Unique Factorization in ACL2: Euclidean Domains

John R. Cowles
Department of Computer Science
University of Wyoming
Laramie, Wyoming
cowles@cs.uwyo.edu

Ruben Gamboa
Department of Computer Science
University of Wyoming
Laramie, Wyoming
ruben@cs.uwyo.edu

ABSTRACT

ACL2 is used to systematically study domains whose elements can be "uniquely" factored into products of "irreducible" elements. The best known examples of such domains are the positive integers, which can be factored into products of primes, and univariate polynomials with rational coefficients, which can be factored into products of irreducible polynomials. There are many other such domains.

Euclidean domains are an algebraic abstraction, of both the positive integers and the rational polynomials, in which the usual proofs of unique factorization, for both the integers and the polynomials, can be generalized.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*Mechanical theorem proving, Computational logic*

General Terms

Verification

Keywords

Boyer-Moore logic,unique factorization, Euclidean domains, ACL2

1. INTRODUCTION

Before the computational logic ACL2 [4], there was the version of the Boyer-Moore Logic known as NQTHM [2], and before NQTHM, there was the version of the logic called THM [1]. In 1979, when the book *A Computational Logic* [1], describing THM, was published, the deepest and hardest theorem, yet verified in the logic, was the Unique Prime Factorization Theorem:

THEOREM 1. *Every positive integer can be factored into a finite product of primes. Moreover, this product is unique except for the order of the primes.*

Mathematicians have identified many domains, other than the positive integers, that can also be said to have some form of unique factorization. For example, polynomials, of a single variable, with rational coefficients, can each be factored into a product of irreducible polynomials. Moreover, this product is essentially unique except for the order of the irreducible polynomials.

We are using ACL2 to carry out a systematic study of such domains. We begin by studying domains where the standard proof, of unique factorization for the positive integers, can be easily generalized. This explains the second half of our title: Mathematicians call those domains, where the standard proof can be readily carried out, Euclidean [3, 8, 7]. Also, we anticipate eventually extending our study to domains where unique factorization holds, but where the proof is not a straight-forward generalization of the standard proof for the positive integers. This is part of an even larger effort formalizing those parts of computational number theory used in modern cryptography.

2. THE STANDARD PROOF

The statement of the theorem, in both THM and ACL2, uses finite lists of positive integers to represent factorizations of positive integers. Explicit functions are used to overcome the complete lack of existential quantification in THM.

The statement of the theorem has two parts.

Existence. This part states that an explicitly defined function produces a prime factorization of its positive integer input argument.

Uniqueness. This part states that any two lists of primes, whose products are equal integers, are equal when the lists are treated as bags (or multisets). That is, the two lists have the same members and each member has the same multiplicity in the two lists.

Existence depends on the Proper Factor binary relation on the positive integers being well-founded.

Definition 1. Positive integer x is a factor (or divisor) of positive integer z just in case there is a positive integer y such that $x \cdot y = z$.

Definition 2. Positive integer x is a proper factor (or proper divisor) of positive integer z just in case x is a factor of z and $1 < x < z$.

Thus, over the positive integers, there are no infinite chains of proper factors.

Uniqueness depends on this key property of primes.

ACL2 '06 Seattle, Washington USA
Copyright 2006 ACL2 Steering Committee 0-9788493-0-2/06/08.

THEOREM 2. *If prime p is a factor of the product of positive integers $a \cdot b$, then either p is a factor of a or p is a factor of b.*

The standard proof of this key theorem employs the theory of the greatest common divisor of two positive integers. The classical way to compute Greatest Common Divisors is by use of the Euclidean Algorithm.

An ACL2 book, described in Appendix A, containing a version of the original THM proof can be found in the file `prime-fac.lisp`. An explicitly defined, but rather inefficient, function in this book can, of course, be executed and actually produces prime factorizations of positive integers.

3. EUCLIDEAN DOMAINS

An Euclidean Domain is a set $\mathbb{E}$ with two binary operations, $+_{\mathbb{E}}$ and $\cdot_{\mathbb{E}}$, an unary operation, $-_{\mathbb{E}}$, and a set element $0_{\mathbb{E}}$ that satisfy these expected algebraic laws: For all $X, Y, Z \in \mathbb{E}$,

Closure. $X +_{\mathbb{E}} Y \in \mathbb{E}$, $X \cdot_{\mathbb{E}} Y \in \mathbb{E}$, $-_{\mathbb{E}} X \in \mathbb{E}$, $0_{\mathbb{E}} \in \mathbb{E}$;

Commutative. $X +_{\mathbb{E}} Y = Y +_{\mathbb{E}} X$, $X \cdot_{\mathbb{E}} Y = Y \cdot_{\mathbb{E}} X$;

Associative. $(X +_{\mathbb{E}} Y) +_{\mathbb{E}} Z = X +_{\mathbb{E}} (Y +_{\mathbb{E}} Z)$,
$(X \cdot_{\mathbb{E}} Y) \cdot_{\mathbb{E}} Z = X \cdot_{\mathbb{E}} (Y \cdot_{\mathbb{E}} Z)$;

Distributive. $X \cdot_{\mathbb{E}} (Y +_{\mathbb{E}} Z) = (X \cdot_{\mathbb{E}} Y) +_{\mathbb{E}} (X \cdot_{\mathbb{E}} Z)$;

Identity. $0_{\mathbb{E}} +_{\mathbb{E}} X = X$:

Inverse. $X +_{\mathbb{E}} (-_{\mathbb{E}} X) = 0_{\mathbb{E}}$:

Euclidean multiplication, $\cdot_{\mathbb{E}}$, behaves like integer multiplication to the extent that it satisfies

- if $X \cdot_{\mathbb{E}} Y = 0_{\mathbb{E}}$, then $(X = 0_{\mathbb{E}} \vee Y = 0_{\mathbb{E}})$.

