Verified Isabelle/HOL tactic for the theory of bounded linear integer arithmetic based on quantifier instantiation and SMT

Rafael Sadykov
Faculty of Mathematics and Mechanics
Lomonosov Moscow State University
Moscow, Russia
sadykovrf@gmail.com

Mikhail Mandrykin
Software Engineering Department,
Ivannikov Institute for System Programming of the RAS
Moscow, Russia
mandrykin@ispras.ru

***Abstract*—SMT solvers are widely applied for deductive verification of C programs using various verification platforms (Why3, Frama-C/WP, F*) and interactive theorem proving systems (Isabelle, HOL4, Coq) as the decision procedures implemented in SMT solvers are complete for some combinations of logical theories (logics), in particular for the QF_UFLIA logic. At the same time, when verifying C programs, it is often necessary to discharge formulas in other logical theories and their combinations, that are also decidable but not supported by all SMT solvers. Theories of bounded integers both with overflow (for unsigned integers in C) and without overflow (for signed integers), and also theory of finite interpreted sets (needed to support frame conditions) are good examples of such theories. One of the possible ways to support such theories is to directly implement them in SMT-solvers, however, this method is often time-consuming, as well as insufficiently flexible and universal. Another way is to implement custom quantifier instantiation strategies to reduce formulas in unsupported theories to formulas in widespread decidable logics such as QF_UFLIA. In this paper, we present an instantiation procedure for translating formulas in the theory of bounded integers without overflow into the QF_UFLIA logic. We formally proved soundness and completeness of our instantiation procedure in Isabelle. The paper presents an informal description of this proof as well as some considerations on the efficiency of the proposed procedure. Our approach is sufficient to obtain efficient decision procedures implemented as Isabelle/HOL proof methods for several decidable logical theories used in C program verification by relying on the existing capabilities of well-known SMT solvers, such as Z3 and proof reconstruction capabilities of the Isabelle/HOL proof assistant.**

***Index Terms*—static verification; quantifier instantiation; SMT formulas; SMT solvers; automated decision procedures; software verification.**

I. Introduction

In the process of developing software in the C language, errors often arise due to improper use of arithmetic operations, which can lead to overflow and result in unexpected behaviour, despite excessive testing. However, with the use of formal verification, it is possible to find all potential errors related to operation on machine integer values. Many deductive verification methods and tools make use of automated solvers for the problem of logical formula satisfiability modulo theories, also called SMT solvers. Algorithms implemented by such solvers are complete for some combinations of logical theories, in particular for the QF_UFLIA logic. Although SMT solvers are applicable to a wide class of problems, it is often necessary to apply decision procedures to formulas in logical theories that are decidable but not always fully supported by SMT solvers. Such logical theories include bounded integers both with and without overflow (useful for unsigned and signed integers in C, correspondingly), the theory of finite interpreted sets (necessary to supporting frame conditions), etc. In this paper we propose a method for validating formulas in the theory of bounded integers based on the use of an SMT solver for QF_UFLIA logic, intended to automatically obtain (reconstruct) proofs within the Isabelle/HOL system for interactive theorem proving [1]. Our method is a complete decision procedure for formulas in the theory of bounded integers, implemented as an Isabelle/HOL method. Its operation consists of the following steps: extending the original formula with instances of the axioms from the theory of bounded integers (the axioms themselves being lemmas proved in the Isabelle HOL-Word theory [2]); interpreting the resulting formula in the QF_UFLIA logic; applying an existing proof-producing SMT solver and subsequently reconstructing the proof in Isabelle/HOL using the tools already employed by its `smt` proof method. The completeness proof of the proposed method is based on transforming the model of the extended formula in QF_UFLIA logic (that can be obtained from the SMT solver) into a model in the theory of bounded integers.

Extending the formula with instantiations of the axioms only linearly increases its size, and as the formulas typically involved in interactive verification are not very large, the resulting overhead is practically negligible. Also, there is no need to alter the Isabelle proof reconstruction tools or the SMT solver. The main goal of our work is to prove the completeness and soundness of the proposed procedure for converting formulas in the theory of bounded integers to the QF_UFLIA logic.

Identify applicable funding agency here. If none, delete this.

A. Related work

We also reviewed several papers on modular integer arithmetic [3], [4] and [5] describing integer overflow errors and verification of programs utilizing bounded integers. "Modular Arithmetic Decision Procedure" [3] proposes transformation methods for formulas in non-linear modular arithmetic with bounded integers, which involve altering the internal algorithms of the SMT solver. The proofs potentially produced by the proposed methods (the paper does not provide an exhaustive description of the corresponding deductive system) could be difficult to reproduce using the existing capabilities of the interactive theorem proving systems. The paper "Modular difference logic is hard" [4] estimates the complexity of the decision procedure for modular IDL (integer difference logic) for a fixed modulus. By definition, IDL formulas are conjunctions of inequalities of the form $x - y < b$, where x and y are integer variables and b is an integer constant. The decision procedure for this logic is polynomial, however the decision procedure for the corresponding modular logic (for a fixed modulus) is NP-complete. The logic of linear integer inequalities (LIA), which extends the IDL by adding the operation of multiplication by a constant, is also an NP-complete problem.

Thus by itself, a decision procedure for bounded linear integer arithmetic is not at all a novel contribution. There are existing decision procedures for more general theories such as modular arithmetic with non-linear multiplication. However, a decision procedure we propose in this paper is suitable for simple implementation within existing proof reconstruction framework of Isabelle/HOL [6], [7] as it does not require any changes in either in the implementation of the SMT solver or its proof format. It is also easily extensible in order to support new function symbols with axiomatically defined semantics. As the theory of bounded linear arithmetic is far from the only decidable logical theory fragment that has practical relevance in program verification yet is not directly supported by existing SMT solvers, we argue that our example demonstrates an interesting approach to extending capabilities of existing interactive theorem provers such as Isabelle/HOL with little programming effort.

II. Preliminaries

In the SMT-LIB standard [8], QF_UFLIA logic is defined for quantifier-free formulas with equality and uninterpreted constant and function symbols interpreted in the combined theories of linear integer arithmetic and uninterpreted functions. The QF_UFLIA logic can be considered a combination of the QF_LIA and QF_UF logics. QF_LIA includes quantifier-free formulas with equality in the theory of linear integer arithmetic (LIA). Its signature includes the following function symbols: $\{+, c\times\cdot, \leq\}$, where c is an integer constant. QF_UF includes quantifier-free formulas with equality in the theory of uninterpreted functions.

