Theory Extension in ACL2(r)

R. Gamboa (ruben@cs.uwyo.edu) and J. Cowles (cowles@uwyo.edu)
Department of Computer Science
University of Wyoming
Laramie, Wyoming, USA

Abstract. ACL2(r) is a modified version of the theorem prover ACL2 that adds support for the irrational numbers using non-standard analysis. It has been used to prove basic theorems of analysis, as well as the correctness of the implementation of transcendental functions in hardware. This paper presents the logical foundations of ACL2(r). These foundations are also used to justify significant enhancements to ACL2(r).

1. Introduction

ACL2(r) is a variant of the theorem prover ACL2 with support for the real and complex numbers via non-standard analysis. Non-standard analysis, introduced by Abraham Robinson [18] in the 1960s, gave rigorous foundations to the sort of reasoning about infinitesimal quantities that was used by Leibniz back when he co-invented the calculus and is still used today by engineers and scientists when applying calculus. A feature of many arguments using non-standard analysis is the use of mathematical induction and recursion in place of standard limit and compactness arguments, and this feature makes non-standard analysis well-suited for reasoning in ACL2, a first-order, inductive theorem prover.

ACL2(r) was introduced in [5], and it has been used to verify basic properties of the reals, such as the Fundamental Theorem of Calculus [8], Taylor's Theorem [6] and results in interval arithmetic [14], as well as the correctness of hardware designs that use transcendental functions [19].

The focus of this paper is on the *logical foundations* of ACL2(r). In introducing ACL2(r), [5] explains these foundations by invoking Internal Set Theory, an axiomatic treatment of non-standard analysis pioneered by Nelson [16] and developed more broadly later as Hrbaček Set Theory [7]. But the treatment in [5] assumes a *fixed* ACL2(r) theory. Given such a theory, [5] explains how we can interpret it using Internal Set Theory, and how this justifies the axioms and inference methods of ACL2(r). But this ignores the dynamic nature of ACL2(r) theories, i.e., it gives no explanation of the way in which an ACL2(r) theory is extended when new function symbols are introduced via definitional events.

The treatment of definitional events in ACL2 is presented rigorously in [13]. Definitional events raise many subtle issues in ACL2, in part because (a) ACL2 has several structuring mechanisms, such as local definitions and theory inclusion, that are indispensible when dealing with large theories but complicate the notion of an ACL2 theory, and (b) ACL2, while strictly first-order, supports via constrained functions and functional instantiation a style of reasoning reminiscent of higher-order reasoning.

This paper shows the logical foundations of ACL2(r) in a manner analogous to the way [13] presents the foundations of ACL2, by explaining how a logical theory is modified by definitional events. That is the first contribution of this paper.

The second contribution of this paper is that it gives a logical justification for important enhancements to ACL2(r). For example, ACL2(r) as described in [5] does not support the introduction of recursive functions that use non-classical functions, e.g., one that recognizes infinitesimals. The logical foundations presented here support this, as well as non-classical analogues of the other definitional principles of ACL2, namely constrained and Skolem functions. These enhancement are being implemented in a new version of ACL2(r).

This paper is organized as follows. Section 2 gives a brief introduction to ACL2(r), while section 3 introduces most of the machinery needed from first-order logic to build the foundations of ACL2(r). The following sections detail the treatment of the definitional axioms in ACL2(r): explicit definitions in section 4, implicit definition of classical functions in section 5, constrained definitions in 6, and Skolem definitions in section 7. The paper concludes in section 8 with an overview of the enhancements to ACL2(r) enabled by this paper.

2. Brief Introduction to Non-Standard Analysis and ACL2(r)

2.1. Non-Standard Analysis

Non-standard analysis is an extension to ordinary set theory that gives rigorous semantics to the notions of "infinitely small" and "infinitely large." It was originally presented in [18], where it is shown via a careful model-theoretic construction that (a) these informal notions can be placed on a solid, logical footing, and (b) the resulting structure provides new ways of proving theorems of ordinary mathematics, without producing new theorems of ordinary mathematics.

ACL2(r) follows the axiomatic approach to non-standard analysis developed by Nelson as Internal Set Theory (IST) [16]. Internal Set Theory is an extension of ZF Set Theory. It extends the language of set theory with a new undefined predicate called *standard*. IST makes an important distinction between *classical* formulas, those that are in the language of ZF Set Theory, and *non-classical* formulas, those that depend on the predicate *standard*.

Classical formulas follow all the familiar properties of ZF Set Theory, but non-classical formulas require special treatment. For example, the specification axiom does not hold for non-classical formulas. Given a set S and a first-order formula $P(x)$ with free variable x, the specification axiom of ZF set theory guarantees the existence of a set $S' = \{x \in S \mid P(x)\}$ that contains precisely the elements of S that satisfy the property $P(x)$. However, in IST this is only true if the formula $P(x)$ is classical. This means, for example, that it is impossible in IST to build the set of all standard natural numbers. It is worth emphasizing that if the formula $P(x)$ is classical then the specification axiom *does* hold in IST. More generally, any set that can be constructed with ZF Set Theory can also be constructed in IST.

The restriction of the specification axiom to classical formulas has some important consequences, particularly important in the context of ACL2. For example, the principle of induction does not apply to non-classical formulas. Suppose $P(x)$ is classical, and assume for simplicity that we are proving $P(x)$ using induction on the natural numbers. That is, given that $P(0)$ is true and that $P(n) \Rightarrow P(n+1)$ is true, we are allowed to conclude that $P(x)$ is true for all x. However, if $P(x)$ is not classical we can only conclude that $P(x)$ is true for all standard x[1].

2.1.1. *The Standardization Principle*

IST does support a weak version of specification called the Standardization Principle. Given the set S and arbitrary property $P(x)$ as above, this principle guarantees the existence of a unique standard set $S' = {}^\circ\{x \in S \mid P(x)\}$ whose standard elements are precisely those standard elements of S that satisfy $P(x)$. Notice that no guarantees are made about the non-standard elements of S'. For example, this principle guarantees that there is a set R' such that $R' = {}^\circ\{x \in R \mid \mathit{standard}(x)\}$, where R is the (classical, hence standard) set of real numbers. The standardization principle guarantees that R' is standard, that it contains all standard members of R, and it is the unique stan-

[1] An alternative view is that the standard natural numbers correspond to the traditional set of natural numbers, and that the non-standard numbers are in fact *new* natural numbers beyond the original number line. In this view, induction on arbitrary formulas works only on the original natural numbers.

dard set that does so. Since R itself is standard and R contains all standard members of R, the principle guarantees that R' is equal to R. The reader can similarly verify that ${}^{\circ}\{x \in R \mid \neg \mathit{standard}(x)\}$ is the empty set.

When $P(x)$ is such that all x that satisfy it must be elements of some set U, the standardization axiom provides a useful construction called a shadow set. Applying the specification axiom to the class $S = \{x \mid P(x)\}$, we can find a unique standard set ${}^{\circ}S = {}^{\circ}\{x \in U \mid P(x)\}$ such that $\mathit{standard}(x) \Rightarrow (x \in {}^{\circ}S \Leftrightarrow P(x))$. The shadow set is extremely useful. Applied to the graph of a non-classical function $f(x)$, it can be used to demonstrate the existence of a unique classical function ${}^{\circ}f(x)$ such that $\mathit{standard}(x) \Rightarrow f(x) = {}^{\circ}f(x)$.

2.1.2. *The Transfer Principle*

IST derives much of its power from the Transfer Principle, which states that if $P(x)$ is a classical property such that $\mathit{standard}(x) \Rightarrow P(x)$ is true, then $P(x)$ is also true for arbitrary x. This powerful principle implies that any object which can be uniquely defined with a classical property — e.g., 0, 1, π, $\sqrt{2}$ — must be standard. Note that this principle does not apply to non-classical formulas.

2.1.3. *The Idealization Principle*

IST also includes the Idealization Principle. Its relevance to ACL2(r) is limited to ensuring that there is at least one non-standard element of each type. In particular, it guarantees the existence of a non-standard natural number, which is necessarily larger than all the standard naturals.

This description of IST is sufficient to build the foundations of ACL2(r), although it gives little insight into the connection with non-standard analysis or for that matter real analysis of any kind. Intuitively the connection stems from associating the "infinitesimaly small" numbers (which can be called ϵ suggestively) with those numbers that are smaller in magnitude than any non-zero standard real. We will not explore this connection in this paper; the interested reader is referred to [17, 4, 15] for details.

2.2. ACL2 and ACL2(r)

ACL2 denotes both a logic [12, 13] and a theorem prover for this logic [10]. The logic, originally based on the Boyer-Moore logic [2, 3], uses the syntax of Common Lisp. Its inference rules include propositional calculus, equality, instantiation, and well-founded induction up to ϵ_0. ACL2 also supports the introduction of new functions via a variety of mechanisms. The event `defun` in ACL2 allows the introduction of possibly mutually recursive functions via an explicit definition. Recursive functions introduced this way are subject to a well-foundedness restriction: i.e., ACL2 will not accept a recursive definition unless it can prove that the associated computation terminates on all possible inputs. The event `encapsulate` in ACL2 allows the introduction of constrained functions. This is equivalent to the common mathematical practice of giving a name to an arbitrary function with some properties, e.g., "let f be a continuous function. . . ." This allows ACL2 to reason about generic functions, e.g., the class of continuous functions, even though it is strictly first-order [1, 13]. Associated with constrained functions is a derived rule of inference called "functional instantiation," which allows ACL2 to conclude that a given statement $Th(g)$ that mentions the function g is true, provided that (a) the formula $Th(f)$ which results by substituting the function g by the constrained function f in $Th(g)$ is true, and (b) the function g satisfies all the constraints of f. Finally, ACL2 supports the introduction of new function symbols via the event `defchoose`. This event allows the introduction of universal and existential quantifiers as Skolem functions. ACL2 provides an initial boot-strap theory, called the Ground Zero Theory, which axiomatizes many Lisp functions.