These domains are called Euclidean because they come equipped with enough additional structure to implement the Euclidean Algorithm for the computation of Greatest Common Divisors. That is, Euclidean division, with quotients and remainders, must be possible. Thus, there are two binary functions, `quot` and `rem`, and an unary function, `Size`, from $\mathbb{E}$ into the nonnegative integers, that satisfy: For all $X, Y \in \mathbb{E}$, with $Y \neq 0_{\mathbb{E}}$,

Closure. $\texttt{quot}(X, Y) \in \mathbb{E}$, $\texttt{rem}(X, Y) \in \mathbb{E}$:

Division Property. $X = Y \cdot_{\mathbb{E}} \texttt{quot}(X, Y) +_{\mathbb{E}} \texttt{rem}(X, Y)$,
$\texttt{rem}(X, Y) = 0_{\mathbb{E}} \vee \texttt{Size}(\texttt{rem}(X, Y)) < \texttt{Size}(Y)$.

In ACL2, these properties are consistently axiomatized using `ENCAPSULATE`. In fact, the ACL2 axiomatization is slightly generalized by replacing $=$ with an equivalence relation with the appropriate congruence properties.

3.1 Multiplicative Identity

THEOREM 3. *Every Euclidean Domain has a multiplicative identity.*

Thus, there always is a domain element, $1_{\mathbb{E}}$, so that for any $X \in \mathbb{E}$,

$$1_{\mathbb{E}} \cdot_{\mathbb{E}} X = X.$$

PROOF. If the domain contains only one element $0_{\mathbb{E}}$, then let $1_{\mathbb{E}}$ be $0_{\mathbb{E}}$.

Otherwise the domain contains a nonzero element. Choose a nonzero domain element, b, of smallest possible `Size`.

By the Division property, b divides every domain element (because the remainder must be $0_{\mathbb{E}}$).

Let $1_{\mathbb{E}}$ be $\texttt{quot}(b, b)$. Then $b \cdot_{\mathbb{E}} 1_{\mathbb{E}}$ is $b \cdot_{\mathbb{E}} \texttt{quot}(b, b)$ which is b (since the remainder is $0_{\mathbb{E}}$).

For any domain element x, $b \cdot_{\mathbb{E}} \texttt{quot}(x, b) = x$. Therefore

$$\begin{aligned} 1_{\mathbb{E}} \cdot_{\mathbb{E}} x &= 1_{\mathbb{E}} \cdot_{\mathbb{E}} (b \cdot_{\mathbb{E}} \texttt{quot}(x, b)) \\ &= (1_{\mathbb{E}} \cdot_{\mathbb{E}} b) \cdot_{\mathbb{E}} \texttt{quot}(x, b) \\ &= b \cdot_{\mathbb{E}} \texttt{quot}(x, b)) \\ &= x. \end{aligned}$$

□

Book 1, of many ACL2 Euclidean Domain books described in Appendix A, in `ed1.lisp` formalizes and verifies this proof in ACL2.

Perhaps the most interesting step to translate into ACL2 is, "Choose a nonzero domain element, b, of smallest possible `Size`." In the ACL2 proof, `defchoose` is first used to choose a nonzero domain element `(a)` (assuming it exists). Then for every nonnegative integer `n`, `defchoose` is used to choose a nonzero domain element `(b n)` with `Size n` (if such an element exists). Then at least one member of this finite list, `(b 0), (b 1), ..., (b (Size (a)))`, must be a nonzero domain element satisfying `(equal (Size (b j) j))`. The first such list element must, in fact, be a nonzero domain element of smallest possible `Size`.

3.2 Well-founded Factors

A proof, that, over nonzero Euclidean domain elements, there are no infinite chains of proper factors, is greatly simplified if this Multiplicative Size Property holds:

- $\texttt{Size}(x) \leq \texttt{Size}(x \cdot_{\mathbb{E}} y)$, for all nonzero domain elements x and y.

Unfortunately, there are Euclidean domains (see the example below) that do not satisfy this property. However, the next theorem shows there is no loss of generality in assuming the Multiplicative Size Property always holds in a given Euclidean domain.

THEOREM 4. *If the Euclidean domain*

$$(\mathbb{E}, +_{\mathbb{E}}, \cdot_{\mathbb{E}}, -_{\mathbb{E}}, 0_{\mathbb{E}}, 1_{\mathbb{E}}, \texttt{quot}_1, \texttt{rem}_1, \texttt{Size}_1),$$

*does not satisfy the **Multiplicative Size Property**, then "new" operations* `quot`*,* `rem`*, and* `Size` *can always be defined so that*

$$(\mathbb{E}, +_{\mathbb{E}}, \cdot_{\mathbb{E}}, -_{\mathbb{E}}, 0_{\mathbb{E}}, 1_{\mathbb{E}}, \texttt{quot}, \texttt{rem}, \texttt{Size}),$$

*is an Euclidean domain that does satisfy the **Multiplicative Size Property**.*

PROOF. If the domain contains only one element $0_{\mathbb{E}}$, then the Multiplicative Size Property holds vacuously.