The theory of linear arithmetic with bounded integers (BLIA) will be defined axiomatically as an extension of the theory of linear integer arithmetic. In addition to the symbols of the LIA theory, the following functional symbols are included in its signature: $\{+_b, c\times_b\cdot, v(\cdot), (\cdot)_b, \leq_b\}$, where c is unbounded integer constant. The terms (if considered separately from predicates) in this theory can be of two sorts: bounded or unbounded integers. Bounded integers take integer values from a certain range $[L\dots U]$ (both ends included). These bounded integers can be obtained using the function $v(\cdot)$. The names a, b, and d are be used in the following to denote variables of the bounded integer type. For the unbounded integer type, we use names x, y, and c, where c always represents an interpreted integer constant. Thus, for any bounded integer a, the relation $L \leq v(a) \leq U$ holds. The semantics of the operations $+_b$ and $\times_b$ is such that their results coincide with the results of the corresponding integer operations applied to the values of bounded arguments if these results can be represented as bounded integers. Otherwise, these results are not defined, i.e, their model is not fixed and can be chosen arbitrarily for each specific formula. The operation $(\cdot)_b$ returns a bounded integer with a given value if this value lies in the range $[L\dots U]$ and an undefined result otherwise. The operation $\leq_b$ compares the values of bounded integers and is essentially a shorthand notation for the expression $v(\cdot) \leq v(\cdot)$. Consider the following examples of formulas in the theory of bounded integers:

$$v(a) \leq 5 \wedge 5 \leq v(a) \wedge (5)_b \times_b a +_b (1)_b \neq (26)_b \text{ (1) and}$$

$$a \leq_b (5)_b \wedge (5)_b \leq_b a \wedge (5 \times v(a) + 1 - 26)_b \neq (0)_b \text{ (2).}$$

We will assume $L = 0$ and $U = 25$. Under these assumptions, formula (1) will be satisfiable, for example, if we assume the bounded variable a to be such that $v(a) = 5$. Then $v(a) = 5$, $(5)_b \times_b a = (25)_b$, but $(5)_b \times_b a +_b (1)_b$ and $(26)_b$ are undefined, since $25 + 1 = 26 > 25 = U$ and can be chosen equal to, say, $(0)_b$ and $(1)_b$ correspondingly. Formula (2) is unsatisfiable, because $a \leq_b (5)_b \wedge (5)_b \leq_b a$ implies that $v(a) = 5$, which means $5 \times v(a) + 1 - 26 = 0$ and $(0)_b = (0)_b$. To solve the problem of formula satisfiability in BLIA, we use translation into the QF_UFLIA logic. The translation procedure for a formula involves instantiation of the axioms defining the BLIA theory within LIA. Instantiation of an axiom for a given formula is defined as a conjunction of a ground substitution of the axiom with the formula. Instantiations are triggered by occurrences of subterms matching certain patterns. Only subterms of the original formula are considered. The described transformation yields a formula F^*, which is then interpreted in the QF_UFLIA logic. Next, the SMT solver's decision procedure is called, producing a proof, which is then reproduced (certified) by Isabelle. To prove the completeness of the transformation procedure, consider the case when a model R of the transformed formula F^* is obtained in the QF_UFLIA logic. In the paper we call the model R the *realization* of formula F. In this paper, we demonstrate how the model M of the original formula F in BLIA can be reconstructed from its realization R. The following sections explain the basic definitions, describe the instantiation procedure and provide an example of its

operation. Then we present the statement and proof of the corresponding completeness theorem.

III. Problem

Let $\mathbb{Z}$ be the set of integers, $\mathbb{Z}_b$ be the set of bounded integers and $\Sigma = \{+_b, \times_b, \leq_b\}$ be the signature of the theory T of linear arithmetic with bounded integers. F is a formula in BLIA, where bounded integer variables range over the set $\mathbb{Z}_v$. We define the BLIA theory within the theory of linear integer arithmetic (LIA) axiomatically using the set of 5 axioms (See Fig. 1). We denote this set as T_0. In the actual implementation of our proof method in Isabelle/HOL, we prove those axioms as lemmas of the `HOL-Word` theory [2]. Since the axiom $A6$ in the definition of the theory T_0 is quantified over two variables a and b, we introduce another re-formulated theory T_1, which differs from T_0 only in the axiom $A6$ (see axiom $A6'$ in Fig. 1). This is done to avoid quadratic overhead in the size of the original formula during instantiation. Therefore, our procedure instantiates axioms of the theory T_1.

Let's show the equivalence of our two axiomatizations T_0 and T_1. We state and prove this equivalence in the following theorem:

Theorem 1. *Every ground (quantifier-free) formula F in the signature Σ is satisfiable in the theory T_0 if and only if it is satisfiable in the theory T_1.*

Proof. We show that whenever a formula F has a model M in theory T_0 it has the same model M in theory T_1 and vice versa.

First assume F has a model M in T_0. Then all the axioms except for $(A6')$ hold in M. Take any arbitrary bounded integer a. Since $(A4)$ holds, we have

$$L \leq v^M(a) \leq U.$$

Let's denote $v^M(a)$ as c. Then $L \leq c = v^M(a) \leq U$ and since $(A5)$ holds, we have

$$v^M\big((c)_b^M\big) = c.$$

Now we have $v^M\big((c)_b^M)\big) = c = v^M(a)$ and by axiom $(A6)$ also have

$$(c)_b^M = a.$$

Now from $v^M(a) = c$ by congruence we have

$$v^M\Big(\big(v^M(a)\big)_b^M\Big) = v^M\big((c)_b^M\big),$$

and again by $(A6)$,

$$\big(v^M(a)\big)_b^M = (c)_b^M.$$

Hence $\big(v^M(a)\big)_b^M = (c)_b^M = a$ and we have $(A6')$ for arbitrary a.

Now assume F has a model M in T_1. Then all the axioms except for $(A6)$ hold in M. Therefore for any arbitrary bounded integers a and b such that $v^M(a) = v^M(b)$ by axiom $(A6')$ we have

$$\big(v^M(a)\big)_b^M = a \text{ and } \big(v^M(b)\big)_b^M = b.$$

From $v^M(a) = v^M(b)$ we have $\big(v^M(a)\big)_b^M = \big(v^M(b)\big)_b^M$ by congruence. Hence $a = b$ and we have $(A6)$ for any arbitrary a and b. □

IV. Instantiation procedure

Now assume we have a ground formula F in BLIA. To check its satisfiability we translate the formula into an equisatiafiable formula F^* in the QF_UFLIA logic. The translation procedure simply instantiates axioms of the definition T_0 according to the following triggers:

- $(A1)$ is instantiated with the terms a and b for every subterm of the form $a +_b b$ occurring in F;
- $(A2)$ is instantiated with the terms c and a for every subterm of the form $c \times_b a$ occurring in F;
- $(A3)$ is instantiated with the terms a and b for every subterm of the form $a \leq_b b$ occurring in F;
- $(A4)$ is instantiated with the term a for every bounded integer subterm occurring in F;
- $(A5)$ is instantiated with the term c for every subterm of the form $(c)_b$ occurring in F;
- $(A6')$ is instantiated with the term a for every bounded integer subterm occurring in F.

We denote the formula resulting from this instantiation procedure as F^*.

Let's illustrate our instantiation procedure on the following sample formula F assuming $L = -25$ and $U = 25$:

$$F = (25)_b \leq_b a \wedge -1 \times_b a +_b (25)_b \neq (0)_b.$$

Relevant instantiations according to our procedure are shown in Fig. 2. Using these instantiations we can show the formula F to be unsatisfiable.

Proposition 1. *The formula F is unsatisfiable.*

Proof. In the proof we use only the instantiations shown in Fig. 2. From the instantiation of $(A5)$ we have $v\big((25)_b\big) = 25$. Then from the instantiation of $(A3)$ we have $25 = v\big((25)_b\big) \leq v(a)$ and also from the instantiation of $(A4)$ have $v(a) \leq 25$. Thus $v(a) = 25$. Now from the instantiation of $(A2)$ we have $v(-1 \times_b a) = -1 \times v(a) = -25$, then from $(A1)$ have $v\big(-1 \times_b a +_b (25)_b\big) = v(-1 \times_b a) + v\big((25)_b\big) = -25 + 25 = 0$. Finally, from the instantiation of $(A6')$ we have $-1 \times_b a +_b (25)_b = \Big(v\big(-1 \times_b a +_b (25)_b\big)\Big)_b = (0)_b$. This is in contradiction with the second conjunct $-1 \times_b a +_b (25)_b \neq (0)_b$ of the formula F. Thus F is unsatisfiable. □

V. Soundness

The soundness of our procedure is straightforward since we only instantiate the axioms from the definition of the BLIA theory.