ACL2(r) modifies ACL2 by introducing some functions to its Ground Zero Theory, definitional principles, and derived inference rules, all of which correspond to their counterparts in IST. For example, ACL2(r) introduces the built-in function `standard` which corresponds to the predicate *standard* of IST. ACL2(r) also classifies function symbols as either classical or non-classical: Functions which depend on `standard` are non-classical, while all other functions (including all functions that are also in ACL2) are classical. As with IST, ACL2(r) restricts the use of induction on formulas that are non-classical. ACL2(r) includes a new definitional principle via the event `defun-std`. This principle corresponds to using the shadow of a non-classical function in IST to construct a new classical function. Finally, ACL2(r) adds the derived inference rule that corresponds to IST's Transfer Principle with the event `defthm-std`. This event allows users to submit to ACL2(r) classical formulas that they want to be proved using transfer.

3. Preliminaries

We are concerned in this paper with first-order *theories*: sets of first-order formulas that are closed under logical consequence. In the context of reasoning about ACL2 or ACL2(r), it is sufficient to restrict ourselves to first-order theories with equality and no other predicate symbols, and we implicitly assume this restriction for the remainder of this paper. We assume that the reader is familiar with the following basic notions: The *language* of a first-order theory is the set of function symbols occurring in its formulas. It is common to assume, as we do in this paper, that the language of a theory is countable. A theory T_1 *extends* a theory T_2 if every theorem in T_2 is also a theorem in T_1. Moreover, T_1 *conservatively extends* T_2 if every theorem of T_1 in the language of T_2 is also a theorem of T_2.

Since we use the classical notion of logical consequence as our only inference scheme, the theories we consider must include axioms describing any other inference rules, such as induction or transfer. Now we consider axiom schemas that characterize the derived inference rules of ACL2(r).

The case for induction is straightforward. ACL2(r) contains the binary function symbol $\prec$, which (intuitively) represents a well-founded relation on the ACL2(r) universe[2]. The induction axiom schema for classical formulas ϕ is given in [13]. We extend this axiom schema here to include non-classical formulas as well. Recall that in the context of Internal Set Theory induction on non-classical formulas only guarantees that the formula is true for standard values.

Definition. Let ϕ be a formula, let x be a free variable (or vector of free variables) in ϕ, and let y be a variable (not occurring in ϕ. Then the *induction axiom for ϕ with respect to x* is given by

$$(\forall x)(((\forall y \prec x)\phi[x := y]) \Rightarrow \phi) \Rightarrow (\forall x)\phi$$

if ϕ is classical, and

$$(\forall x)(((\forall y \prec x)\phi[x := y]) \Rightarrow \phi) \Rightarrow (\forall x)(standard(x) \Rightarrow \phi)$$

otherwise. A first-order theory T is said to be *complete with respect to induction* if it includes every induction axiom in the language of T. □

The transfer principle is also simple: We need to add a transfer axiom for every possible classical formula ϕ. Notice that we only add these axioms for classical formulas, since the transfer principle can only be used in these cases.

Definition. Let ϕ be a classical formula with free variables $x_1, \ldots, x_n$ and no other free variables. The *transfer axiom for ϕ* is as follows:

$$(\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} standard(x_i)) \Rightarrow \phi) \Rightarrow (\forall x_1 \ldots x_n)\phi$$

[2] The well-founded relation $\prec$ is *not* a well-ordering, because it is not total. Readers familiar with ACL2 may think of $x \prec y$ as `(o-< (acl2-count x) (acl2-count y))`. A similar definition for the corresponding ACL2 ordering is given in [13].

A first-order theory T is said to be *complete with respect to transfer* if it includes every transfer axiom in the language of T. □

We now justify an inference rule needed to justify `defun-std`, the definitional principle of ACL2(r) that allows under certain conditions the introduction of classical functions with a non-classical definition. The formal justification given in [5] for `defun-std` uses the idea of a shadow set of a function graph to define the unique classical function that agrees with the graph on standard arguments. But the problem is that this argument, with its appeal to shadow sets and graphs of functions, is in the language of set theory, a far richer setting than the standard language of ACL2. To avoid this difficulty, we introduce the classical function symbol f_τ for each possible term τ a priori. These function symbols are disjoint from the set of function symbols that can be introduced by an ACL2(r) user.

Definition. Let L be a language. *L contains all its term functions* if for any term τ in the language of L, L contains a function symbol f_τ with arity n, where n is the number of free variables in τ. □

Definition. Let τ be a term with free variables $x_1, \ldots, x_n$ and no other free variables. The *standardization axiom for τ* is as follows:

$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} standard(x_i)) \Rightarrow standard(\tau))) \Rightarrow$$
$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} standard(x_i)) \Rightarrow f_\tau(x_1, \ldots, x_n) = \tau))$$

We introduce the shorthand notation $SA(\tau, f_\tau)$ to represent this axiom. The function symbols f_τ are said to be *non-visible.* All other function symbols are *visible.* A term or formula is said to be *visible* if it mentions only visible function symbols; otherwise, it is said to be *non-visible.* A first-order theory T is said to be *complete with respect to standardization* if it includes all the standardization axioms in the language of T. □

Note: The preceding definition implies that a theory can only be complete with respect to standardization if its language contains all its term functions. □

It is possible to start with a countable theory T that does not contain any standardization axioms and derive a theory T' that extends T and is complete with respect to standardization. The process is as follows. Let Φ_0 be an enumeration of the formulas $\phi_{0,1}$, $\phi_{0,2}$, ... of T, and let f_1, f_2, ... be function symbols not in the language of T. Define the enumerations Φ_i of formulas $\phi_{i,1}$, $\phi_{i,2}$, ... as follows. Let L_0 be the language of T and L_i be the language of L_{i-1} with the addition of f_i. Then Φ_i is an enumeration of the formulas $\phi_{i,1}$, $\phi_{i,2}$, $\ldots$ in the language of L_i that are not in the language of L_{i-1}. We are now ready to introduce the standardization axioms as follows:

$$\begin{array}{ll} SA(\phi_{i,j}, f_{(j-1)^2+i+1}) & \text{for } i < j \\ SA(\phi_{i,j}, f_{i^2+2i+2-j}), & \text{for } i \geq j \end{array}$$

The indices of the f_k are chosen to correspond to a left-right, bottom-up enumeration of the $\phi_{i,j}$, just as in a typical proof of the denumerability of a denumerable union of denumerable sets. Notice that the $\phi_{i,j}$ enumerate all the formulas in the language of T and all the f_k. Therefore, the axioms resulting from this enumeration can be used to complete the theory T with respect to standardization as claimed.

Observe that for $j \geq 1$, $(j-1)^2 + i + 1 > i$. Similarly, when $i \geq j \geq 1$, $i^2 + 2i + 2 - j > i$. Therefore, whenever $SA(\phi_{i,j}, \tau_k)$ is an axiom generated by this construction, we have that $i < k$, which guarantees that f_k is not in the language L_i. This means that we can use the standardization axioms to generate a partial ordering on the formulas of the theory; we refer to this ordering as the ordering implied by standardization. First let f_τ be the f_k such that $SA(\tau, f_k)$ is one of the standardization axioms. We say that $\tau_1 \ll \tau_2$ if f_{τ_1} appears in τ_2, and the ordering implied by standardization is the transitive closure of $\ll$. Note that this implied ordering is a well-founded partial order of the formulas. The process outlined above to generate the standardization axioms for a given theory generates a valid partial ordering $\ll$, since whenever $SA(\tau, f_k)$ is a standardization axiom we know that τ is in the language of L_l for some $l < k$. For arbitrary theories complete under standardization — i.e., when the completion is not formed by the process outlined above — it is possible that the $\ll$ relation involved is in fact not a partial order. It may have loops, for example.

Convention. For the remainder of this paper, when we say that a theory T is complete under standardization, we are also asserting that the ordering implied by the standardization axioms of T is a valid partial ordering. $\square$

Definition. A first-order theory T is said to be *r-complete* if it is complete with respect to induction, transfer, and standardization. The *r-completion* of the theory T is the theory resulting from extending T by the induction, transfer, and standardization axioms in the language of T. $\square$

The basic story that we lay out in this paper is as follows. We start with an r-complete theory GZ of ACL2(r). Then we show that the axiomatic events of ACL2(r) — `defun`, `defun-std`, `encapsulate`, and `defchoose` — conservatively extend an r-complete theory T into an r-complete theory T'. We will develop these ideas in the following sections. The important point of the story is that every ACL2(r) history — i.e., every sequence of definitions, theorems, and other ACL2(r) events — corresponds in our theoretical setting to the r-completion of the definitional events in the history.

Moreover, in an r-complete theory the derived inference rules of ACL2(r) — `defthm-std`, the non-standard principle of induction — are simply first-order consequences of the axioms. Users suggest theorems to ACL2(r) with the events `defthm` and `defthm-std`. The event `defthm` directly invokes the theorem prover of ACL2(r), which will use (classical and non-classical) induction schemes to prove formulas automatically. The induction axioms introduced in this section correspond directly to the induction schemes used by the theorem prover. ACL2(r) does not use transfer directly, but it offers the event `defthm-std` that allows users to suggest theorems to be proved by transfer. That is, when the user suggests via `defthm-std` that the classical formula ϕ with free variables $x_1, x_2, \ldots, x_n$, is a theorem, ACL2(r) verifies this by checking that $(\forall x_1 \ldots x_n)(\bigwedge_{i=1}^{n} standard(x_i) \Rightarrow \phi$ is a theorem instead. If this weaker theorem is true, ACL2(r) accepts ϕ as a theorem as well. The formal justification for `defthm-std` follows from the definition of r-complete theories.