Otherwise the domain contains a nonzero element. For each nonzero domain element, x, choose a nonzero domain element $c(x)$ such that $\texttt{Size}_1(x \cdot_{\mathbb{E}} c(x))$ is as small as possible. Define

$$\begin{aligned} \texttt{Size}(x) &= \texttt{Size}_1(x \cdot_{\mathbb{E}} c(x)) \\ \texttt{quot}(x, y) &= c(y) \cdot_{\mathbb{E}} \texttt{quot}_1(x, y \cdot_{\mathbb{E}} c(y)) \\ \texttt{rem}(x, y) &= \texttt{rem}_1(x, y \cdot_{\mathbb{E}} c(y)) \end{aligned}$$

Then the Division Property remains true of these “new” operations.

$$\begin{aligned} x &= [(y \cdot_{\mathbb{E}} c(y)) \cdot_{\mathbb{E}} \mathtt{quot}_1(x, y \cdot_{\mathbb{E}} c(y))] +_{\mathbb{E}} \mathtt{rem}_1(x, y \cdot_{\mathbb{E}} c(y)) \\ &= y \cdot_{\mathbb{E}} [c(y) \cdot_{\mathbb{E}} \mathtt{quot}_1(x, y \cdot_{\mathbb{E}} c(y))] +_{\mathbb{E}} \mathtt{rem}_1(x, y \cdot_{\mathbb{E}} c(y)) \\ &= y \cdot_{\mathbb{E}} \mathtt{quot}(x, y) +_{\mathbb{E}} \mathtt{rem}(x, y) \end{aligned}$$

and either $\mathtt{rem}(x, y) = \mathtt{rem}_1(x, y \cdot_{\mathbb{E}} c(y)) = 0_{\mathbb{E}}$ or

$$\begin{aligned} \mathtt{Size}(\mathtt{rem}(x, y)) &= \mathtt{Size}_1(\mathtt{rem}(x, y) \cdot_{\mathbb{E}} c(\mathtt{rem}(x, y))) \\ &\leq \mathtt{Size}_1(\mathtt{rem}(x, y) \cdot_{\mathbb{E}} 1_{\mathbb{E}}) \\ &= \mathtt{Size}_1(\mathtt{rem}(x, y)) \\ &= \mathtt{Size}_1(\mathtt{rem}_1(x, y \cdot_{\mathbb{E}} c(y))) \\ &< \mathtt{Size}_1(y \cdot_{\mathbb{E}} c(y)) \\ &= \mathtt{Size}(y) \end{aligned}$$

The “new” operations also satisfy the Multiplicative Size Property.

$$\begin{aligned} \mathtt{Size}(x) &= \mathtt{Size}_1(x \cdot_{\mathbb{E}} c(x)) \\ &\leq \mathtt{Size}_1(x \cdot_{\mathbb{E}} [y \cdot_{\mathbb{E}} c(x \cdot_{\mathbb{E}} y)]) \\ &= \mathtt{Size}_1([x \cdot_{\mathbb{E}} y] \cdot_{\mathbb{E}} c(x \cdot_{\mathbb{E}} y)) \\ &= \mathtt{Size}(x \cdot_{\mathbb{E}} y) \end{aligned}$$

□

Book 2a, described in Appendix A, in `ed2a.lisp` formalizes and verifies this proof in ACL2.

The most interesting step to translate into ACL2 is, “For each nonzero domain element, x, choose a nonzero domain element $c(x)$ such that $\mathtt{Size}_1(x \cdot_{\mathbb{E}} c(x))$ is as small as possible.” In the ACL2 proof, for every nonzero domain element `x` and every nonnegative integer `n`, `defchoose` is used to choose a nonzero domain element `(c x n)` with $\mathtt{Size}_1(\mathtt{x} \cdot_{\mathbb{E}}$ `(c x n)`$) =$ `n` (if such an element exists).

Using `Size1` and `*_e` as the ACL2 names for $\mathtt{Size}_1$ and $\cdot_{\mathbb{E}}$, and observing that $1_{\mathbb{E}}$ is a nonzero domain element with $\mathtt{Size}_1(\mathtt{x} \cdot_{\mathbb{E}} 1_{\mathbb{E}}) = \mathtt{Size}_1(\mathtt{x})$, it must be that `(c x (Size1 x))` is a nonzero domain element with the property, `(equal (Size1 (*_e x (c x (Size1 x))))(Size1 x))`.

So at least one member of this finite list, `(c x 0), (c x 1), ..., (c x (Size1 x))`, must be a nonzero domain element satisfying `(equal (Size1 (*_e x (c x j))) j)`. The first such element in the list must, in fact, be a nonzero domain element such that `(Size1 (*_e x (c x j)))` is as small as possible.

3.2.1 Example

Here is an example of an Euclidean domain

$$(\mathbb{E}, +_{\mathbb{E}}, \cdot_{\mathbb{E}}, -_{\mathbb{E}}, 0_{\mathbb{E}}, 1_{\mathbb{E}}, \mathtt{quot}_1, \mathtt{rem}_1, \mathtt{Size}_1)$$

that does not satisfy the Multiplicative Size Property.