VI. Completeness

In this section we assume that we have an arbitrary ground (quantifier-free) formula F in BLIA, its translation F^* to QF_UFLIA obtained according to our instantiation procedure, and the model R of the translation F^* in QF_UFLIA that

$$\Sigma = \{+_b, \times_b, \leq_b\},\ a,\ b \in \mathbb{Z}_b,\ (c)_b \in \mathbb{Z}_b,\ v(a) \in \mathbb{Z}, c \in \mathbb{Z}\ \text{— constant}$$

$$\begin{aligned}
\forall a\, b \in \mathbb{Z}_b.\ \ & L \leq v(a) + v(b) \leq U \implies v(a +_b b) = v(a) + v(b), && (A1)\\
\forall a \in \mathbb{Z}_b. \forall c \in \mathbb{Z}.\ \ & L \leq c \times v(a) \leq U \implies v(c \times_b a) = c \times v(a), && (A2)\\
\forall a\, b \in \mathbb{Z}_b.\ \ & a \leq_b b \iff v(a) \leq v(b), && (A3)\\
\forall a \in \mathbb{Z}_b.\ \ & L \leq v(a) \leq U, && (A4)\\
\forall c \in \mathbb{Z}.\ \ & L \leq c \leq U \implies v\big((c)_b\big) = c, && (A5)\\
\forall a\, b \in \mathbb{Z}_b.\ \ & v(a) = v(b) \implies a = b. && (A6)
\end{aligned}$$

$$\forall a \in \mathbb{Z}_b.\quad \big(v(a)\big)_b = a. \qquad (A6')$$

Fig. 1. Axioms defining the theory of bounded integers

$$\begin{aligned}
\underline{-1 \times_b a +_b (25)_b} \in F \Rightarrow\ & -25 \leq v(-1 \times_b a) + v\big((25)_b\big) \leq 25 \implies v\big(-1 \times_b a +_b (25)_b\big) = v(-1 \times_b a) + v\big((25)_b\big), && (A1)\\
\underline{-1} \times_b \underline{a} \in F \Rightarrow\ & -25 \leq -1 \times v(a) \leq 25 \implies v(-1 \times_b a) = -1 \times v(a), && (A2)\\
\underline{(25)_b} \leq_b \underline{a} \in F \Rightarrow\ & (25)_b \leq_b a \iff v\big((25)_b\big) \leq v(a), && (A3)\\
a \in F \Rightarrow\ & -25 \leq v(a) \leq 25, && (A4)\\
(\underline{25})_b \in F \Rightarrow\ & -25 \leq 25 \leq 25 \implies v\big((25)_b\big) = 25, && (A5)\\
-1 \times_b a +_b (25)_b \in F \Rightarrow\ & \Big(v\big(-1 \times_b a +_b (25)_b\big)\Big)_b = -1 \times_b a +_b (25)_b. && (A6')
\end{aligned}$$

Fig. 2. Translation of the sample formula $F = (25)_b \leq_b a \wedge -1 \times_b a +_b (25)_b \neq (0)_b$

we call realization. We use plain terms and predicates (e.g. $v(u) + v(t)$) to denote the syntactic terms and predicates as they occur in the formulas F and F^* while their valuations in the realization R are denoted with the superscript R e.g. $v^R(u^R) +_b^R v^R(t^R)$. Free bounded integer terms are denoted with letters t and u, free integer terms are denoted with k or n, while arbitrarily chosen bounded integer values are debited with letters a and b, and unbounded integer values are denoted with x, y or c. The section proceeds by gradually reconstructing the model M of the original formula F in BLIA. First we prove several auxiliary lemmas.

Lemma 1. *A term of the form $v(t)$ occurs in F^* if and only if the bounded integer term t occurs in F.*

Proof. First assume a bounded integer term t occurs in F. Then it is a bounded integer subterm of F and according to the rule for instantiating the axiom $(A4)$, an instantiation of this axiom with t occurs in F^*. Since the instantiation contains the term $v(t)$, it occurs in F^*.

Now assume a term of the form $v(t)$ occurs in F^*. Let's show a term of this form can arise during the instantiation procedure only if the term t already occurs in F. We show this by induction on the instantiation rules. Consider the rule for instantiating $(A1)$. An instantiation of $(A1)$ contains three occurrences of the symbol v, namely $v(t +_b u)$, $v(t)$ and $v(u)$. The axiom is instantiated only if the term $t +_b u$ occurs in F. But this implies that all the terms t, u and $t +_b u$ already occur in F. The proofs regarding the rules for instantiation of other axioms proceeded similarly. The case when term $v(t)$ occurs in F itself is straightforward. □

Lemma 2. *A term of the form $\big(v(t)\big)_b$ occurs in F^* if and only if the bounded integer term t occurs in F.*

Proof. First assume a bounded integer term t occurs in F. Then it is a bounded integer subterm of F and an instance of the axiom $(A6')$ for t containing the term $\big(v(t)\big)_b$ occurs in F^*. Thus $\big(v(t)\big)_b$ occurs in F^*.

Now assume a term of the form $\big(v(t)\big)_b$ occurs in F^*. A term of this form can appear in axiom instantiations only if the term t occurs in F since the only axiom containing a term of this form is $(A6')$ and it is instantiated whenever t already occurs in F. The case when a term $\big(v(t)\big)_b$ directly occurs in F is straightforward. □

Lemma 3. *A term of the form $(n)_b$ occurs in F^* if and only if either it already occurs in F or the term n has the form $v(t)$ where t occurs in F.*

Proof. If a bounded integer term t occurs in F, then the term $\big(v(t)\big)_b$ occurs in F^* by the Lemma 2. Thus $(n)_b$ occurs in F^* where $n = v(t)$.

Now consider the case when a term $(n)_b$ occurs in F^* and it does not occur in F. We need to show that such a term can arise during instantiation only if it already occurs in F or if n has the form $v(t)$ where t occurs in F. We can show this by induction on instantiation rules. Only the axioms $(A5)$ and $(A6')$ contain the symbol $(\cdot)_b$. According to the instantiation rules in case of $(A5)$, $(n)_b$ already occurs in F. In case of $(A6')$ n has the required form. □

At this point we introduce some notation that we are using further in the proof. We denote the image of a subset X of

the domain A under function $f : A \to B$ as $f[X]$. We also introduce the following shorthand:

$$\{f(x) \mid P(x)\} \equiv f\big[\{x \mid P(x)\}\big],$$

where $\{x \mid P(x)\}$ is the common set-builder notation describing the set defined by the predicate $P(x)$ (the set of all elements satisfying the predicate). We only use such sets where the predicates $P(x)$ take individual terms, bounded or unbounded integers as input, so that the sets built using this notations are always well-defined. Now we use these notations to define the following two sets:

$$\begin{aligned} B^R &\equiv \left\{t^R \mid v(t) \in F^*\right\}, \\ C^R &\equiv \left\{n^R \mid (n)_b \in F^*\right\}, \end{aligned}$$

where, as was noticed above, t^R and n^R denote the valuations of terms t and n in the realization R. As the realization R contains partial models for functions v and $(\cdot)_b$ defined on those elements of the corresponding domains that are denoted by terms occurring in F^*, we can consider functions v^R and $(\cdot)_b^R$ (we later call them realizations of functions v and $(\cdot)_b$ correspondingly) well-defined on their corresponding domains B^R and C^R. Now let's prove some necessary properties of the realizations v and $(\cdot)_b$.