Why is the r-complete Ground Zero theory of ACL2(r) consistent? The answer comes from [16] and [5]. In [16] it is shown that Internal Set Theory is a conservative extension of ZF set theory. Moreover, in IST the predicate *standard* is given an interpretation. IST restricts the ways in which non-classical terms (i.e., those defined in terms of *standard*) can be manipulated. For instance, it restricts the use of induction for non-classical formulas. But IST provides inference rules that justify the use of non-classical formulas, such as idealization and standardization. As shown in [5] these have direct counterparts in the r-complete theory GZ.

4. Explicit Definitions with defun

The story of `defun` is told definitively in [13], where it is shown that `defun` events result in conservative extensions of an ACL2 theory. In this section, we modify the argument in [13] to ACL2(r) theories. We will show three main results. First, definitional axioms introduced with `defun` result in a conservative extension of the theory. Second, if the original theory is r-complete, so is the extension. Third, the traditional notion in Nqthm and ACL2 of measure admissibility implies the notion of interpreter admissibility which will be used in this section to show the first two results.

4.1. Conservativity of Extension

Definition. A *definitional axiom* D over a theory T is a finite conjunction of equations of the following form

$$f(x_1, \ldots, x_n) = \mathit{term}$$

where the function symbols f in the left-hand side of this axioms are distinct function symbols disjoint from the function symbols in T, *term* is a term in the union of the language of T with the set of left-hand side function symbols of D, the variables x_i are distinct, and the only variables that are free in *term* are the x_i. If any term in the right-hand side of the equations uses a non-classical symbol, the function symbols f in the left-hand side are also taken to be non-classical. Otherwise, these function symbols are considered classical. □

Remark. Ignoring the distinction of symbols into classical and non-classical, this definition is equivalent to the definition given in [13]. Therefore any function that can be introduced into an ACL2 session using a definitional axiom of ACL2 can also be introduced into ACL2(r) using the same definitional axiom. □

For the remainder of this section fix an r-complete theory T and a definitional axiom D over T. Further, assume that T is an extension of the ACL2(r) Ground Zero Theory[3]. Let F be the set of function symbols introduced by D, i.e., those in the left-hand side of equations of D.

Following [13] we define the canonical interpreter for D as follows: Suppose D contains the equation

$$f(x_1, \ldots, x_n) = \mathit{term}$$

and let d be a variable not in *term*. Replace this equation with the following

$$f'(d, x_1, \ldots, x_n) = \mathit{if}\ \ zp(d)\ \mathit{then}\ NIL\ \mathit{else}\ \mathit{term}_{d-1,1}$$

where $zp(d)$ is is false precisely when d is a positive integer. The formula $u_{d,b}$ corresponds to a bounded execution of the term u. The variable d

[3] This is not strictly necessary. What we need are axioms relating to car, cdr, etc.

limits the execution by preventing more than d uses of the definitions in F. The b in $u_{d,b}$ is used to differentiate the occurrence of subterms that are in the top-level of if-terms. Originally, b is set to 1, and the recursive definition below sets it to 0 when a subterm is inside a non-if function. The formal definition of $u_{d,b}$ is as follows:

– if u is a constant or variable, then $u_{d,b} = cons(u, NIL)$

– else if u is $\mathit{if}\ t_0\ \mathit{then}\ t_1\ \mathit{else}\ t_2$, $b = 1$, and $(t_0)_{d,0} = NIL$, then $u_{d,b} = NIL$

– else if u is $\mathit{if}\ t_0\ \mathit{then}\ t_1\ \mathit{else}\ t_2$, $b = 1$, and $car((t_0)_{d,0}) \neq NIL$, then $u_{d,b} = (t_1)_{d,1}$

– else if u is $\mathit{if}\ t_0\ \mathit{then}\ t_1\ \mathit{else}\ t_2$ and $b = 1$, then $u_{d,b} = (t_2)_{d,1}$

– else if u is $f(t_1, \ldots, t_n)$ and at least one of $(t_i)_{d,0} = NIL$, then $u_{d,b} = NIL$

– else if u is $f(t_1, \ldots, t_n)$ where $f \notin F$ (f is possibly *if*), then $u_{d,b} = cons(f(car((t_1)_{d,0}), \ldots, car((t_n)_{d,0})), NIL)$

– else u must be $f(t_1, \ldots, t_n)$ for some $f \in F$, and $u_{d,b} = f'(d, car((t_1)_{d,0}), \ldots, car((t_n)_{d,0}))$

The symbols f' can be thought of as new function symbols, not in the language of T or F. However, we can define them instead as expressions in the language of T as follows. It is clear that the functions f' terminate: The variable d serves to limit the number of times a term involving f' is "opened" and all other branches through the definition of $u_{d,b}$ dive into a subterm of u. A computation of f' can be thought of as a sequence of equalities, e.g., the sequence produced by expanding the leftmost term into its definition: $f'(d, t_1, \ldots, t_n) = u$ if and only if there is a sequence of terms starting with $f'(d, t_1, \ldots, t_n)$ and ending with u such that each element (other than the first) of the sequence follows from the previous one by the expansion of its leftmost term. This condition can be stated as a first-order formula in the language of T; in other words, the f' are first-order definable in T. From now on, when we say f' what we mean is this first-order definition in the language of T, so that in fact the f' are *not* new function symbols.

We are interested only in definitions D that correspond to computations that are guaranteed to terminate on all possible inputs. When all the definitions in D are classical, we insist that for each formula

$$f(x_1, \ldots, x_n) = term$$

in D, it is a theorem of T that

$$(\forall x_1 \dots x_n)(\exists d)(f'(d, x_1, \dots, x_n) \neq NIL)$$

When at least one definition in D is non-classical, we require a stronger condition, namely that the value of d that shows f' terminates is standard:

$$(\forall x_1 \dots x_n)(\exists d)(standard(d) \wedge f'(d, x_1, \dots, x_n) \neq NIL)$$

Definitions that meet these criteria are called *interpreter admissible*.

Definition. Let T be an r-complete theory and D an interpreter admissible definitional axiom over T. Then T^D is the extension of T by the universal completions of the following equations, one for each f defined in D:

$$f(x_1, \dots, x_n) = \begin{cases} car(f'(d, x_1, \dots, x_n)), & \\ & \text{where } d \text{ is the least natural} \\ & \text{(additionally required to be} \\ & \text{standard if } D \text{ is non-classical)} \\ & \text{such that} \\ & car(f'(d, x_1, \dots, x_n)) \neq NIL \\ & \\ NIL, & \text{if there is no such } d \end{cases}$$

□

Observation. The theory T^D is a conservative extension of T. This follows because the new functions $f \in F$ are explicitly defined using only terms in the language of T (i.e., without recursion). Recall, in particular, that the f' are first-order definable in T. □

In the next section, we define the theory T_D as the r-completion of T^D, and it is T_D that serves to explain what happens when a definitional axiom is introduced into ACL2(r). While it is clear that T^D is a conservative extension of T, it remains to be seen that T_D is also a conservative extension of T. To demonstrate this, we prove that the induction and transfer axioms in the language of T^D are first-order derivable in T^D from the induction and transfer axioms in the language of T, and that the standardization axioms in the language of T^D are first-order derivable with explicit definitions from the standardization axioms in the language of T. Consequently, T_D is a conservative extension of T^D.

Lemma (Term Equivalence). Let T_2 be the extension of T_1 formed by explicit definitions of new function symbols in F. Then for every term τ in the language of T_2, there is a term τ' in the language of T_1 with the same free variables as τ such that $\tau = \tau'$ is a theorem of T_2. Moreover, τ' is classical if τ is classical.

Proof. We proceed by induction on the terms τ of T_2. If τ is a variable or constant symbol, then τ is already in the language of T_1 (since we are extending the language of T_1 only by introducing new function symbols), and $\tau = \tau$ is certainly a theorem of T_2. Otherwise, τ is of the form $f(\tau_1, \ldots, \tau_n)$ for some terms τ_i. Using the induction hypotheses, there are terms τ_1', …, τ_n' in the language of T_1 such that $\tau_i = \tau_i'$ is a theorem of T_2 for each i, and τ_i' has the same free variables as τ_i. Moreover, if τ is classical, each of the τ_i are classical, and so are each of the τ_i'. If $f \notin F$, then f must be in the language of T_1, in which case letting $\tau' = f(\tau_1', \ldots, \tau_n')$ it follows that $\tau = \tau'$ is a theorem of T_2, τ' has the same free variables as τ, and clearly τ' is classical if τ is classical. Otherwise, f is one of the functions explicitly defined to extend T_1. I.e., there is a term τ_f in the language of T_2 such that $(\forall x_1 \ldots x_n)(f(x_1, \ldots, x_n) = \tau_f)$ is an axiom of T_2, and moreover the x_i are the only variables free in τ_f. But then letting $\tau' = \tau_f[x_i := \tau_i']$, we can conclude in T_2 that $\tau = \tau'$, and τ' is in the language of T_1. Since τ' replaces all free variables in τ_f, it follows that τ' has no more free variables than τ, and it is easy to extend τ' so that it has exactly the same free variables as τ. Moreover, if τ is classical f is a classical function, which means that τ_f is classical. So τ' is also classical. □

Lemma (Formula Equivalence). Let T_2 be the extension of T_1 formed by explicit definitions of new function symbols in F. Then for every formula ϕ in the language of T_2, there is a formula ϕ' in the language of T_1 such that $\phi \Leftrightarrow \phi'$ is a theorem of T_2 and ϕ' has the same free variables as ϕ. Moreover, ϕ' is classical if ϕ is classical.

Proof. This is an simple extension of the Term Equivalence Lemma, proved using induction on the logical structure of ϕ. □

Theorem (Induction Completeness Preservation). Let T_1 be a theory that is complete with respect to induction and let T_2 be the extension of T_1 formed by explicit definitions of new function symbols in F. Then T_2 is complete with respect to induction.