Let $\mathbb{E}$ be the integers, with the usual binary operations of addition and multiplication, for $+_{\mathbb{E}}$ and $\cdot_{\mathbb{E}}$; the usual unary minus for $-_{\mathbb{E}}$; 0 for $0_{\mathbb{E}}$; and 1 for $1_{\mathbb{E}}$.

For $\mathtt{quot}_1$, use the Common Lisp function `round` and for $\mathtt{rem}_1$, use `rnd-rem`, defined by

(rnd-rem x y) = `(- x (* y (round x y)))`.

For $\mathtt{Size}_1$, use `abs-3` defined by

$$\mathtt{(abs\text{-}3\ x)} = \begin{cases} 2 & \text{if } x = 3 \\ |x| & \text{otherwise} \end{cases}$$

Book 2b, described in Appendix A, in `ed2b.lisp` verifies that, with these definitions,

$$(\mathbb{E}, +_{\mathbb{E}}, \cdot_{\mathbb{E}}, -_{\mathbb{E}}, 0_{\mathbb{E}}, 1_{\mathbb{E}}, \mathtt{quot}_1, \mathtt{rem}_1, \mathtt{Size}_1)$$

is an Euclidean domain, but the Multiplicative Size Property is violated: For example,

`(abs-3 -3)` $= 3 > 2 =$ `(abs-3 (* -3 -1))`.

3.3 Greatest Common Divisors

The definition of the “is a factor of” binary relation makes sense in any Euclidean domain, $\mathbb{E}$.

Definition 3. Element $x \in \mathbb{E}$ is a factor (or divisor) of $z \in \mathbb{E}$ just in case there is a $y \in \mathbb{E}$ such that $x \cdot_{\mathbb{E}} y = z$.

Definition 4. Element $g \in \mathbb{E}$ is a common factor (or divisor) of $x \in \mathbb{E}$ and $y \in \mathbb{E}$ just in case g is factor of both x and y. A greatest common divisor, g, of x and y is a common factor such that any other common factor of x and y is also a factor of g.

In an Euclidean domain with the Multiplicative Size Property, a greatest common divisor will have the largest size of any of the common factors.

THEOREM 5. *For all nonzero $x \in \mathbb{E}$ and all $y \in \mathbb{E}$, there are $a, b \in \mathbb{E}$ such that $a \cdot_{\mathbb{E}} x +_{\mathbb{E}} b \cdot_{\mathbb{E}} y$ is a greatest common divisor of x and y.*

PROOF. Let $x \neq 0_{\mathbb{E}}$ and y be elements of $\mathbb{E}$ and let

$$L = \{d = a \cdot_{\mathbb{E}} x +_{\mathbb{E}} b \cdot_{\mathbb{E}} y \mid d \neq 0_{\mathbb{E}} \land a \in \mathbb{E} \land b \in \mathbb{E}\}.$$

Since $x = 1_{\mathbb{E}} \cdot_{\mathbb{E}} x +_{\mathbb{E}} 0_{\mathbb{E}} \cdot_{\mathbb{E}} y \in L$, $L \neq \emptyset$. Choose $a(x, y) \in \mathbb{E}$ and $b(x, y) \in \mathbb{E}$ so that $\mathtt{Size}(a(x, y) \cdot_{\mathbb{E}} x +_{\mathbb{E}} b(x, y) \cdot_{\mathbb{E}} y)$ is as small as possible. Then $g = a(x, y) \cdot_{\mathbb{E}} x +_{\mathbb{E}} b(x, y) \cdot_{\mathbb{E}} y$ must be a greatest common divisor of x and y:

- g is a factor of x.
  If not, then when x is divided by g, the remainder is not zero. Then the following are contradictory.
  - $g \in L$ has the smallest possible `Size`.
  - $\mathtt{Size}(\mathtt{rem}(x, g)) < \mathtt{Size}(g)$
  - $\mathtt{rem}(x, g) \in L$, since

    $$\begin{aligned} \mathtt{rem}(x, g) &= x -_{\mathbb{E}} g \cdot_{\mathbb{E}} \mathtt{quot}(x, g) \\ &= x -_{\mathbb{E}} (a(x, y) \cdot_{\mathbb{E}} x +_{\mathbb{E}} b(x, y) \cdot_{\mathbb{E}} y) \\ &\quad \cdot_{\mathbb{E}} \mathtt{quot}(x, g) \\ &= x -_{\mathbb{E}} a(x, y) \cdot_{\mathbb{E}} \mathtt{quot}(x, g) \cdot_{\mathbb{E}} x \\ &\quad -_{\mathbb{E}} b(x, y) \cdot_{\mathbb{E}} \mathtt{quot}(x, g) \cdot_{\mathbb{E}} y \\ &= (1_{\mathbb{E}} -_{\mathbb{E}} a(x, y) \cdot_{\mathbb{E}} \mathtt{quot}(x, g)) \cdot_{\mathbb{E}} x \\ &\quad +_{\mathbb{E}} b(x, y) \cdot_{\mathbb{E}} (-_{\mathbb{E}} \mathtt{quot}(x, g)) \cdot_{\mathbb{E}} y \end{aligned}$$

- g is a factor of y.
  Similar to the previous item.