Lemma 4. $v^R[B^R] \subseteq C^R$.

Proof.

$$\begin{aligned} v^R[B^R] &= \left\{v^R(a) \mid a \in B^R\right\} \\ &= \left\{v^R(a) \mid a \in \left\{t^R \mid v(t) \in F^*\right\}\right\} \\ &= \left\{v^R(t^R) \mid v(t) \in F^*\right\} \\ &\quad \text{(by definition of set-builder)} \\ &= \left\{v^R(t^R) \mid t \in F\right\} \\ &\quad \text{(by Lemma 1)} \\ &= \left\{v^R(t^R) \mid \big(v(t)\big)_b \in F^*\right\} \\ &\quad \text{(by Lemma 2)} \\ &= \left\{\big(v(t)\big)^R \mid \big(v(t)\big)_b \in F^*\right\} \\ &\quad \text{(by definition of valuation)} \\ &= \left\{n^R \mid (n)_b \in F^* \wedge \exists t.\ n = v(t)\right\} \\ &\quad \text{(by definition of set-builder)} \\ &\subseteq \{n \mid (n)_b \in F^*\} \\ &= C^R. \end{aligned}$$

□

Lemma 5. $v^R[B^R] \subseteq [L, U]$

Proof. By definition of B^R, for any integer value c such that $c \in v^R[B^R]$ we have $c = v^R(t^R)$, where $v(t) \in F^*$. By Lemma 1 this implies $t \in F$. This, in turn, implies that the axiom $(A4)$ was instantiated with the term t, i. e. $L \le v(t) \le U$ is a ground (not occurring under any symbol other than $\wedge$) subterm of F^*. Since R is a model of F^*, $L \le v^R(t^R) \le U$ and thus $L \le c \le U$ for any $c \in B^R$. □

Lemma 6. $v^R[B^R] \subseteq C^R \cap [L, U]$.

Proof. Directly from Lemmas 4 and 5. □

Lemma 7. *v^R is injective (on its domain B^R).*

Proof. Take any arbitrary bounded integer values x and y from the set B^R. Then we have $x = t^R$, $y = u^R$, $v(t) \in F^*$ and $v(u) \in F^*$. Then by Lemma 1, $t \in F$ and $u \in F$ and thus the axiom $(A6')$ was instantiated with these terms ensuring the predicates $\big(v(t)\big)_b = t$ and $\big(v(u)\big)_b = t$ are ground subterms of F^*. Since R is a model of F^*, we have $\big(v^R(t^R)\big)_b^R = t^R$ and $\big(v^R(u^R)\big)_b^R = u^R$ and since R is a model in the QF_UFLIA logic that includes congruence of uninterpreted functions (such as $(\cdot)_b$), we have $v^R(t^R) = v^R(u^R) \implies \big(v^R(t^R)\big)_b^R = \big(v^R(u^R)\big)_b^R$, or, equivalently, $v^R(t^R) = v^R(u^R) \implies t^R = u^R$ i. e. $v^R(x) = v^R(y) \implies x = y$ for any x and y from the set B^R. □

Lemma 8. *v^R is surjective on $C^R \cap [L, U]$.*

Proof.

$$\begin{aligned} C^R &= \left\{n^R \mid (n)_b \in F^*\right\} \\ &\quad \text{(by definition of } C^R\text{)} \\ &= \left\{n^R \mid (n)_b \in F\right\} \cup \left\{\big(v(t)\big)^R \mid t \in F\right\} \\ &\quad \text{(by Lemma 3)} \\ &= \left\{n^R \mid (n)_b \in F\right\} \cup \left\{\big(v(t)\big)^R \mid v(t) \in F^*\right\} \\ &\quad \text{(by Lemma 1)} \\ &= \left\{n^R \mid (n)_b \in F\right\} \cup \left\{v^R(t^R) \mid v(t) \in F^*\right\} \\ &\quad \text{(by definition of valuation)} \\ &= \left\{n^R \mid (n)_b \in F\right\} \cup v\big[\left\{t^R \mid v(t) \in F^*\right\}\big] \\ &\quad \text{(by definitions of set-builder and image of a set)} \\ &= \left\{n^R \mid (n)_b \in F\right\} \cup v[B^R] \\ &\quad \text{(by definition of } B^R\text{)}. \end{aligned}$$

Thus to prove that $C^R \cap [L, U] \subseteq v^R[B^R]$ we need to show that $\left\{n^R \mid (n)_b \in F\right\} \cap [L, U] \subseteq v^R[B^R]$. Consider any term n such that $(n)_b \in F$. Since $(n)_b \in F$, $(A5)$ was instantiated with n and thus $L \le n \le U \implies v\big((n)_b\big) = n$ is a ground subterm of F^* and, as R is a model of F^*, $L \le n^R \le U \implies v^R\big((n^R)_b^R\big) = n^R$. So if $n^R \in [L, U]$ then $n^R = v^R\big((n^R)_b^R\big) \in \left\{v^R\big((n^R)_b^R\big) \mid (n)_b \in F\right\}$. Thus

$$\left\{n^R \mid (n)_b \in F\right\} \cap [L, U]$$

$$
\begin{aligned}
&\subseteq \left\{v^R\left((n^R)^R_b\right) \mid (n)_b \in F\right\}\\
&= \left\{v^R\left(\left((n)_b\right)^R\right) \mid (n)_b \in F\right\}\\
&\quad \text{(by definition of valuation)}\\
&= \left\{v^R(t^R) \mid t \in F \wedge \exists n.\ t = (n)_b\right\}\\
&\quad \text{(by definition of set-builder)}\\
&\subseteq \left\{v^R(t^R) \mid t \in F\right\}\\
&= \left\{v^R(t^R) \mid v(t) \in F^*\right\}\\
&\quad \text{(by Lemma 1)}\\
&= v^R\left[\left\{t^R \mid v(t) \in F^*\right\}\right]\\
&\quad \text{(by definition of set-builder and image of a set)}\\
&= v^R[B^R]\\
&\quad \text{(by definition of } B^R\text{).}
\end{aligned}
$$

□

Lemma 9. *v^R is a bijectoin between B^R and $C^R \cap [L, U]$ and $(\cdot)^R_b$ is its unique inverse.*

Proof. The fact that v^R is a bijectoin between B^R and $C^R \cap [L, U]$ follows directly from Lemmas 7 and 8. Thus v^R has a unique inverse. This inverse can be uniquely characterized by the equation $(v^R)^{-1}\left(v^R(a)\right) = a$ for every $a \in B^R$. But for every $a \in B^R$, value a can be represented as t^R where $v(t) \in F^*$. By Lemma 1, $t \in F$ and so axiom $(A6')$ was instantiated with the term t. So $\left(v^R(t^R)\right)^R_b = t^R$ and thus $\left(v^R(a)\right)^R_b = a$ for every $a \in B^R$. So the unique inverse coincides with the function $(\cdot)^R_b$. □

Lemma 10. $(\cdot)^R_b[C^R] \subseteq B^R$

Proof.