Proof. We prove this by showing that each induction axiom over the language of T_2 is a theorem of T_2. Let ϕ be an induction axiom over the language of T_2. Recall that there are two types of induction axioms, depending on whether the underlying formula is classical or not. We consider each case separately. Suppose ϕ takes the following form, where ψ is classical:

$$(\forall x)(((\forall y \prec x)\psi[x := y]) \Rightarrow \psi) \Rightarrow (\forall x)\psi$$

Using the Formula Equivalence Lemma, we can find ψ' in the language of T_1 such that $\psi \Leftrightarrow \psi'$ is a theorem of T_2 and ψ' is also classical. Therefore, the following is an induction axiom in T_1 and hence a theorem of T_2:

$$(\forall x)(((\forall y \prec x)\psi'[x := y]) \Rightarrow \psi') \Rightarrow (\forall x)\psi'$$

Since $\psi \Leftrightarrow \psi'$ is also a theorem of T_2, it trivially follows that

$$(\forall x)(((\forall y \prec x)\psi[x := y]) \Rightarrow \psi) \Rightarrow (\forall x)\psi$$

is a theorem of T_2.

A similar argument suffices to show that ϕ is a theorem of T_2 when ϕ is of the form

$$(\forall x)(((\forall y \prec x)\psi[x := y]) \Rightarrow \psi) \Rightarrow (\forall x)(\mathit{standard}(x) \Rightarrow \psi)$$

for a non-classical ψ. So we conclude that T_2 contains all induction axioms over its language; i.e., it is complete with respect to induction. □

Theorem (Transfer Completeness Preservation). Let T_1 be a theory that is complete with respect to transfer and let T_2 be the extension of T_1 formed by explicit definitions of new function symbols in F. Then T_2 is complete with respect to transfer.

Proof. Let ϕ be a transfer axiom over the language of T_2. Then ϕ has the form

$$(\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \psi) \Rightarrow (\forall x_1 \ldots x_n)\psi$$

for some classical formula ψ in the language of T_2. There is a formula ψ' in the language of T_1 such that $\psi \Leftrightarrow \psi'$ is a theorem of T_2, and moreover ψ' is classical. That means that the transfer axiom for ψ' is a theorem of T_1:

$$(\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \psi') \Rightarrow (\forall x_1 \ldots x_n)\psi'$$

But then ϕ is provable in T_2. □

Matters are not as straightforward in the case of standardization. The problem is that the standardization axiom requires a classical function symbol f_τ for each term τ in the language of T_2. But since the language of T_2 extends the language of T_1 (by the new function symbols in F), we now have formulas τ (using one of the new function symbols) for which there is no f_τ. What the next lemma shows is that it is always possible to extend the theory T_2 to contain the needed function symbols.

Theorem (Standardization Completeness Preservation). Let T_1 be a theory that is complete with respect to standardization and let T_2 be the extension of T_1 formed by explicit definitions of new function symbols in F. Then there is a conservative extension ${}^*T_2$ of T_2 (and hence of T_1) that is complete with respect to standardization. We call ${}^*T_2$ the *standardization completion of* T_2 with respect to T_1.

Proof. In section 3 we introduced a construction that starts with a theory T and produces an extension T' that is complete with respect to standardization. We apply this process to T_2, but we adapt it so that the new function symbols introduced are defined explicitly in terms of functions in T_1. This ensures that the resulting theory is a conservative extension that is also complete with respect to induction and transfer.

The modification is as follows. Suppose we are considering a term τ in the language of T_2 (or an extension of it) that is not in the language of T_1. By the Term Equivalence Lemma, there is a term τ' in the language of T_1 such that T_2 proves $\tau = \tau'$. Let $x_1, \ldots, x_n$ be all the free variables in τ and define $f_\tau(x_1, \ldots, x_n) = f_{\tau'}(x_1, \ldots, x_n)$.

Since T_2 is an extension of T_1 and T_1 is complete with respect to standardization, T_2 contains all the standardization axioms for terms in the language of T_1. In particular, it contains the standardization axiom for τ':

$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \mathit{standard}(\tau'))) \Rightarrow$$
$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f_{\tau'}(x_1, \ldots, x_n) = \tau'))$$

It immediately follows that the following is a theorem of the extension of T_2 introducing f_τ:

$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \mathit{standard}(\tau))) \Rightarrow$$
$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f_\tau(x_1, \ldots, x_n) = \tau))$$

Moreover f_τ is classical, since it is defined in terms of a classical function, i.e., $f_{\tau'}$. So the standardization axiom for τ is a theorem of this extension of T_2. □

Definition. Let T be an r-complete theory and D an interpreter admissible definitional axiom over T. The *r-complete extension of* T *by* D, denoted by T_D, is the theory formed as follows: (1) Extend T by D, (2) take the standardization completion of the resulting theory, (3) take the inductive completion of the resulting theory, and finally (4) take the transfer completion of the resulting theory. □

When a definitional axiom D is introduced into the ACL2(r) theory T, the ACL2(r) theory of the session is extended to T', the complete extension of T by D. As in [13], T' is a subtheory of the (standardization, inductive, and transfer) completion of T_D. Moreover, it is clear that the ACL2(r) event of `defthm-std` is justified in T', since T' is complete with respect to transfer. Similarly, the classical and non-classical induction principles of ACL2(r) follow from the fact that T' is complete with respect to induction. In section 5, we explore the justification for `defun-std`, and in section 7 we do so for `defchoose`.

4.3. Measure Admissibility

In the previous sections we defined interpreter admissibility and showed how functions that are defined explicitly with formulas satisfying interpreter admissibility can be added to an r-complete ACL2(r) session resulting in an r-complete conservative extension. But ACL2(r) does not check that function definitions are interpreter admissible. Instead it uses the notion of measure admissibility first introduced in Nqthm to determine if a function definition is acceptable. The main goal of this section is to show that any definition that is measure admissible is also interpreter admissible. Before presenting the main result, we need to establish some basic properties of interpreter admissibility. The argument presented in this section was first developed in [13] in the context of ACL2. In this section we adapt the argument of [13] to allow for non-classical functions.

Lemma (Interpreter Monotonicity). Let D be a definitional axiom over a theory T, and let u be a term in the language of $T \cup \{D\}$. Let d and d' be variables not occurring in u. Then the following is a theorem of T:

$$\begin{aligned}&standard(d) \wedge natp(d) \wedge natp(d') \wedge d \leq d' \wedge u_{d,b} \neq NIL \Rightarrow \\ &\quad u_{d,b} = u_{d',b} \wedge u_{d',b} = cons(car(u_{d,b}), NIL)\end{aligned}$$

Proof. Assume without loss of generality that the variables d and d' do not occur in D. It suffices to show that the following is a theorem of T:

$$\begin{aligned}&\mathit{standard}(d) \wedge \mathit{natp}(d) \wedge \mathit{natp}(d') \wedge d \leq d' \wedge u_{d,b} \neq \mathit{NIL} \Rightarrow \\ &\quad u_{d',b} = \mathit{cons}(\mathit{car}(u_{d,b}), \mathit{NIL})\end{aligned}$$

Moreover, we can restrict our attention to terms u that are right-hand sides of equations in D — the complete theorem follows from this by structural induction.

We will prove a slight generalization of this. Let $A_{d,d'}$ be the universal closure of the following conjuction of formulas as u ranges over subterms of D and b ranges over $\{0, 1\}$:

$$\begin{aligned}&\bigwedge_{u,b} \mathit{natp}(d) \wedge \mathit{natp}(d') \wedge d \leq d' \wedge u_{d,b} \neq \mathit{NIL} \Rightarrow \\ &\quad u_{d',b} = \mathit{cons}(\mathit{car}(u_{d,b}), \mathit{NIL})\end{aligned}$$

Note that since D is finite, $A_{d,d'}$ is well defined.

Let $y_1, \ldots, y_n$ be all the variables occurring in D. We now prove that $(\forall y_1, \ldots, y_n)(\forall d')(\mathit{standard}(d) \wedge \mathit{natp}(d) \Rightarrow A_{d,d'})$ is a theorem of T. We prove this using induction on d. Note: the hypothesis that d is standard is required precisely so we can use induction in this step. Recall that induction on non-classical formulas (such as a formula involving D) can only be used to justify the theorem for standard values.

It suffices to show that the following is true in T:

$$\begin{aligned}&((\forall e < d)(\forall y_1, \ldots, y_n)(\forall d') \\ &\quad (A_{e,e'} \Rightarrow (d \leq d' \wedge u_{d,b} \neq \mathit{NIL} \Rightarrow u_{d',b} = \mathit{cons}(\mathit{car}(u_{d,b}), \mathit{NIL}))\end{aligned}$$

where e ranges over the naturals.

To prove this, we use induction on the subterms u of D. Assuming the appropriate hypotheses, we will show $u_{d',b} = \mathit{cons}(\mathit{car}(u_{d,b}), \mathit{NIL})$ by considering each of the possibilities for u.

When u is a constant or variable, $u_{d',b}$ and $u_{d,b}$ are both equal to $\mathit{cons}(u, \mathit{NIL})$, so the conclusion follows trivially.