- Any common factor of x and y must also be a factor of any linear combination of x and y. Therefore any other common factor of x and y is also a factor of g.

□

The most interesting step of this proof to translate into ACL2 is, "Choose $a(x, y) \in \mathbb{E}$ and $b(x, y) \in \mathbb{E}$ so that $\texttt{Size}(a(x,y)\cdot_{\mathbb{E}} x +_{\mathbb{E}} b(x,y)\cdot_{\mathbb{E}} y)$ is as small as possible." In the ACL2 proof, for every nonzero domain element `x`, every domain element `y`, and every nonnegative integer `n`, `defchoose` is used to choose a pair of domain elements `((a x y n) (b x y n))` with $\texttt{(a x y n)} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y n)} \cdot_{\mathbb{E}} \texttt{y} \neq 0_{\mathbb{E}}$ and $\texttt{Size}(\texttt{(a x y n)} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y n)} \cdot_{\mathbb{E}} \texttt{y}) = \texttt{n}$ (if such a pair exists).

So at least one member of this finite list,

$$\begin{gathered}\texttt{(a x y 0)} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y 0)} \cdot_{\mathbb{E}} \texttt{y},\\ \texttt{(a x y 1)} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y 1)} \cdot_{\mathbb{E}} \texttt{y},\\ \ldots,\\ \texttt{(a x y (Size x))} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y (Size x))} \cdot_{\mathbb{E}} \texttt{y}\end{gathered}$$

must be a nonzero domain element satisfying

$$\texttt{Size}(\texttt{(a x y j)} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y j)} \cdot_{\mathbb{E}} \texttt{y}) = \texttt{j}.$$

The first such element in the list must, in fact, be a nonzero domain element such that

$$\texttt{Size}(\texttt{(a x y j)} \cdot_{\mathbb{E}} \texttt{x} +_{\mathbb{E}} \texttt{(b x y j)} \cdot_{\mathbb{E}} \texttt{y})$$

is as small as possible.

3.3.1 Key Prime Property

Suppose p is an irreducible (or prime) element of $\mathbb{E}$ and for $x, y \in \mathbb{E}$, p is a factor of $x \cdot_{\mathbb{E}} y$, but p is not a factor of x. We show that p must be a factor of y.

Proof. Since p is not a factor of x, it can be shown that $1_{\mathbb{E}}$ is a greatest common divisor of x and p. So there are domain elements, a and b such that $a \cdot_{\mathbb{E}} x +_{\mathbb{E}} b \cdot_{\mathbb{E}} p = 1_{\mathbb{E}}$. Then $a \cdot_{\mathbb{E}} x \cdot_{\mathbb{E}} y +_{\mathbb{E}} b \cdot_{\mathbb{E}} p \cdot_{\mathbb{E}} y = 1_{\mathbb{E}} \cdot_{\mathbb{E}} y = y$. Since p is a factor of both $x \cdot_{\mathbb{E}} y$ and p, p must also be a factor of y. □

3.4 Verified Euclidean Domains

Book 3, described in Appendix A, in `ed3.lisp` uses the Multiplicative Size Property to verify the existence of finite factorizations into irreducible domain elements, in an arbitrary Euclidean domain, axiomatized using an `encapsulate`. Book 3 also uses Greatest Common Divisors and the Key Prime Property to verify the uniqueness of such factorizations. Instances of particular domains, with unique factorization, can then be established via `functional instantiation` of this general theory of Euclidean domains.

There are many choices for quotient and remainder that turn the integers into an Euclidean domain. Two choices are `floor` and `mod`, and also, `truncate` and `rem`. Two other choices use `ceiling` or `round` for the quotient. In all four of these examples, absolute value, `abs`, can be used for `Size`. See the descriptions of Books 4aa, 4ba, 4ca, and 4da in Appendix A.

The Gaussian integers are the complex numbers with integer real and imaginary parts. They can be turned into an Euclidean domain by taking `Size` to be the square of the complex absolute value and basing quotient on separately rounding the real and imaginary parts of the complex quotient.

3.4.1 Executable Skolem Functions

In Book 3, the proofs of the existence and uniqueness of the factorizations make extensive use of `defchoose` and first-order quantification in the form of `defun-sk`. As a consequence, since the choice functions and the Skolem functions are not executable, the above functional instantiations do not produce executable functions that can be used to actually compute the factorizations that have just been shown to exist. This is in contrast to the proof, for the positive integers, in `prime-fac.lisp`, which did provide an executable function that factors positive integers.

We want to use `functional instantiation` of Book 3 to prove unique factorization in specific instances. We also want to end up with executable functions that will actually compute the factorizations. This can be done by replacing certain, of the nonexecutable Skolem functions, introduced by using first-order quantifiers, with executable versions of those Skolem functions. That is, given a nonexecutable witness function introduced by using first-order quantification, define an executable replacement for that witness function that provably satisfies the same "SUFF" or "NECC" axioms as the original witness function. This plan is carried out in Books 4ab, 4bb, 4cb, and 4db. The theorems are verified via `functional instantiation` of Book 3, but the books contain executable factorization functions.