$$
\begin{aligned}
(\cdot)^R_b[C^R] &= (\cdot)^R_b\left[\left\{n^R \mid (n)_b \in F^*\right\}\right]\\
&\quad \text{(by definition of } C^R\text{)}\\
&= (\cdot)^R_b\left[\left\{n^R \mid (n)_b \in F\right\} \cup \left\{\left(v(t)\right)^R \mid t \in F\right\}\right]\\
&\quad \text{(by Lemma 3)}\\
&= (\cdot)^R_b\left[\left\{n^R \mid (n)_b \in F\right\}\right] \cup\\
&\quad (\cdot)^R_b\left[\left\{\left(v(t)\right)^R \mid t \in F\right\}\right]\\
&\quad \text{(by definition of set image)}\\
&= \left\{(n^R)^R_b \mid (n)_b \in F\right\} \cup \left\{\left(\left(v(t)\right)^R\right)^R_b \mid t \in F\right\}\\
&\quad \text{(by definition of set-builder and set image)}\\
&= \left\{\left((n)_b\right)^R \mid (n)_b \in F\right\} \cup \left\{\left(v^R(t^R)\right)^R_b \mid t \in F\right\}\\
&\quad \text{(by definition of valuation).}
\end{aligned}
$$

$$
\begin{aligned}
\left\{\left((n)_b\right)^R \mid (n)_b \in F\right\} &=\\
&= \left\{t^R \mid t \in F \wedge \exists n.\ t = (n)_b\right\}\\
&\quad \text{(by definition of set-builder)}\\
&\subseteq \left\{t^R \mid t \in F\right\}\\
&= \left\{t^R \mid v(t) \in F^*\right\}\\
&\quad \text{(by Lemma 1)}\\
&= B^R\\
&\quad \text{(by definition of } B^R\text{).}
\end{aligned}
$$

For every $t \in F$, $(A6')$ was instantiated with t, therefore $\left(v(t)\right)_b = t$ is a ground subterm of F^* and since R is a model of R, $\left(v^R(t^R)\right)^R_b = t^R$. Thus by definition of set-builder,

$$
\begin{aligned}
\left\{\left(v^R(t^R)\right)^R_b \mid t \in F\right\} &= \left\{t^R \mid t \in F\right\}\\
&= \left\{t^R \mid v(t) \in F^*\right\}\\
&\quad \text{(by Lemma 1)}\\
&= B^R\\
&\quad \text{(by definition of } B^R\text{).}
\end{aligned}
$$

Using the above representation of $(\cdot)^R_b[C^R]$ we finally have the required $(\cdot)^R_b[C^R] \subseteq B^R$. □

At this point we end up with the situation depicted in Figure 3. v^R is a bijection between B^R and $C^R \cap [L, U]$, $(\cdot)^R_b$ is its unique inverse on the domain B^R. $(\cdot)^R_b$ is defined on a larger domain C^R, where it might not be injective in general. However, its range is still the set B^R. To proceed further with our reconstruction of the model in BLIA, without loss of generality, we now consider the case $[L, U] \setminus C^R \neq \emptyset$. If the sets C^R and $[L, U]$ coincide, the following proof is still correct despite involving some vacuous reasoning (certain arguments are performed on empty domains and therefore can be safely omitted). Similarly, without loss of generality we assume the inequalities $L \leq 0 \leq U$.

We now consider the set $[L, U] \setminus C^R$, which has a finite cardinality not exceeding $U - L + 1$. We therefore arbitrarily choose some $\left|[L, U] \setminus C^R\right|$ distinct elements from any domain disjoint from B^R and establish a bijection v' between those opaque elements and the set $[L, U] \setminus C^R$. Let's denote the resulting set of the corresponding distinct opaque elements as $\mathbb{Z}'_b$ and the corresponding unique inverse of the bijection v' as $(\cdot)'_b$. Now we are ready to introduce the definition of the reconstructed model M of the original formula F that is shown in Figure 4. We proceed by establishing the necessary properties of this definition.

Lemma 11. *The reconstructed model M shown in Figure 4 is well-defined.*

Proof. The reconstructed model M shown in Figure 4 includes both the definition of the domain $\mathbb{Z}_b$ of bounded integers as well as the definitions for all the functions from the corresponding signature of the BLIA theory ($+_b$, $\times_b$, $\leq_b$, v

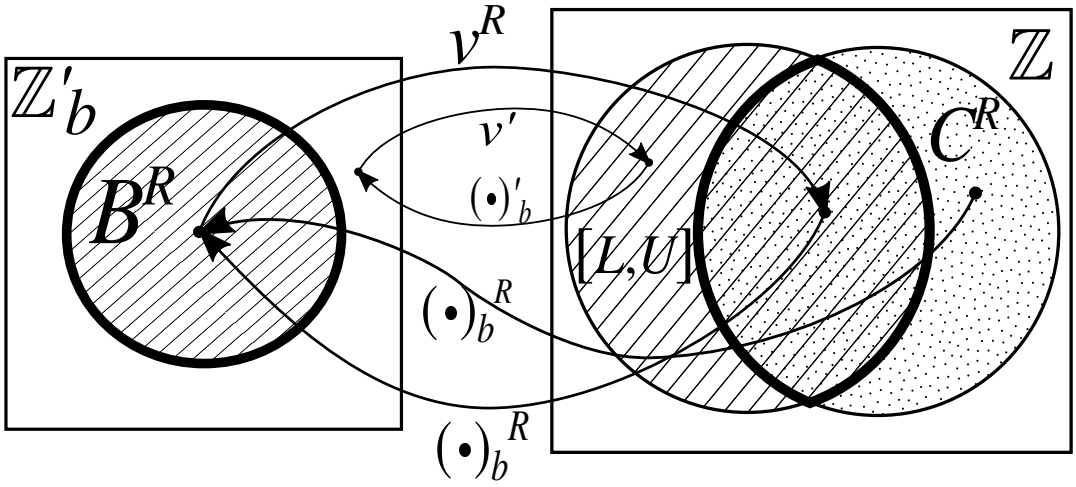

Fig. 3. Extending the bijection between B^R and $C^R \cap [L,U]$ to the whole sets $\mathbb{Z}_b = B^R \cup \mathbb{Z}'_b$ and $[L,U]$.

$$
\begin{aligned}
\mathbb{Z}_b^M &= B^R \cup \mathbb{Z}'_b,\\
v^M(a) &= \begin{cases} v^R(a), & a \in B^R,\\ v'(a), & a \notin B^R, \end{cases}\\
(c)_b^M &= \begin{cases} (c)_b^R, & c \in C^R,\\ (c)'_b, & c \in [L,U] \setminus C^R,\\ \epsilon\, \mathbb{Z}_b^M, & c \notin C^R \cup [L,U], \end{cases}\\
a +_b^M b &= \begin{cases} a +_b^R b, & (a,b) \in \left\{(t^R, u^R) \mid t +_b u \in F^*\right\},\\ \left(v^M(a) + v^M(b)\right)_b^M, & \\ & (a,b) \notin \left\{(t^R, u^R) \mid t +_b u \in F^*\right\},\\ & \quad L \le v^M(a) + v^M(b) \le U,\\ (0)_b, & \text{otherwise}, \end{cases}\\
c \times_b^M a &= \begin{cases} c \times_b^R a, & (c,a) \in \left\{(n, t^R) \mid n \times_b t \in F^*\right\},\\ \left(c \times v^M(a)\right)_b^M, & \\ & (c,a) \notin \left\{(n, t^R) \mid n \times_b t \in F^*\right\},\\ & \quad L \le c \times v^M(a) \le U,\\ (0)_b, & \text{otherwise}, \end{cases}\\
a \le_b^M b &= v^M(a) \le v^M(b).
\end{aligned}
$$

Fig. 4. Definition of the reconstructed model M in BLIA.

and $(\cdot)_b$). Thus we have to show that the functions defined as shown in the Figure do indeed map any combination of their arguments taken from the corresponding domains to the elements from the corresponding ranges. The cases for the function v and predicate $\le_b$ are trivial. However, we have to ensure that the chosen domain $\mathbb{Z}_b^M$ does indeed contain all the values taken by functions $(\cdot)_b^M$, $+_b^M$ and $\times_b^M$, otherwise our definition of the model is inconsistent.