Otherwise, if u is not an if-term it takes the form $u = f(t_1, \ldots, t_n)$. Since $u_{d,b} \neq \mathit{NIL}$, we can immediately conclude that $(t_i)_{d,b} \neq \mathit{NIL}$ for each of the t_i. Moreover, since each of the t_i is a subterm of D, we can conclude from $A_{d,d'}$ in the hypothesis $(t_i)_{d',0} = \mathit{cons}(\mathit{car}((t_i)_{d,0}), \mathit{NIL})$. If $f \notin F$, then we have

$$\begin{aligned}u_{d',b} &= \mathit{cons}(f(\mathit{car}((t_1)_{d',0}), \ldots, \mathit{car}((t_n)_{d',0})), \mathit{NIL}) \\ &= \mathit{cons}(f(\mathit{car}((t_1)_{d,0}), \ldots, \mathit{car}((t_n)_{d,0})), \mathit{NIL}) \\ &= u_{d,b}\end{aligned}$$

When $f \in F$ we proceed differently. From the definition of $u_{d,b}$, we have that $u_{d',b} = f'(d', car((t_1)_{d',0}), \ldots, car((t_n)_{d',0}))$, so we need to show the following:

$$\begin{array}{c} f'(d', car((t_1)_{d',0}), \ldots, car((t_n)_{d',0})) = \\ f'(d, car((t_1)_{d,0}), \ldots, car((t_n)_{d,0})) \end{array}$$

We will show something stronger, namely that for arbitrary $x_1, \ldots, x_n$ such that $f'(d', x_1, \ldots, x_n) \neq NIL$,

$$f'(d', x_1, \ldots, x_n) = cons(car(f'(d, x_1, \ldots, x_n)), NIL).$$

Let $f(x_1, \ldots, x_n) = body$ be the equation in F that defines f. Since $d' \geq d$ and $f(x_1, \ldots, x_n)_{d,b} \neq NIL$, we know that $d' \geq d > 0$, so $f'(d, x_1, \ldots, x_n) = body_{d-1,1}$ and $f'(d', x_1, \ldots, x_n) = body_{d'-1,1}$, neither of which is NIL. Observe that *body* is one of the subterms of right-hand side equations of D, so from the induction hypothesis we have that $A_{d-1,d'-1}$ is true. Therefore,

$$\begin{array}{rcl} f'(d', x_1, \ldots, x_n) & = & body_{d'-1,1} \\ & = & cons(car(body_{d-1,1}), NIL) \\ & = & cons(car(f'(d, x_1, \ldots, x_n) x_1, \ldots, x_n)), NIL) \end{array}$$

It then follows that $u_{d',b} = u_{d,b}$ in this case as well.

The casse when $u = if\ u_0\ then\ u_1\ else\ u_2$ is the only one remaining for consideration. If $b = 0$, then this case is identical to the case when $u = f(t_1, \ldots, t_n)$ and $f \neq F$. If $b = 1$, then this case is similar to the previous one, except that only one of u_1 or u_2 needs to be considered. We leave that consideration to the reader. □

Corollary (Classical Interpreter Monotonicity). Let D be a classical definitional axiom over a theory T, and let u be a classical term in the language of $T \cup \{D\}$. Let d and d' be variables not occurring in u. Then the following is a theorem of T:

$$d \leq d' \wedge u_{d,b} \neq NIL \Rightarrow u_{d,b} = u_{d',b} \wedge u_{d',b} = cons(car(u_{d,b}), NIL)$$

Proof. By transfer from the Interpreter Monotonicity Lemma. □

Lemma (Interpreter Eliminability). Let u be a term in the language of $T \cup \{D\}$. Then the following is a theorem of T_D:

$$standard(d) \wedge u_{d,b} \neq NIL \Rightarrow u_{d,b} = cons(u, NIL)$$

Proof. We will prove this using induction on the structure of u.

If u is a variable or constant, the result is immediate.

If u is an if, it is handled similarly to the following case, and we omit the details.

If $u = f(u_1, \ldots, u_n)$, then we can conclude for each u_i that $(u_i)_{d,b} \neq NIL$, so by the induction hypothesis $(u_i)_{d,b} = cons(u_i, NIL)$. If $f \notin F$, then

$$\begin{aligned} u_{d,b} &= cons(f(car((u_1)_{d,0}), \ldots, car((u_n)_{d,0})), NIL) \\ &= cons(f(u_1, \ldots, u_n), NIL) \\ &= cons(u, NIL) \end{aligned}$$

If $f \in F$, $u_{d,b} = f'(d, u_1, \ldots, u_n)$. Since d is standard and $u_{d,b} \neq NIL$, it follows that $f(u_1, \ldots, u_n) = car(f'(e, u_1, \ldots, u_n))$ where e is the least standard such that $f'(e, u_1, \ldots, u_n) \neq NIL$. Necessarily, $e \leq d$, so using the Interpreter Monotonicity Lemma we can conclude that

$$\begin{aligned} u_{d,b} &= u_{e,b} \\ &= f'(e, u_1, \ldots, u_n) \\ &= cons(f(u_1, \ldots, u_n), NIL) \\ &= cons(u, NIL) \end{aligned}$$

Above we used implicitly the lemma that $f'(e, u_1, \ldots, u_n)$ is either NIL or of the form $cons(X, NIL)$ for some X. □

Lemma (Classical Interpreter Eliminability). Let D be a classical definitional axiom and u be a classical term in the language of $T \cup \{D\}$. Then the following is a theorem of T_D:

$$u_{d,b} \neq NIL \Rightarrow u_{d,b} = cons(u, NIL)$$

Proof. By transfer from the Interpreter Eliminability Lemma. □

Lemma (Divergence Infectiousness). Let u and d be terms. Then the following is a theorem of T_D: if d is standard, then $u_{d,0} = NIL$ if and only if for some subterm $f(t_1, \ldots, t_n)$ of u where $f \in F$,

$$f'(d, t_1, \ldots, t_n) = NIL.$$

Proof. By induction on the structure of the term u, using the Interpreter Eliminability Lemma. □

Lemma (Classical Divergence Infectiousness). Let D be a classical definitional axiom and u and d be classical terms in the language of $T \cup \{D\}$. Then the following is a theorem of T_D: $u_{d,0} = NIL$ if and only if for some subterm $f(t_1, \ldots, t_n)$ of u where $f \in F$, $f'(d, t_1, \ldots, t_n) = NIL$.

Proof. By transfer from the Divergence Infectiousness Lemma. □

Lemma (Interpreter Correctness). Suppose that D is interpreter admissible over the theory T. Then the axioms D are derivable from T_D.

Proof. Let $f(x_1, \ldots, x_n) = u$ be an arbitrary equation of D. Since D is interpreter admissible, for all possible values of the x_i there is a d such that $f'(d, x_1, \ldots, x_n) \neq NIL$, and moreover d is standard when D is a non-classical definition. We may assume d is the least such d. Thus,

$$\begin{aligned} f(x_1, \ldots, x_n) &= car(f'(d, x_1, \ldots, x_n)) \\ &= car(u_{d-1,1}) \\ &= u \end{aligned}$$

where the last equality follows from the appropriate Interpreter Eliminability Lemma. But this means that $f(x_1, \ldots, x_n) = u$ is derivable from T_D, and since $f(x_1, \ldots, x_n) = u$ is an arbitrary equation of D we are finished. □

We are now ready to show that measure admissibility implies interpreter admissibility. Be begin with a formal definition of measure admissibility.

Definition. Let D be a definitional axiom over a theory T. A *measure* for D associates with each conjunct $f(x_1, \ldots, x_n) = u$ of D a function $m_f(x_1, \ldots, x_n)$ that is first-order definable in T. Moreover, when D is non-classical the m_f must return standard values always. That is, $(\forall x_1, \ldots, x_n) standard(m_f(x_1, \ldots, x_n))$ is a theorem of T. □

Definition. A term t *rules* an occurrence of a term s in a term b if and only if b is of the form *if* b_0 *then* b_1 *else* b_2 and one of the following is true:

– the occurrence of s is in b_1 and either t is equal to b_0 or t rules s in b_1

– the occurrence of s is in b_2 and either t is equal to $\neg b_0$ or t rules s in b_2.

□

Definition. Let D be a definitional axiom with measure m over a theory T. The *measure theorem for D via m*, denoted $m(D)$, is the conjunction of all the terms

$$t_1 \wedge \ldots \wedge t_k \Rightarrow m_g(u_1, \ldots, u_l) \prec m_f(x_1, \ldots, x_n)$$

where $f(x_1, \ldots, x_n) = u$ is an equation of D, $g(u_1, \ldots, u_l)$ occurs in u, g is also defined in D, and the t_i are all the terms that rule the occurrence of $g(u_1, \ldots, g_l)$ in u. □

Definition. A definitional axiom D over a theory T is *measure admissible over* T if there is some measure m for D over T such that $m(D)$ is a theorem of T. □

We are now ready to prove that measure admissible definitions are also interpreter admissible. The following lemma is key.

Lemma (Measure Admissibility). Let u be a term of $T \cup \{D\}$. Then the following is a theorem of T. Suppose that for every subterm $f(t_1, \ldots, t_n)$ of u where $f \in F$

$$G \Rightarrow (\exists d)(standard(d) \wedge f'(d, t_1, \ldots, t_n) \neq NIL)$$

where G is the conjunction of all the terms that rule the occurrence of $f(t_1, \ldots, t_n)$ in u. Then,

$$(\exists d)(standard(d) \wedge u_{d,1} \neq NIL)$$

Proof. We prove this theorem in T_D. Since T_D is a conservative extension of T, this suffices to prove the theorem in T. We proceed by induction on the structure of u.

If u is of the form *if* u_0 *then* u_1 *else* u_2, and the occurrence of $f(t_1, \ldots, t_n)$ is in either u_1 or u_2, we proceed as follows. Assume without loss of generality that the occurrence is in u_1. Since no terms rule u_0 or any subterm of u_0, it follows from the Divergence Infectiousness Lemma that there is a standard d_0 such that $(u_0)_{d_0,0} \neq NIL$, and by the Monotonicity Lemma, $(u_0)_{d,0} \neq NIL$ for any standard $d \geq d_0$. For any such standard $d \geq d_0$, $(u_0)_{d,0} = cons(u_0, NIL)$ by the Interpreter Eliminability Lemma, so $u_{d,1} = (u_1)_{d,1}$. Applying the inductive hypothesis to u_1, we find a standard d_1 such that $(u_1)_{d_1,1} \neq NIL$. Letting d be the maximum of d_0 and d_1, we find a standard value of d such that $u_{d,1} \neq NIL$ as desired.