3.4.2 Unique Factorization

The integers and the Gaussian integers are both Euclidean, and hence, both have unique factorization. How, then, do we explain the following apparently distinct factorizations? In the integers,

$$\begin{aligned} 60 &= 2 \cdot 2 \cdot 3 \cdot 5 \\ &= 2 \cdot -2 \cdot 3 \cdot -5 \end{aligned}$$

In the Gaussian integers,

$$\begin{aligned} 60 &= 2 \cdot 2 \cdot 3 \cdot (2+i) \cdot (2-i) \\ &= 2 \cdot -2 \cdot 3 \cdot (-1+2i) \cdot (1+2i) \end{aligned}$$

The relation "is a factor of" is reflexive and transitive, but often it is not symmetric. The symmetric closure of "is a factor of" is an equivalence relation, "is an associate of," defined by

Definition 5. Element $x \in \mathbb{E}$ is an associate of $y \in \mathbb{E}$ just in case x is a factor of y and y is a factor of x.

Notice that 2 and -2 are associates, as are 5 and -5. Also

$$\begin{aligned} 2+i &= -i \cdot (-1+2i) \\ -1+2i &= i \cdot (2+i) \\ 2-i &= -i \cdot (1+2i) \\ 1+2i &= i \cdot (2-i) \end{aligned}$$

Thus $2+i$ and $-1+2i$ are associates, and so are $2-i$ and $1+2i$.

Uniqueness of factorization is restored by the following convention: Two factorizations, whose products are equal, are equivalent if one can be obtained from the other by replacing any factor with an associate of that factor. Thus, the two factorizations of 60 in the integers are equivalent, as are the two factorizations in the Gaussian integers.

3.4.3 Additional Euclidean Domains

As mentioned in the introduction, polynomials, of a single variable, with rational coefficients, can be uniquely factored into products of irreducible polynomials. This set of polynomials is turned into an Euclidean domain, by taking the `Size` of a polynomial to be the degree of the polynomial.

Let $\mathbb{Z}$ be the integers. For each negative and squarefree $d \in \mathbb{Z}$, let $\mathbb{D}_d$ be the subset of the complex numbers given by

$$\mathbb{D}_d = \begin{cases} \{a + b \cdot \sqrt{d} \mid a, b \in \mathbb{Z}\} & \text{if } d \equiv 2, 3 \bmod 4 \\ \{a + b \cdot \frac{1+\sqrt{d}}{2} \mid a, b \in \mathbb{Z}\} & \text{if } d \equiv 1 \bmod 4 \end{cases}$$

The Gaussian integers is the special case where $d = -1$.

THEOREM 6 ([8, PAGE 697], [7, PAGE 101]). *For negative, squarefree integers d, $\mathbb{D}_d$ is an Euclidean domain iff $d \in \{-1, -2, -3, -7, -11\}$.*

When $\mathbb{D}_d$ is an Euclidean domain, the Euclidean Size function is the square of the complex absolute value, i.e., $\mathtt{Size}(x + y \cdot i) = x^2 + y^2$. When $\mathbb{D}_d$ is not an Euclidean domain, then it is not Euclidean for any choice of Size.

4. NON EUCLIDEAN UNIQUE FACTORS

There are non Euclidean domains with unique factorization.

For example, polynomials, of a single variable, with coefficients from the integers, can each be factored into a product of irreducible polynomials. Moreover, this product is essentially unique except for the order of the irreducible polynomials.

But there is no way to make this set of polynomials, with integer coefficients, into an Euclidean domain: The Division Property, requiring quotient and remainder polynomials, with integer coefficients, fails, as can be seen when the attempt is made to divide $x^2 + 1$ by $5 \cdot x + 2$. If rational coefficients are allowed, then the quotient and remainder polynomials are easily seen to exist.

More examples of non Euclidean domains with unique factorization are given in this theorem.

THEOREM 7 ([8, PAGE 697], [7, PAGE 93]). *For negative, squarefree integers d, $\mathbb{D}_d$ has unique factorization iff $\mathbb{D}_d$ is Euclidean or $d \in \{-19, -43, -67, -163\}$.*

5. FAILURE TO HAVE UNIQUE FACTORS

There are two ways for unique factorization to fail. First, the existence of a finite factorization could fail. Second, finite factorizations could always exist, but not be unique.

Here is an example of the first. Recall, univariate polynomials, over the integers, are of the form $a_n \cdot x^n + \cdots + a_2 \cdot x^2 + a_1 \cdot x + a_0$, where the a_i are integers and a_n is nonzero. The polynomial is monic if $a_n = 1$.

Let $\mathbb{C}$ be the complex numbers and let $\mathbb{P}$ be the set of monic polynomials, of a single variable, with integer coefficients. Let

$$\mathbb{A} = \{\alpha \in \mathbb{C} \mid (\exists p \in \mathbb{P})(p(\alpha) = 0)\}$$

Although not obvious, $\mathbb{A}$ is closed under complex addition, multiplication, unary minus, and the taking of square roots. For example, $\sqrt[2^i]{3}$, for $i = 1, 2, \ldots$, are all in $\mathbb{A}$ because $\sqrt[2^i]{3}$ is a root for the monic polynomial equation $x^{2^i} - 3 = 0$. Since $\alpha = \sqrt{\alpha} \cdot \sqrt{\alpha}$, then $3, \sqrt{3}, \sqrt{\sqrt{3}} = \sqrt[4]{3}, \sqrt{\sqrt[4]{3}} = \sqrt[8]{3}, \ldots$ is an infinite sequence of proper factors in $\mathbb{A}$.