Consider the cases for function $(\cdot)_b^M$. In the first case $(c)_b^R \in B^R$ for any $c \in C^R$ according to Lemma 10, thus $(c)_b^R \in \mathbb{Z}_b^M$. In the second case $(c)'_b \in \mathbb{Z}'_b \subseteq \mathbb{Z}_b^M$ by construction. In the third case $\epsilon\, \mathbb{Z}_b^M \in \mathbb{Z}_b^M$ by definition of the Hilbert epsilon-operator ϵ (choosing arbitrary element of a non-empty set) since $\left|\mathbb{Z}_b^M\right| \ge \left|\mathbb{Z}'_b\right| = \left|[L,U] \setminus C^R\right| > 0$ (in the general case either $\mathbb{Z}'_b$ or B^R is non-empty).

Now consider the function $+_b^M$. The second and the third cases ($\left(v^M(a) + v^M(b)\right)_b^M$ and $(0)_b$) are well-defined since $(c)_b^M$ is well-defined for any c as shown above. In the first case we proceed by induction on the instantiation rules and show that the term of the form $t +_b u$ can only appear during instantiation when it already occurs in F (the only relevant axiom $(A1)$ is instantiated whenever $t +_b u$ is in F). Then $t +_b u \in F^*$ implies $t +_b u \in F$ and by Lemma 1, $v(t +_b u) \in F^*$. Since $a = t^R$, $b = u^R$, $a +_b b = t^R +_b^R u^R = (t +_b u)^R \in B^R \subseteq \mathbb{Z}_b^M$ by definition of B^R. The proof for the function $\times_b^M$ is similar. □

Lemma 12. *The axioms* $(A4)$*,* $(A5)$ *and* $(A6')$ *of the BLIA theory hold in the model* M*.*

Proof. Consider axiom $(A4)$ for an arbitrary $a \in \mathbb{Z}_b^M = B^R \cup \mathbb{Z}'_b$. In case $a \in B^R$, $v^M(a) = v^R(a) \in [L,U]$ by Lemma 5. In case $a \in \mathbb{Z}'_b$, $v^M(a) = v'(a) \in [L,U] \setminus C^R \subseteq [L,U]$ by construction.

Consider the axiom $(A5)$ for an arbitrary integer c. As is stated by the axiom, we have to show the equality $v^M\left((c)_b^M\right) = c$ for any $c \in [L,U] = \left(C^R \cap [L,U]\right) \cup \left([L,U] \setminus C^R\right)$. In case $c \in C^R \cap [L,U]$ using the definitions in Figure 4 and Lemma 9 we get $(c)_b^M = (c)_b^R \in B^R$ and $v^M\left((c)_b^M\right) = v^R\left((c)_b^R\right) = c$. In case $c \in [L,U] \setminus C^R$ we get $(c)_b^M = (c)'_b \in \mathbb{Z}'_b$ and $v^M\left((c)_b^M\right) = v'\left((c)'_b\right) = c$ by construction.

Finally, in case of the axiom $(A6')$ we have to show $\left(v^M(a)\right)_b^M = a$ for any $a \in \mathbb{Z}_b^M = B^R \cup \mathbb{Z}'_b$. In case $a \in B^R$ by Lemma 9 we have $v^M(a) = v^R(a) \in C^R \cap [L,U]$ and $\left(v^M(a)\right)_b^M = \left(v^R(a)\right)_b^R = a$. In case $a \in \mathbb{Z}'_b$ we get $v^M(a) = v'(a) \in [L,U] \setminus C^R$ and $\left(v^M(a)\right)_b^M = \left(v'(a)\right)'_b = a$ by construction. □

Lemma 13. *The axioms* $(A1)$*,* $(A2)$ *and* $(A3)$ *of the BLIA theory hold in the model* M*.*

Proof. First we establish the axiom $(A1)$. We consider the cases in the definition of the $+_b^M$ operation.

Consider the case $(a,b) \in \{(t^R, u^R) \mid t +_b u \in F^*\}$. In the proof of the Lemma 11 (the last paragraph) we have shown that $t +_b u \in F^*$ implies $t +_b u \in F$. By the instantiation rule for the axiom $(A1)$ we conclude that $L \le v(t) + v(u) \le U \implies v(t +_b u) = v(t) + v(u)$ is a ground subterm of F^* and, since R is a model of F^*, we have $L \le v^R(t^R) + v^R(u^R) \le U \implies v^R(t^R +_b^R u^R) = v^R(t^R) + v^R(u^R)$. The fact $(a,b) \in \{(t^R, u^R) \mid t +_b u \in F^*\}$ implies $a = t^R$ and $b = u^R$, so $L \le v^R(a) + v^R(a) \le U \implies v^R(a +_b^R b) = v^R(a) + v^R(b)$. Now since $t +_b u \in F$, we have $t \in F$, $u \in F$, and by the Lemma 1, $v(t +_b u) \in F^*$, $v(t) \in F^*$ and $v(u) \in F^*$. This means $a +_b^R b = t^R +_b^R u^R = (t +_b u)^R \in B^R$, $a = t^R \in B^R$ and $b = t^R \in B^R$ by definition of B^R. Thus by the definitions in Figure 4, $L \le v^M(a) + v^M(b) \le U \implies v^M(a +_b^M b) = v^M(a) + v^M(b)$.

Now consider the case $L \le v^M(a) + v^M(b) \le U$. By the Lemma 12 the axiom $(A5)$ holds in M and thus $L \le c \le U \implies v^M((c)_b^M) = c$ for any c and, in particular, $L \le v^M(a) + v^M(b) \le U \implies v^M\Big((v^M(a) + v^M(b))_b^M\Big) = v^M(a) + v^M(b)$. By definition of $+_b^M$ in Figure 4 this implies $L \le v^M(a) + v^M(b) \le U \implies v^M(a +_b^M b) = v^M(a) + v^M(b)$.

Finally, in case $v^M(a) + v^M(b) \notin [L, U]$, the premise $L \le v^M(a) + v^M(b) \le U$ does not hold and thus the axiom $(A1)$ holds trivially.

The proof for the axiom $(A2)$ is similar and the case of $(A3)$ directly follows from the definition of $\le_b^M$ in Figure 4. □

Lemma 14. *The model M can be extended with uninterpreted constants occurring in F so that for any subterm $t \in F$ its interpretations in the model M and the realization R coincide i. e. $t^M = t^R$.*

Proof. Without loss of generality, we consider the case when t does not contain occurrences of uninterpreted functions with arity greater than zero. The extension of this proof to terms with uninterpreted functions (rather than constants) is straightforward. The proof is carried out by induction on the structure of the term t.

We start by considering zero-arity subterms i. e. constants. Interpreted constants (numbers) have the same fixed interpretation in both R and M. Let's choose the interpretation of uninterpreted constants occurring in F to be the same in both M and R. This is consistent since if a term t occurs in F, then by Lemma 1, $v(t)$ occurs in F^* and therefore $t^R \in B^R \subseteq \mathbb{Z}_b^M$.