If u has any other form, then none of its subterms have any terms ruling them in u. So the conclusion follows trivially from the Divergence Infectiousness Lemma. □

Corollary (Classical Measure Admissibility). Let D be a classical definitional axiom and u be a term of $T \cup \{D\}$. Then the following is a theory of T. Suppose that for every subterm $f(t_1, \ldots, t_n)$ of u where $f \in F$, the following is true:

$$G \Rightarrow (\exists d)(f'(d, t_1, \ldots, t_n) \neq NIL)$$

where G is the conjunction of all the terms that rule the occurrence of $f(t_1, \ldots, t_n)$ in u. Then,

$$(\exists d)(u_{d,1} \neq NIL)$$

Proof. By transfer from the Measure Admissibility Lemma. □

Now we can present the main result of this section.

Theorem (Interpreter Admissibility). Let D be a measure admissible definitional axiom over the theory T. Then D is also interpreter admissible over T.

Proof. Since D is measure admissible, there is a measure m for D over T such that the measure theorem $m(D)$ is a theorem of T. Suppose that D is not interpreter admissible over T.

If D is not classical, the measure function m_f for each f defined by D always returns standard values. So it follows from the inductive completeness of T and the (assumed) failure of interpreter admissibility that there is an equation $f(x_1, \ldots, x_n) = u$ of D such that

$$\neg(\exists d)(standard(d) \wedge f'(d, x_1, \ldots, x_n) \neq NIL)$$

even though for all $g(y_1, \ldots, y_m)$ defined by D the following is true

$$\begin{aligned} & m_g(y_1, \ldots, y_m) \prec m_f(x_1, \ldots, x_n) \Rightarrow \\ & \quad (\exists d)(standard(d) \wedge g'(d, y_1, \ldots, y_m) \neq NIL) \end{aligned}$$

But in this case, the measure theorem and the Measure Admissibility Lemma can be used to conclude that there is some standard d such that $u_{d,1} \neq NIL$. But then, $f'(d+1, x_1, \ldots, x_n) = u_{d,1}$ by definition. This means that there is a standard d such that $f'(d, x_1, \ldots, x_n) \neq NIL$, resulting in a contradiction.

The case when D is classical is completely analogous. □

5. Implicit Definitions with defun-std

We now turn our attention to `defun-std`, which allows the introduction of a classical symbol from a non-classical body. Before such a definition is accepted, ACL2(r) checks that the body produces standard outputs when it is given standard inputs. This is meant to ensure the existence of the classical function introduced by this event.

Definition. A *standardizing definitional axiom D from a non-classical term* over a theory T is an equation of the following form

$$(\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} standard(x_i)) \Rightarrow f(x_1, \ldots, x_n) = term)$$

where the classical function symbol f is not in the language of T, *term* is a possibly non-classical term in the language of T such that *term* is provably (in T) standard whenever all the x_i are standard, the variables x_i are distinct, and these are the only variables free in *term*. $\square$

Consider an r-complete theory T and the following classical definitional axiom D from a non-classical term over T:

$$(\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f(x_1, \ldots, x_n) = \mathit{term})$$

We will now show how to construct a theory T' that is an r-complete, conservative extension of T such that D is a theorem of T'. Since T is r-complete, the following is a theorem of T.

$$\begin{aligned}((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \mathit{standard}(\mathit{term}))) \Rightarrow \\ ((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f_{\mathit{term}}(x_1, \ldots, x_n) = \mathit{term}))\end{aligned}$$

Note that $x_1, \ldots, x_n$ are precisely the free variables of *term*. Moreover, notice that the hypothesis in this theorem can be discharged from the restrictions imposed on *term*, namely that it return standard values for standard values of its parameters. Now consider the following equation:

$$f(x_1, \ldots, x_n) = f_{\mathit{term}}(x_1, \ldots, x_n)$$

Since f_{term} is a classical function in the language of T, this equation actually comprises a classical definitional axiom D' over T. Therefore, the theory T can be extended conservatively into an r-complete theory T' such that D' is a theorem of T'. But then D is necessarily a theorem of T'.

We prefer to introduce the function f using an axiom over the visible language of T[4]. So consider again the following definitional axiom

$$(\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f(x_1, \ldots, x_n) = \mathit{term})$$

We can safely assume that *term* is a term over the visible language of T. Otherwise, *term* must use a function symbol f_τ. We can remove f_τ simply by using `defun` to introduce the new (visible) function f' such that f' is equal to f_τ. Let T'' be the completion with respect to transfer of the extension of T by this visible definitional axiom. We claim that

[4] This is what is actually done in the implementation of ACL2(r). The non-visible function symbols are never used directly in the implementation.

this theory is precisely the theory T' defined above. The reason is that since f_τ and f are classical, we can use the transfer axiom to prove that f is equal to f_τ from the definitional axiom given above. Hence T'' is also complete with respect to induction and standardization.

6. Constrained Definitions with encapsulate

In this section, we consider how ACL2(r) works with `encapsulate` events and subsequent functional instantiations.

The story of `encapsulate` itself is a simple one. Essentially, an encapsulate event lets the user introduce as an axiom a theorem about a given function without introducing a definitional axiom for the function. However, the user must demonstrate that at least one function satisfies the proposed axiom by defining a "witness" function that does so. A careful argument given in [13] shows that this can be done conservatively[5]. The gist of this argument is that the theory introduced by an encapsulate event is a subtheory of the one that would result if the functions introduced by that encapsulate were simply replaced by their witnesses. Since we already know that explicitly defining a function results in a conservative extension, the (weaker) theory resulting from an encapsulate event is necessarily conservative. Note that this weaker theory is still r-complete.

The difficulty, however, lies with the correctness of functional instantiation, a derived inference rule in ACL2(r). Functional instantiation makes use of functional substitution, defined below. Give a theorem ϕ of T and a functional substitution that maps some function symbols in ϕ to other function symbols *preserving constraints on the replaced function symbols*, functional instantiation allows us to conclude that ϕ', the result of mapping the function symbols in ϕ by the functional substitution, is also a theorem of T.

Definition. Let T be a theory. A *simple functional substitution* is a function over the function symbols of (the language of) T that preserves arity and classicalness. That is, it maps classical function symbols to classical function symbols, non-classical function symbols to non-classical function symbols, unary function symbols to unary function symbols, binary function symbols to binary function symbols, etc. Moreover, a simple functional substitution is required to map each

[5] Despite the apparent simplicity of functional instantiation, its soundness as an inference rule is quite difficult to show, especially in conjunction with ACL2's other structuring mechanisms. Some soundness bugs in the treatment of functional instantiation in early versions of ACL2 motivated the development of [13].

function in the Ground Zero theory of ACL2(r) to itself. A simple functional substitution that maps each non-visible symbol of (the language of) T to itself and each visible symbol of T to a (possibly different) visible symbol of T is called a *visible simple functional substitution.* If X is a formula of T and fs is a simple functional substitution, the formula $X\backslash fs$ is the formula that results by substituting each functional instance in X with the function to which fs maps it. □

To see the validity of functional instantiation as a proof rule, we can proceed as follows. Suppose that ϕ is a theorem of some r-complete theory T in ACL2(r), and let fs be a functional substitution over this theory. We know there is a proof of ϕ in T. Suppose that $A\backslash fs$ is a theorem of T for each axiom A used in this proof of ϕ. Then it follows that $\phi\backslash fs$ is a theorem of T.

The trick is to show that $A\backslash fs$ is a theorem of T for every axiom A used in the proof. The reason this is difficult is that the axioms of T include not just definitional axioms, but also induction, transfer, and standardization axioms. The ACL2(r) implementation explicitly checks that $A\backslash fs$ is true of the definitional axioms, but it does not check any of the other proposed axioms. We will show that these are in fact axioms as well.

To make this notion explicit, [13] introduces the notion of a labeled formula. For our purposes, we can think informally of the labeled formulas of a theory T as the set of axioms directly introduced by the user during the course of an ACL2(r) session that defined T. This includes the axioms that define or constrain new function symbols, but it excludes all the induction, transfer, and standardization axioms added automatically by ACL2(r) on the user's behalf. Observe that all labeled formulas are in the visible language of T. With this notion we are ready to prove the validity of simple functional instantiation.

The following technical lemma is proved in [13].

Lemma. Suppose that ϕ is a theorem of a given first-order theory T and that fs is a simple functional substitution whose domain is disjoint from the set of function symbols of T. Then $\phi\backslash fs$ is a theorem of T. □

This lemma makes a deceptively simple claim: If a theorem involves function symbols that are not mentioned in the axioms of the theory in which it is proved, then the meaning of those function symbols is irrelevant, so the functions they represent can be replaced with different functions. We use this lemma to prove the following theorem.

Theorem (Simple Functional Instantiation). Let T be a first-order theory that is r-complete, let fs be a visible simple functional substitution over the language of T, and let ϕ be a theorem of T such that ϕ mentions only visible function symbols. Moreover, suppose that $A\backslash fs$ is a theorem of T for each labeled formula A in T. Then $\phi\backslash fs$ is a theorem of T.

Proof. Since ϕ is a theorem of T, there is some proof of ϕ in T. Fix one such proof. Let P be the conjunction of the axioms used in this proof of ϕ. Then $P \Rightarrow \phi$ is a theorem of a subtheory of T that does not contain any axioms about the function symbols in ϕ, e.g., the Ground Zero theory GZ.