Now for an example of the second. According to Theorem 6, $\mathbb{D}_{-3} = \{a + b \cdot \frac{1+\sqrt{-3}}{2} \mid a, b \in \mathbb{Z}\}$ is an Euclidean domain with unique factorization. The proper subset

$$\mathbb{D} = \{a + b \cdot \sqrt{-3} \mid a, b \in \mathbb{Z}\} \subset \mathbb{D}_{-3}$$

is also closed under complex addition, multiplication, and unary minus. In both $\mathbb{D}$ and $\mathbb{D}_{-3}$,

$$2 \cdot 2 = 4 = (1 + \sqrt{-3}) \cdot (1 - \sqrt{-3}),$$

so 4 appears to have two distinct factorizations. In $\mathbb{D}$, the two apparent factorizations are, in fact, distinct. That is, in $\mathbb{D}$, 2, $(1 + \sqrt{-3})$, and $(1 - \sqrt{-3})$ are all irreducible (or prime), but none is a factor of the other two. Thus, in $\mathbb{D}$, the Key Prime Property fails: 2 is a factor of the product $(1+\sqrt{-3}) \cdot (1-\sqrt{-3})$, but 2 is not a factor of either $(1+\sqrt{-3})$ or $(1 - \sqrt{-3})$.

However, in $\mathbb{D}_{-3}$, the two factorizations are equivalent because 2, $(1 + \sqrt{-3})$, and $(1 - \sqrt{-3})$ are all associates, and hence, equivalent with respect to factorization. Recall that the three being associates means that each of the three are factors of the other two:

$$\begin{aligned}
1 + \sqrt{-3} &= 2 \cdot \frac{1 + \sqrt{-3}}{2} \\
1 + \sqrt{-3} &= (1 - \sqrt{-3}) \cdot \left(-1 + \frac{1 + \sqrt{-3}}{2}\right) \\
1 - \sqrt{-3} &= 2 \cdot \left(1 - \frac{1 + \sqrt{-3}}{2}\right) \\
1 - \sqrt{-3} &= (1 + \sqrt{-3}) \cdot \left[-\left(\frac{1 + \sqrt{-3}}{2}\right)\right] \\
2 &= (1 + \sqrt{-3}) \cdot \left(1 - \frac{1 + \sqrt{-3}}{2}\right) \\
2 &= (1 - \sqrt{-3}) \cdot \frac{1 + \sqrt{-3}}{2}
\end{aligned}$$

Each of these rightmost factors is clearly in $\mathbb{D}_{-3} - \mathbb{D}$.

6. CONCLUSIONS

The algebraic theory of Euclidean domains is described above and explored in several ACL2 books that are described in Appendix A. More ACL2 books are proposed in Appendix B.

Further study of unique factorization domains in ACL2 would include exploring the counterexample domain $\mathbb{A}$ and also $\mathbb{D}_d$ for many d mentioned in Theorems 6 and 7. There are also many other unique factorization domains that are not Euclidean to explore using ACL2.

7. REFERENCES

[1] R. S. Boyer and J S. Moore. *A Computational Logic.* Academic Press, 1979.

[2] R. S. Boyer and J S. Moore. *A Computational Logic Handbook.* Academic Press, second edition, 1998.

[3] I. N. Herstein. *Topics in Algebra.* Blaisdell, 1964.

[4] M. Kaufmann, P. Manolios, and J S. Moore. *Computer-Aided Reasoning: An Approach.* Kluwer Academic Press, 2000.

[5] I. Medina-Bule, J. A. Alonso-Jimènez, and F. Palomo-Lozano. Automatic verification of polynomial rings fundamental properties in ACL2. In M. Kaufmann and J S. Moore, editors, *ACL2 Workshop 2000 Proceedings*, October 30–31, 2000.

[6] I. Medina-Bule, F. Palomo-Lozano, and J. A. Alonso-Jimènez. Implementation in ACL2 of well-founded polynomial orderings. In D. Borrione, M. Kaufmann, and J S. Moore, editors, *3rd Intl.*

Workshop on the ACL2 Theorem Prover and its Applications, pages 66–77, April 8–9, 2002.

[7] I. N. Stewart and D. O. Tall. *Algebraic Number Theory*. Chapman and Hall, second edition, 1987.

[8] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge, second edition, 2003.

APPENDIX

A. ACL2 EUCLIDEAN DOMAIN BOOKS

`prime-fac.lisp`**:** Unique Prime Factorization for Positive Integers. The outline of Bob and J's original proof, as developed in THM, is followed. Their proof is described in the book [1].

`ed1.lisp`**:** Book 1. Multiplicative Identity Existence.
Every Euclidean Domain has a multiplicative identity.

`ed2a.lisp`**:** Book 2a. Multiplicative Size Property.
There is no loss of generality in assuming the Multiplicative Size Property: For all nonzero domain elements x and y,

$$\texttt{Size}(x) \leq \texttt{Size}(x \cdot y).$$

If the original Size function does not satisfy this property, then it (and the original quotient and remainder) can replaced by "new" operations that do satisfy this and the division property.