Similarily, if t is an argument of function v occurring in F and $t^M = t^R$, then $t^R \in B^R$ and by definition in Figure 4 $v^M(t^M) = v^R(t^R)$, which by definition of valuation gives $\big(v(t)\big)^M = \big(v(t)\big)^R$. If $(n)_b$ occurs in F, then by Lemma 3 it also occurs in F^* and thus $n^R \in C^R$. Hence if $n^M = n^R$ then $\big((n)_b\big)^M = (n^M)_b^M = (n^R)_b^M = (n^R)_b^R = \big((n)_b\big)^R$.

If $n \times_b t$ occurs in F then by construction of the instantiated formula it also occurs in F^*. If, furthermore $t^M = t^R$, then $(n \times_b t)^M = n^M \times_b^M t^M = n^R \times_b^M t^R = n^R \times_b^R t^R = (n \times_b t)^R$. Same applies to $+_b$.

Finally, if $t \le_b u$ occurs in F then according to the instantiation rules, the axiom $(A3)$ was instantiated with t and u and thus $t^R \le_b^R u^R \iff v^R(t^R) \le v^R(u^R)$. By induction hypothesis we have $v^M(t^M) = v^R(t^R)$ and $v^M(u^M) = v^R(u^R)$. Moreover, by definition in Figure 4 we have $t^M \le_b^M u^M \iff v^M(t^M) \le v^M(u^M)$. Thus $t^M \le_b^M u^M \iff v^R(t^R) \le v^R(u^R) \iff t^R \le_b^R u^R$. □

Now the proof of the completeness theorem directly follows from the lemmas we proved at this point:

Theorem 2. *Every ground formula F in BLIA is satisfiable if and only if its translation F^* is satisfiable in QF_UFLIA.*

Proof. If F is satisfiable in BLIA, F^* is satisfiable in QF_UFLIA due to soundness of the instantiation procedure. Assume F^* is satisfiable in QF_UFLIA with the model R. Then by the Lemmas 11, 12 and 13, the reconstructed model M is a model of the BLIA theory. Moreover, by the Lemma 14, the whole formula F as a term has the same value in both R and M extended with the corresponding uninterpreted constants. Since F is a ground subterm of F^* and therefore is an identical truth in R, it is also satisfied by M. □

VII. Formalization

The completeness proof presented in the previous section was formalized in Isabelle/HOL proof assistant. In this formalization we employed untyped deep embedding of object-level formulas so that we could apply very simple induction with just two possible constructors (a function and a variable) on the structure of the formulas and reason about the interpretation of these formulas in different models of our object logic. There were two most notable differences distinguishing the fully formal proof in Isabelle/HOL from the still somewhat informal presentation given in the paper.

First, the use of untyped deep embedding, although it greatly simplifies the structural representation of the formulas, allows nonsensical malformed formulas to be represented and potentially interpreted in the object logic. For example, a formula like the following:

$$\big((a)_b \le n\big) + (b)_b \times_b 2,$$

where every function symbol is applied to at least one argument of an incompatible sort, can be represented in the embedding. To exclude such formulas we formulated explicit well-formedness constraints (in the form of a special predicate) and used them as preconditions in various lemmas as well as in our definition of satisfiability. In our approach an interpretation can only model a well-formed formula.

The second problem is not as obvious. The traditional definition of interpretation for a quantified formula (such as an axiom) involves an additional parameter usually called valuation μ that maps opaque variables to their corresponding interpretations. Thus using this definition was have to manipulate

not only the formulas and interpretations (models), but also the corresponding valuations. In this reasoning style establishing a seemingly trivial substitution principle, i. e. if $\forall a.\ F^M(a)$ we have $F^M(x^M) = \big(F(x)\big)^M$ if a is not free in x, requires defining an intermediate valuation $\mu' = \mu \circ [a \mapsto x^M]$ and evaluating the formula F in the model M with both the valuation μ and μ'. We instead employ a substitution-based definition of interpretation formalized as follows:

$$\forall a \in \mathcal{D}.\ F^M(a) \iff \Big(\forall t.\ vars(t) = \emptyset \Rightarrow \big([t/a]F(a)\big)^M\Big),$$

where any model M should satisfy the following restriction: $\forall a \in \mathcal{D}.\ \exists t.\ t^M = a$. Here $\mathcal{D}$ is the domain chosen for interpretation of terms (particularly, variables) of the corresponding sort, $[t/a]$ denotes substitution of the variable a with the term t. Since the substituted term t is ground, there's no need in special efforts for avoiding variable captures. Moreover, in our formalization we never needed nested quantifiers, so we used toplevel schematic variables instead of quantifiers and thus greatly simplified the formula representation. In fact, we further simplified our substitution-based approach by restricting the form of the ground bounded integer terms t in the substitution. If

$$\forall a \in \mathbb{Z}_b.\ \exists c \in \mathbb{Z}.\ (c)_b = a,$$

we can switch from quantification over bounded integer terms t to simple quantification over integers:

$$\Big(\forall t.\ vars(t) = \emptyset \Rightarrow \big([t/a]F(a)\big)^M\Big) \Leftrightarrow \forall c \in \mathbb{Z}.\ \big(F(c)_b\big)^M,$$

thus completely turning object-level quantification (in BLIA/QF_UFLIA) into the meta-level quantification in HOL itself. This is the most common form of "object-level" quantification that we use in our formal proof [9][1].

VIII. Implementation and evaluation

We implemented the instantiation-based decision procedure described in this paper within our Isabelle framework for extending SMT tactic with trigger-based quantifier instantiation. The framework is called TSMT [9] and is not described in this paper. Here we only briefly mention that it allows to perform preliminary saturation of the current goal (being a ground formula) with all quantifier instantiations triggered by matching subformulas of a certain form present in the goal. The quantifiers are provided as lemmas and the triggering subformulas can be specified using special lemma attributes. The framework also supports proof reconstruction for the saturated formula (using existing capabilities for Z3 proof reconstruction in Isabelle) as well as extracting the model (counterexample) of the current goal and showing the extracted model to the user.

Evaluating a specific tactic for interactive proof assistant is not a simple task as there are not many readily available benchmarks for such tools and the vast majority of available proofs are already specially tailored for the use of existing tactics with all their particular traits and limitations. The closest available Isabelle/HOL tactic that provides similar capabilities is the `uint_arith` tactic from the `HOL-Word` theory. It also tries to saturate the current goal with additional assumptions about arithmetic on bounded integers and relies the actual work of discharging the goal to the standard arithmetic tactics (simplification procedures, `presburger` and `linarith`). Nonetheless, unlike our approach, the translation implemented in the `uint_arith` tactic is not complete in general. The `uint_arith` tactic is not very widely used in most practical applications, however, as its capabilities are similar to that of `unat_arith` tactic (a similar tactic for natural numbers rather than integers), which is often used instead. Since arithmetic on natural numbers is not directly implemented in most SMT solvers, we based our tactic on the integer arithmetic. So to evaluate our implementation we took several most time-consuming goals solved by the `unat_arith` tactic and manually transformed them into the corresponding goals suitable for both `uint_arith` and our implementation (the transformation is mostly purely syntactic, apart from adding some missing constraints of the form $n \geq 0$). The goals were taken from the real-world examples featuring verified C implementations of functions `memcpy` and `quicksort` formalized within AutoCorres [10] framework. The examples included more than 130 invocations of the `uint_arith` tactic, of wihich we selected 11 most time-consuming (still originally finishing within several seconds). In addition to the existing examples we added several very simple lemmas about bounded integer arithmetic, where the `uint_arith` tactic fails to prove the goal.

