary Υ and the set of terms T is a bound-exploration witness for A. Without loss of generality we assume that T is closed under subterms and contains true, false, undef.

The following terminology is taken from [1]. We call $a \in X$ *a critical element* if there exists a term $t \in T$ such that $\mathsf{Val}(t, X) = a$. An update $(f, a_1, \ldots, a_n; a_0)$ is critical if all a_i are critical.

Lemma 1. *For $(X, Y) \in \tau_A$, all updates from $Y - X$ are critical.*

Proof. Suppose that $(f, a_1, \ldots, a_n; a_0) \in Y - X$ and a_i is not critical for some i. We consider a structure X' obtained from X by replacing a_i by a fresh element a_i'. Since X' is isomorphic to X it is also a state of our algorithm. By the induction on the construction of a term t we can show that $\mathsf{Val}(t, X) = \mathsf{Val}(t, X')$ for every term $t \in T$. Then according to the bounded choice postulate we conclude that $\mathcal{F}(A, X) = \mathcal{F}(A, X')$. Since $a_i \notin X'$ we obtain then a_i cannot occur in any element from $\mathcal{F}(A, X')$, whence it cannot occur in any element of $\mathcal{F}(A, X)$, in particularly in $Y - X$, contradiction. □

Corollary 1. *For every X, the set $\mathcal{F}(A, X)$ is finite, and its elements are finite sets. Furthermore, the set $\tau_A(X) = \{Y \mid (X, Y) \in \tau_A\}$ is finite.*

Proof. Indeed, $\mathcal{F}(A, X) = \{Y - X \mid (X, Y) \in \tau\}$. By the previous lemma all updates from $Y - X$ are critical. Since both the set T and the vocabulary of A are finite, the set of all critical updates is finite. Thus, for every Y the set $Y - X$ is finite. Then $\mathcal{F}(A, X)$ is finite since the number of all sets of critical updates is also finite. Since different $Y \in \tau_A(X)$ yield different $Y - X$, the set $\tau_A(X)$ is also finite. □

We define the desired program π in three steps.

Step 1. Suppose that $(X, Y) \in \tau_A$. By corollary 1 $Y - X$ is a finite set. Suppose that the list $(f^j, a_1^j, \ldots, a_{n^j}^j; a_0^j)$ for $j = 1, \ldots, k$ comprises all updates from $Y - X$. By lemma 1, $a_i^j = \mathsf{Val}(t_i^j, X)$ where t_i^j are terms from T.

We define the rule $R^{X,Y}$ as the do-in-parallel composition of the update rules $t_0^j := f^j(t_1^j, \ldots, t_{n^j}^j)$ for $j = 1, \ldots, k$. It is clear that

$$\mathcal{F}(R^{X,Y}; X) = \{Y - X\}. \tag{2.1}$$

Step 2. By corollary 1 the set $\tau_A(X)$ is finite. Suppose that it consists of the states Y_1,...,Y_m. We define R^X as follows

$$\textsf{choose among rules} \quad R^{X,Y_1}, \ \ldots \ , \ R^{X,Y_m}$$

In view of 2.1 it is clear that

$$\mathcal{F}(R^X; X) = \mathcal{F}(A; X). \tag{2.2}$$

Step 3. Now we can define the whole program π. For every state $X \in \mathcal{S}(A)$ we define a boolean term φ^X as follows:

$$\begin{aligned} \varphi^X \ \rightleftharpoons \ & \bigwedge\{t_1 = t_2 \mid t_1, t_2 \in T, \ \mathsf{Val}(t_1, X) = \mathsf{Val}(t_2, X)\} \wedge \\ & \bigwedge\{t_1 \neq t_2 \mid t_1, t_2 \in T, \ \mathsf{Val}(t_1, X) \neq \mathsf{Val}(t_2, X)\}. \end{aligned} \tag{2.3}$$

Since T is finite, there are finitely many different φ^X. It is also clear that φ^X is true in X, that is, $\mathsf{Val}(\varphi^X, X) = \mathsf{true}$.