`ed2b.lisp`**:** Book 2b.
Multiplicative Size Property-CounterExample.
The Integers (with an unusual Size function) are shown to be an Euclidean Domain without the Multiplicative Size Property. Here Quotient is `round` and Remainder is `rnd-rem`, a version of `rem` using round in place of truncate.

`ed3.lisp`**:** Book 3. Algebraic Theory.
Axioms and convenient notation for the theory of an Euclidean Domain. Integral Domain Theory. Divides-p Theory. Associates-p Theory. Unit-p Theory.
Reducible-p and Irreducible-p Theory. Factorization Existence Theory. GCD Theory. Unit-associates-p Theory. Unique Factorization Theory.

`ed4aa.lisp`**:** Book 4aa.
Example: Integers with `floor` and `mod`.
The Integers are shown to be an Euclidean Domain with unique factorization. Here Size is `abs`; Quotient is `floor` and Remainder is `mod`. This version uses quantifiers (`defun-sk`) and is **non-executable**.

`ed4ab.lisp`**:** Book 4ab.
Example: Integers with `floor` and `mod`.
Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 4aa.

`ed4ba.lisp`**:** Book 4ba.
Example: Integers with `truncate` and `rem`.
The Integers are shown to be an Euclidean Domain with unique factorization. Here Size is `abs`; Quotient is `truncate` and Remainder is `rem`. This version uses quantifiers (`defun-sk`) and is **non-executable**.

`ed4bb.lisp`**:** Book 4bb.
Example: Integers with `truncate` and `rem`.
Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 4ba.

`ed4ca.lisp`**:** Book 4ca. Example: Integers with `ceiling`.
The Integers are shown to be an Euclidean Domain with unique factorization. Here Size is `abs`; Quotient is `ceiling` and Remainder is `c-mod`, a version of `mod` using `ceiling` in place of `floor`. This version uses quantifiers (`defun-sk`) and is **non-executable**.

`ed4cb.lisp:` Book 4cb. Example: Integers with `ceiling`.

Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 4ca.

`ed4da.lisp:` Book 4da. Example: Integers with `round`.

The Integers are shown to be an Euclidean Domain with unique factorization. Here Size is `abs`; Quotient is `round` and Remainder is `rnd-rem`, a version of `rem` using `round` in place of `truncate`. This version uses quantifiers (`defun-sk`) and is **non-executable**.

`ed4db.lisp:` Book 4db. Example: Integers with `round`.

Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 4da.

`ed5aa.lisp:` Book 5aa. Example: Gaussian Integers.

The Gaussian Integers, complex numbers with integer real and imaginary parts, are shown to be an Euclidean Domain with unique factorization. Here Size is `sqr-abs`, the square of complex `abs`; Quotient is based on rounding the real and imaginary parts of the complex quotient and Remainder is a version of `rem` using the above rounding in place of `truncate`. This version uses quantifiers (`defun-sk`) and is **non-executable**.

`ed5ba.lisp:` Book 5ba.

Example: Complex numbers of the form $a+b\sqrt{2}i$ where a and b are integers and $i=\sqrt{-1}$.

This version uses ACL2r.

This version uses quantifiers (`defun-sk`) and is **non-executable**.

`ed6a.lisp:` Book 6a.

Example: Polynomials in one variable over a field.

Univariate polynomials with coefficients from an arbitrary field are shown to be an Euclidean Domain with unique factorization. Here Size is the degree of a polynomial; Quotient and Remainder are defined as expected for polynomials. This version uses quantifiers (`defun-sk`) and is **non-executable**.

ACL2 Univariate Polynomial Books are based on more general ACL2 Polynomial Books [6, 5] by I. Medina-Bule, F. Palomo-Lozano, and J. A. Alonso-Jimènez.

B. PROPOSED EUCLIDEAN DOMAIN BOOKS

`ed5ab.lisp:` Book 5ab. Example: Gaussian Integers.

Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 5aa.

UNDER CONSTRUCTION

`ed5bb.lisp:` Book 5bb.

Example: Complex numbers of the form $a+b\sqrt{2}i$ where a and b are integers and $i=\sqrt{-1}$.

Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 5ba.

This version uses ACL2r.

UNDER CONSTRUCTION

`ed5ca.lisp:` Book 5ca.

CounterExample: Complex numbers of the form $a+b\sqrt{3}i$ where a and b are integers and $i=\sqrt{-1}$.

Unique factorization fails, so these numbers do not form an Euclidean domain.

Eg. $2\cdot 2 = 4 = (1+\sqrt{-3})\cdot(1-\sqrt{-3})$

This version uses ACL2r.

UNDER CONSTRUCTION

`ed5cb.lisp:` Book 5cb. Example: Eisensteinean Integers.

Complex numbers of the form $a+b\cdot\frac{1+\sqrt{3}i}{2}$ where a and b are integers and $i=\sqrt{-1}$.

This version uses quantifiers (defun-sk) and is non-executable.

This version uses ACL2r.

UNDER CONSTRUCTION

`ed5cc.lisp:` Book 5cc. Example: Eisensteinean Integers.

Complex numbers of the form $a+b\cdot\frac{1+\sqrt{3}i}{2}$ where a and b are integers and $i=\sqrt{-1}$.

Executable factorization algorithms are obtained by using computable Skolem functions in place of quantifiers (`defun-sk`) used above in Book 5cb.

This version uses ACL2r.

UNDER CONSTRUCTION