We construct a simple functional substitution fs' that is an extension of fs as follows. Consider all standardization axioms A used in the fixed proof of T. Let τ_1, …, τ_m be the terms standardized by these axioms, such that τ_j is not less than τ_i according to the partial ordering implied by standardization when $i < j$. Then let $fs_0 = fs$, and define fs_i as the extension of fs_{i-1} that maps f_{τ_i} to $f_{\tau_i\backslash fs_{i-1}}$. The functional substitution fs' is equal to fs_m, i.e., the final extension. Since ϕ uses only visible symbols, notice that $\phi\backslash fs' = \phi\backslash fs$.

The preceding lemma assures us that $(P \Rightarrow \phi)\backslash fs'$ is also a theorem of this subtheory. But that means that $(P\backslash fs' \Rightarrow \phi\backslash fs')$ is a theorem of this subtheory and hence also of T. We will complete the proof by showing that $P\backslash fs'$ is a theorem of T, and hence $\phi\backslash fs'$ (which is syntactically equal to $\phi\backslash fs$) is also a theorem of T.

Consider each conjunct A of P, i.e., each axiom used in the fixed proof of ϕ. If A is a labeled formula of T, then by hypothesis $A\backslash fs$ is a theorem of T. Since labeled formulas are in the visible language of T, it follows that $A\backslash fs'$ is equal to $A\backslash fs$ and we're done.

If A is either an induction or a transfer axiom, then $A\backslash fs'$ is also an induction or transfer axiom. The reason is that $A\backslash fs'$ preserves the structure of A, changing only the function symbols. Since fs' also preserves classicalness, the formula $A\backslash fs'$ will be of the right type (e.g., classical or non-classical induction axiom as appropriate). And since T is r-complete, it follows that $A\backslash fs'$ is a theorem of T.

Finally, suppose A is a standardization axiom. Then A has the form

$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \mathit{standard}(\tau))) \Rightarrow$$
$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f_\tau(x_1, \ldots, x_n) = \tau))$$

where τ is a term with free variables $x_1, \ldots x_n$ and f_τ is classical. So $A \backslash fs'$ has the following form

$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow \mathit{standard}(\tau \backslash fs'))) \Rightarrow$$
$$((\forall x_1 \ldots x_n)((\bigwedge_{i=1}^{n} \mathit{standard}(x_i)) \Rightarrow f_\tau(x_1, \ldots, x_n) \backslash fs' = \tau \backslash fs'))$$

There must be an i such that τ is equal to τ_i, one of the formulas used in the construction of fs'. Then $f_\tau(x_1, \ldots, x_n) \backslash fs'$ is equal to $f_{\tau_i \backslash fs_{i-1}}(x_1, \ldots, x_n)$. Because the τ_i are ordered according to the implied ordering imposed by standardization, τ_i can not contain any of the f_{τ_j} for $j \geq i$. What this means is that $f_{\tau_i \backslash fs_{i-1}}(x_1, \ldots, x_n)$ is equal to $f_{\tau_i \backslash fs'}(x_1, \ldots, x_n)$. Therefore, $A \backslash fs'$ has the form of a standardization axiom, and since T is r-complete $A \backslash fs'$ is a theorem of T. $\Box$

ACL2(r) allows more general functional instantiations than are justified by the Simple Functional Instantiation Theorem. In particular, ACL2(r) allows the substitution of an n-ary functional variable f with a term of the form `(lambda` $(x_1, \ldots, x_n)$ `body)` where `body` may have free variables (other than the x_i). [13] shows how the Simple Functional Instantiation Theorem can be extended to justify these more general functional instantiations. The argument is essentially this. First, replace any free variables in the `body` with new, arbitrary, zero-arity functions (i.e., constants), and call the result `body`'. Then we can apply the Simple Functional Instantiation Theorem to `body`'. Since no assumptions are made on the new zero-arity functions (i.e., constants) introduced, they can be replaced with universally quantified variables in the final theorem.

The only complication is that we must ensure that classicalness is preserved when the new function symbols (corresponding to the free variables in `body`) are introduced. When the theorem we are proving via functional instantiation is classical, we can proceed as follows. When we replace the free variables in `body` with new function symbols, we replace them with classical function symbols. Observe that in any given model, a zero-arity classical function symbol must correspond to a standard constant. So when we generalize these zero-arity functions (constants) with universally quantified variables in the final theorem, we are forced to assume that these variables are standard. But since the theorem is classical, we can use transfer to remove this assumption.

When we are proving a non-classical theorem and the constrained functions we are instantiating are also non-classical, we can proceed in a straightforward manner. Simply replacing the free variables in `body` with unconstrained zero-arity functions (hence possibly non-standard constants) allows us to invoke the Simple Functional Instantiation Theorem directly. There is no need to worry about the classicalness of the zero-arity functions, since the proof of the original theorem already uses them in a non-classical context.

On the other hand, when the theorem we are proving is non-classical and the function we are instantiating is classical, we must disallow free variables in `body`. To see this, consider an arbitrary classical function $f(x)$. Since f is classical, it follows that

$$\mathit{standard}(x) \Rightarrow \mathit{standard}(f(x))$$

If we were to substitute $\lambda(x)(x+y)$ into this theorem, we would conclude that

$$\mathit{standard}(x) \Rightarrow \mathit{standard}(x+y)$$

But this is false, e.g., when $x = 0$ and y is an arbitrary non-standard number[6].

7. Skolem Definitions with defchoose

The final method of introducing new function symbols into ACL2(r) is `defchoose`. Let ϕ be a formula with free variables v, x_1, x_2, $\ldots$, x_n, and no other free variables. Let f be a new function symbol of arity n. The Skolem axiom introducing f from ϕ with respect to v is

$$\phi \Rightarrow \textbf{let } v = f(x_1, x_2, \ldots, x_k) \textbf{ in } \phi$$

What this axiom states is that the function f can "choose" an appropriate v for a given x_1, x_2, $\ldots$, x_k, provided such a choice is at all possible.

The event `defchoose` allows the user of ACL2(r) to introduce the new function f from an arbitrary formula ϕ with free variables v, x_1, x_2, $\ldots$, x_k as above. This presents two theoretical questions: (a) if you introduce the new symbol f into the r-complete theory T, why is the extended theory T' also r-complete? and (b) is the new symbol f classical? In this section we will address these issues.

Consider the extension of an r-complete theory T by a Skolem axiom introducing f from some formula ϕ; call the resulting theory T'. We assume here that the only variables free in ϕ are v, x_1, x_2, $\ldots$, x_n, and that f is an n-ary function symbol so that the Skolem axiom is given by

$$\phi \Rightarrow \textbf{let } v = f(x_1, x_2, \ldots, x_k) \textbf{ in } \phi$$

[6] This argument identified a bug in an earlier implementation of ACL2(r).

As we have seen in previous sections, T' will be a conservative, r-complete extension of T provided we can find an explicit definition of f in the language of T. The problem is that for a given set of x_i there may be more than one v that satisfies ϕ, so the value of f is not made explicit by this axiom.

One way to make the value chosen by f explicit is to introduce a well-ordering of the objects in the universe. With such a well-ordering, f can simply choose the smallest v that satisfies ϕ. The following lemma shows that we can introduce such a well-ordering to the initial ACL2(r) ground theory.

Lemma (Well-Ordering). It is consistent to extend the ACL2(r) Ground Zero Theory with a new predicate that well-orders the universe.

Discussion. Forcing is a logical technique that is used often to show that adding a particular axiom or function symbol to a theory is consistent. In fact, [13] uses model-theoretic forcing to show the consistency of `defchoose` in ACL2. The forcing argument originally given in [13], since corrected and extended in [11], can be modified to justify defchoose in ACL2(r), essentially by building an appropriate well-ordering on the fly. However, to do so requires that we reason about both external and internal sets. This is impossible to do in Internal Set Theory, which only knows about internal sets, so it is necessary to carry out the forcing argument in a larger set theory, such as Hrbaček Set Theory [7].

We prefer to give a more traditional (among non-standard analysts) argument proposed by Kaufmann [9]. This argument uses only the model-theoretic constructions that are typically used to justify non-standard analysis in the first place. What we will do is show how to transform a model of ACL2 into a model that includes non-standard analysis and a well ordering of the elements.

Proof (Well-Ordering Lemma). Consider a model of the Ground Zero Theory of ACL2 that includes the irrationals and complex numbers, and that satisfies the Axiom of Choice. [10] describes a constructive construction of such a model, without the irrational numbers, and that construction can be modified trivially to add the irrationals. Assign a name to each function on the universe of this model, so the language of the model is uncountable. Call the resulting model MS. Typical model-theoretic arguments[7] show that this model can be extended to a non-standard model MNS such that all first-order sentences that are true in MS also hold in MNS (a so-called elementary extension). MNS contains the function *standard* in a natural way: The standard elements

[7] A common approach is to take ultraproducts of the initial model.

are those in the universe of MS[8]. Since MNS satisfies the Axiom of Choice, its elements can be well-ordered. Let WO be a well-ordering relation in MNS.

Remark: Nelson's construction in the proof of the conservativity of IST over ZFC [16] uses the same model-theoretic arguments that show the existence of MNS. In fact, MNS is a model of Internal Set Theory; that is, MNS satisfies the axioms of IST. Consider the familiar properties of IST. Transfer holds in MNS directly by construction. Since MNS is an elementary extension of MS, any first-order property that is true in MS (and hence true of all standard elements of MNS) is also true in MNS. Standardization holds because all functions in the original model MS are named. Take any function f in MNS that maps standard elements to standard elements. Its restriction to the standard elements of MNS is a function f^{st} from the standard elements of MNS to the standard elements of MNS, and since the standard elements of MNS correspond to the elements of MS, there is a corresponding function $\hat{f}$ in MS. This function in MS is converted to a classical function ${}^*\hat{f}$ in MNS by the process of constructing MNS. This function ${}^*\hat{f}$ is precisely the standardization of f. Finally, non-standard induction holds in MNS because MS is a standard model: Since the interpretation of $\prec$ in MS really is well-founded, induction holds on MS, and hence on the standard elements in MNS. And classical induction follows simply from non-standard induction and transfer.