Subgoal	CPU time, s	
	`uint_arith`	`(tsmt ubound)`
memcpy_wp'_1	0.366	0.329
memcpy_wp'_2	< 0.1	0.450
memcpy_word_1	< 0.1	0.320
memcpy_word_2	< 0.1	0.350
memcpy_wp'_3	0.201	0.320
memcpy_wp'_4	0.508	0.315
memcpy_wp'_5	0.250	0.509
partition_correct_1	0.468	0.581
partition_correct_2	0.401	0.441
qsort_unat_sub_sub1	< 0.1	0.526
quicksort_correct_1	0.251	0.493
simple_example	— (< 0.1)	0.196
simple_2a_mn_1_pl_b	— (1.202)	0.860
simple_div2_mul2	— (0.102)	0.756
simple_2_min_mn_1_pl_min	— (0.897)	< 0.1
simple_div3_mul3	— (0.145)	0.779

Fig. 5. Evaluation of TSMT BLIA and `uint_arith` tactics on sample formulas and real subgoals from AutoCorres examples.

The evaluation results are shown in Figure 5. Goal solving times are shown in seconds where they exceed 0.1, otherwise the time is marked as < 0.1. The time for goals not solved

[1]The corresponding Isabelle theory in the repository is called `TSMT_Bounded_Complete`.

by a tactic is marked as "—" together with the time required for the attempt before returning the resulting proof state (a saturated goal in case of `uint_arith`). Since Isabelle/HOL workflow is based on interactive formal documents, the solving time is important as it directly influences the overall time required for updating the document model upon every user interaction (especially for unstructured proofs). The results demonstrate that our implementation is not significantly slower than the existing tactic despite involving exhaustive quantifier instantiation, invocation of external provers (Z3) as well as parsing and reconstruction of the resulting proofs. Moreover, it provides several notable advantages, namely:

- It enjoys completeness property so it is guaranteed to solve any goal within a certain class, whereas the `uint_arith` tactic is not complete and its ability to solve a particular goal is not easily predictable in general;
- Due to being complete, our tactic is able not only to verify a correct goal, but, often more importantly, also to meaningfully refute incorrect goals within a certain class (where the tactic is complete). This ability is important in the context of iterative proof development where it is widely known that intermediate attempts to prove incorrect statements are more pervasive than failures due to incompleteness of the tactics/proof methods.
- Our tactic has approximately the same performance both in case of a successful proof attempt and a counterexample extraction, because it behaves the same in both cases, while `uint_arith` sequentially attempts to solve the saturated goal with all the registered arithmetic tactics and so can spend additional time on unsuccessful attempts.
- Our approach is easily extensible, since semantics of new function symbols can be easily supported by adding the corresponding lemmas with their triggers. In fact, the ease of extensibility of our tactic is comparable to that of the standard Isabelle simplifier. In the examples we extended our tactic with support for integer division, subtraction, maximum and minimum to match the capabilities of the standard arithmetic tactics in Isabelle.
- Extensing our tactic does not require ML programming, the tactic is implemented on top of the general TSMT tactic (main part of our instantiation framework) simply as a group of lemmas with the corresponding instantiation triggers, while `uint_arith` is implemented directly in Isabelle/ML.

The primary limitation of our current implementation in comparison with the existing `uint_arith` tactic is inability to properly handle quantifiers nested in the goal itself and not augmented with the corresponding triggers. Our tactic currently ignores any quantifiers present in the goal. The examples mentioned in Figure 5 are available in our example theory [9][2].

IX. Conclusion and future work

In this paper we presented the following contributions:

- An axiomatic definition of the theory of bounded linear integer arithmetic (BLIA) embedded into the logic of linear integer arithmetic and uninterpreted functions with equality and congruence (UFLIA);
- A method for deciding the satisfiability of formulas in BLIA based on translation of the formula into the QF_UFLIA logic, where existing proof-producing decision procedures can be applied. The translation is formulated as a trigger-based instantiation procedure imposing only linear overhead in formula size during translation;
- A proof of completeness for the proposed instantiation procedure;
- A formalization of the completeness proof in Isabelle/HOL;
- An implementation of the proposed decision procedure as an Isabelle/HOL proof method with support of proof reconstruction and counterexample extraction. The implementation was evaluated on several examples extracted from real applications of similar tactics and shown to have similar efficiency while enjoying completeness property;

Major directions of future work include developing some predictable (although incomplete) approach for handling quantifiers occurring in the goal without corresponding triggers (the strategies for quantifier handling inside existing provers are very efficient, but rarely predictable). Another major direction is formalization and evaluation of instantiation-based decision procedures for other decidable theories such as an efficiently decidable fragment of the theory of bit-vectors, modular linear integer arithmetic (with wrap-around semantics), a decidable fragment of a theory formalizing various operations on lists, a theory of address arithmetic with bounded addresses (but with C-like separation to disjoint memory blocks), theory of interpreted sets (for which there are several completeness proofs, but no formalization in a proof assistant) and other practically relevant theories and fragments.

[2]The corresponding Isabelle theory in the repository is called `TSMT_Bounded_Examples`.

X. Acknowledgement

The research was carried out with funding from the Ministry of Science and Higher Education of the Russian Federation (the project unique identifier is RFMEFI60719X0295).

References

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

[2] J. Dawson, "Isabelle theories for machine words," *Electronic Notes in Theoretical Computer Science*, vol. 250, no. 1, pp. 55 – 70, 2009, proceedings of the Seventh International Workshop on Automated Verification of Critical Systems (AVoCS 2007). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1571066109003302

[3] D. Babic and M. Musuvathi, "Modular arithmetic decision procedure," Tech. Rep. MSR-TR-2005-114, August 2005. [Online]. Available: https://www.microsoft.com/en-us/research/publication/modular-arithmetic-decision-procedure/

[4] N. Bjørner, A. Blass, Y. Gurevich, and M. Musuvathi, "Modular difference logic is hard," Tech. Rep. MSR-TR-2008-140, October 2008. [Online]. Available: https://www.microsoft.com/en-us/research/publication/modular-difference-logic-is-hard/

[5] B.-Y. Wang, "On the satisfiability of modular arithmetic formulae," in *Automated Technology for Verification and Analysis*, S. Graf and W. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 186–199.

[6] S. Böhme, "Proof reconstruction for Z3 in Isabelle/HOL," in *7th International Workshop on Satisfiability Modulo Theories (SMT '09)*, 2009.

[7] S. Böhme and T. Weber, "Fast LCF-style proof reconstruction for Z3," in *Interactive Theorem Proving*, M. Kaufmann and L. C. Paulson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 179–194.

[8] C. Barrett, P. Fontaine, and C. Tinelli, "The SMT-LIB Standard: Version 2.6," Department of Computer Science, The University of Iowa, Tech. Rep., 2017, available at `www.SMT-LIB.org`.

[9] R. Sadykov and M. Mandrykin, "Completeness of instantiation procedure for bounded linear integer arithmetic. Formalization in Isabelle/HOL," https://forge.ispras.ru/projects/tsmt/repository/, April 2020.

[10] D. Greenaway, J. Andronick, and G. Klein, "Bridging the gap: Automatic verified abstraction of C," in *Interactive Theorem Proving*, L. Beringer and A. Felty, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 99–115.