Let $X_1, \ldots, X_n$ be the the maximal list of states of A for which formulas φ^X are different. We put π to be the following program

```
do in parallel
    if $\varphi^{X_1}$ then $R^{X_1}$
        ...
    if $\varphi^{X_n}$ then $R^{X_n}$
```

We have to prove that $\mathcal{F}(\pi; Y) = \mathcal{F}(A; Y)$ for every state Y. Since there is a single formula φ^{X_i} which is true in Y, it remains to show that from $\mathsf{Val}(\varphi^X, Y) = \mathsf{true}$ it follows that $\mathcal{F}(R^X; Y) = \mathcal{F}(A; Y)$

The proof consists in a series of lemmas. We say that the states X and Y coincide over T and write $X =_T Y$ if for every term $t \in T$ one has $\mathsf{Val}(t, X) = \mathsf{Val}(t, Y)$.

Lemma 2. *If $X =_T Y$ then $\mathcal{F}(R^X; Y) = \mathcal{F}(A; Y)$.*

Proof. The following equalities hold:

$$\mathcal{F}(R^X;Y) = \mathcal{F}(R^X;X) = \mathcal{F}(A;X) = \mathcal{F}(A;Y).$$

The first equality follows from $X =_T Y$ since R^X uses only terms from T. The second one follows from (2.2). The last one is a corollary of the bounded choice postulate. □

Lemma 3. *Suppose that $\mathcal{F}(R^X;Y) = \mathcal{F}(A;Y)$ and the states Y and Z are isomorphic. Then $\mathcal{F}(R^X;Z) = \mathcal{F}(A;Z)$.*

Proof. Let $i : Y \mapsto Z$ be an isomorphism. We extend i to updates and sets of updates. Then

$$\mathcal{F}(R^X;Z) = i(\mathcal{F}(R^X;Y)) = i(\mathcal{F}(A;Y)) = \mathcal{F}(A;Z)$$

where the first equality can be checked directly, the second one follows from the conditions of the current lemma and the third one is a corollary of the last part of the abstract state postulate. □

Lemma 4. *Let X and Y be two states and $\mathsf{Val}(\varphi^X, Y) = \mathsf{true}$. Then $\mathcal{F}(R^X;Y) = \mathcal{F}(A;Y)$.*

Proof. **Case 1**. If $\mathsf{Dom}(X) \cap \mathsf{Dom}(Y) = \emptyset$, we consider the state $\tilde{Y}$ obtained from Y by replacing $\mathsf{Val}(t,Y)$ by $\mathsf{Val}(t,X)$ for all terms $t \in T$. Since $\tilde{Y} \simeq Y$ we have $\tilde{Y} \in \mathcal{S}(A)$ by the abstract state postulate. Since $\tilde{Y} =_T X$, then φ^X is true in $\tilde{Y}$ whence by lemma 2 we conclude $\mathcal{F}(R^X;\tilde{Y}) = \mathcal{F}(A;\tilde{Y})$. Since $\widetilde{Y} \simeq Y$, by lemma 3 we obtain $\mathcal{F}(R^X;Y) = \mathcal{F}(A;Y)$.

Case 2. Now suppose that $\mathsf{Dom}(X) \cap \mathsf{Dom}(Y) \neq \emptyset$. Construct an isomorphic copy of Y (denoted by Y') such that $\mathsf{Dom}(X) \cap \mathsf{Dom}(Y') = \emptyset$. Then the condition φ^X is true in Y' since it is true in Y. Therefore $\mathcal{F}(R^X,Y') = \mathcal{F}(A,Y')$ by case 1. Since $Y' \simeq Y$ we conclude $\mathcal{F}(R^X,Y) = \mathcal{F}(A,Y)$ by lemma 3. □

□

Bibliography

[1] Yuri Gurevich. *Sequential Abstract State Machines Capture Sequential Algorithms.* ACM Trans. on Computational Logic 1:1 (2000), 77–111.

[2] Andreas Blass and Yuri Gurevich. *Ordinary Small-Step Algorithms, I.* ACM Trans. on Computational Logic 7:2 (2006).