Therefore, MNS is a model of ACL2(r)—i.e., a model of ACL2 that also satisfies the axioms of IST—that contains a well-ordering relation WO. $\square$

We have just shown that it is consistent to add a well-ordering to a model of ACL2(r), so we assume that WO is such an ordering in the Ground Zero Theory of ACL2(r). Note that we do *not* assume in the Ground Zero Theory that we have a name for every possible function symbol—in fact, we assume only countable many function symbols in this theory. Also note that users never see this well-ordering explicitly; it is only used implicitly to define the functions using `defchoose`. As with `encapsulate`, the explicit definition of the function is not kept. Rather, ACL2(r) retains the Skolem Axiom introducing f from the formula ϕ.

We can now address the second question concerning `defchoose`: Are the new function symbols classical or not? The answer is that when the formula ϕ is classical, the function f that is introduced by a `defchoose` is also classical. Otherwise, the function is non-classical.

[8] More precisely, the universe of MNS contains an embedding of the universe of MS, and the elements in this embedding are the standard elements of MNS.

Lemma (Non-classsical Skolemization). Let T be an ACL2(r) theory. Let ϕ be a (possibly non-classical) formula with free variables v, x_1, x_2, $\ldots$, x_n, and no other free variables. Let f be a new non-classical function symbol of arity n. The r-closure of the extension of T by the Skolem axiom introducing f from ϕ with respect to v is a conservative extension of T.

Proof. It is sufficient to give an explicit definition for f. We can do so with the well-ordering relation WO. That is, let $f(x_1, \ldots, x_n)$ be the WO-minimal x such that $\phi(x, x_1, \ldots, x_n)$ holds. If there is no x such that $\phi(x, x_1, \ldots, x_n)$ holds, then let $f(x_1, \ldots, x_n)$ be `nil`. It is clear from the definition that the function f defined in this way satisfies the Skolem axiom introducing f from ϕ with respect to v. □

When ϕ is classical, we can use non-classical Skolemization and the standardization principle to find a *classical* function f that satisfies the Skolem axiom introducing f from ϕ. This is important, because it shows that ACL2(r)'s defchoose behaves analogously to ACL2's defchoose for the important case that ϕ is a classical formula; i.e., for all uses of defchoose in an ordinary ACL2 theory.

Lemma (Classsical Skolemization). Let T be a theory of ACL2(r). Let ϕ be a classical formula with free variables v, x_1, x_2, $\ldots$, x_n, and no other free variables. Let f be a new classical function symbol of arity n. The r-closure of the extension of T by the Skolem axiom introducing f from ϕ with respect to v is a conservative extension of T.

Proof. It is sufficient to give an explicit definition for f. We do so as follows. First, let ψ be the formula $\phi \wedge \mathit{standard}(v)$. Using the Non-classical Skolemization Lemma, we know there is an r-complete conservative extension of T with the (new) non-classical symbol g of arity n such that the Skolem axiom introducing g from ϕ with respect to v holds. We can now define f as the standardization of g.

To complete the proof, we need only show that the function f satisfies the Skolem axiom introducing f from ϕ with respect to v. Since ϕ is classical and f is classical (because it was defined explicitly using standardization), we need only consider standard values of the free variables in the Skolem axiom. Suppose v_1, $\ldots$, v_n are all standard. If there is no standard v such that $\phi(v, v_1, \ldots, v_n)$ holds, then by the transfer principle there is no v such that $\phi(v, v_1, \ldots, v_n)$ holds—in which case the Skolem axiom trivially holds. Otherwise, there is such a standard v. But then, we have that $\phi(v, v_1, \ldots, v_n) \wedge \mathit{standard}(v)$ holds. Therefore, $\psi(v, v_1, \ldots, v_n)$ is satisfied. By the Skolem axiom introducing g from ψ, we know that $\phi(g(v_1, \ldots, v_n), v_1, \ldots, v_n)$ holds. And since f is identical to g for standard arguments, $\phi(f(v_1, \ldots, v_n), v_1, \ldots, v_n)$ also holds. Since the choice of $v_1, \ldots, v_n$ was arbitrary, we have just shown that the Skolem axiom introducing f holds for all standard values of its free variables. Applying the transfer principle completes the proof. □

8. Concluding Remarks

The results in this paper show that the principles of definitional extension in ACL2(r) are sound, in fact conservative. Together with the exposition in [13] this gives a clear explanation of what it means to say that a formula is a theorem of a sequence of ACL2(r) events: The formula is a theorem if it is first-order derivable from (a) the Ground Zero theory of ACL2(r), (b) the union of all the definitional events in the ACL2(r) sequence that define a function used in the theorem, and (c) the induction, transfer, and standardization axioms of the language on the ACL2(r) sequence. This serves to put ACL2(r) in a firm logical foundation.

This paper describes a stronger version of ACL2(r) than the one described in [5], justifying enhancements to ACL2(r) that will make it more useful. In fact, deficiencies in ACL2(r) that we encountered while working on Taylor's Theorem [6] led to the research described in this paper.

These enhancements include the following:

– The previous version of ACL2(r) contained `standard-numberp`, a function with intended semantics of "standard *and* numeric." There was no way to refer to standard atoms, lists, etc. The new version of ACL2(r) introduced the function `standard` which can be applied meaningfully to all arguments. E.g., ACL2(r) now has support for standard lists. Moreover, ACL2(r) has been enhanced so that it recognizes that classical functions produce standard results for standard arguments. Previously, the user had to prove such theorems directly, sometimes requiring considerable effort.

– ACL2(r) supports the introduction of recursive functions with non-classical bodies; earlier versions of ACL2(r) only permitted classical recursive functions. This restores the beautiful symmetry between induction and recursion that is a hallmark of the Boyer-Moore Theorem Prover and ACL2. Notice in particular that the restriction on such recursive functions—that the measures give standard results for any arguments—mimics the restriction of the non-standard principle of induction—that induction can only be used to establish the truth of a formula for standard arguments.

This also has a practical consequence to ACL2(r). The logic of ACL2 has very limited support for quantifiers. As a result, ACL2 users often use recursive definitions to model bounded quantifiers. For example, it is a theorem of ACL2 that if all elements of a list are rational, the sum of the elements of this list is also rational. Rather than using a quantifier to write the hypothesis of this theorem, an ACL2 user would phrase this theorem by defining the function `rational-listp` that is true of all lists whose elements are rationals:

```
(defun rational-listp (l)
  (if (null l)
      t
    (and (rationalp (car l))
         (rational-listp (cdr l)))))
```

Using `rational-listp`, the desired theorem can be proved as follows:

```
(defthm rational-listp-sum-is-rational
  (implies (rational-listp l)
           (rationalp (sumlist l))))
```

This rephrasing of formulas involving quantifiers into recursive predicates is second nature to users of ACL2. But this technique did not work well in the previous version of ACL2(r), which proved frustrating to ACL2(r) users.

To see the problem, consider a similar theorem of ACL2(r): If a list has standard elements and the list is of standard length, the sum of the elements of the list also standard. While true, this theorem was inexpressible in the prior version of ACL2(r), because the function `standard-listp` requires the use of recursion with a non-classical body. In the current (under development) version of ACL2(r), this function can be defined as follows:

```
(defthm standard-listp (l)
  (if (null l)
      t
    (if (not (standardp (length l)))
        nil
      (and (standardp (car l))
           (standard-listp (cdr l))))))
```

The definition of this function is similar in structure to that of `rational-listp`. However, the extra condition on the length of the input list is required to guarantee that the measure function (e.g., the length of the input argument if that is standard, zero otherwise) always returns a standard value, as required by the theory developed in section 4.3.

- ACL2(r) supports the introduction of non-classical constrained functions with `encapsulate`. Previous versions of ACL2(r) allowed the user to define only classical functions with `encapsulate`. The formal description of encapsulate also illustrated a soundness bug (since fixed) in the implementation of `encapsulate` in earlier versions of ACL2(r).

- ACL2(r) now supports the introduction of non-classical functions with `defchoose`.

We are currently busy completing the enhancements to ACL2(r) outlined above. We expect these enhancements to be ready for distribution with the next general release of ACL2.

Acknowledgements

The authors would like to thank Matt Kaufmann for his patience, promptness, and thoroughness in answering all questions related to [13]. It is simply stating the facts to say that this paper could not have been written without his help. We would also like to thank him for fruitful discussions that led to this paper in the first place.

References

1. Boyer, R. S., D. Goldschlag, M. Kaufmann, and J. S. Moore: 1991, 'Functional Instantiation in First Order Logic'. In: V. Lifschitz (ed.): *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy.* pp. 7–26.
2. Boyer, R. S. and J. S. Moore: 1979, *A Computational Logic.* Orlando: Academic Press.
3. Boyer, R. S. and J. S. Moore: 1988, *A Computational Logic Handbook.* San Diego: Academic Press.
4. Diener, F. and M. Diener (eds.): 1995, *Nonstandard Analysis in Practice.* Springer.
5. Gamboa, R. and M. Kaufmann: 2001, 'Nonstandard analysis in ACL2'. *Journal of Automated Reasoning* **27**(4), 323–351.

6. Gamboa, R. and B. Middleton: 2002, 'Taylor's Formula with Remainder'. In: *Proc of the Third International Workshop of the ACL2 Theorem Prover and its Applications (ACL2-2002)*.
7. Kanovei, V. and M. Reeken: 2004, *Nonstandard Analysis, Axiomatically*. Springer.
8. Kaufmann, M.: 2000, 'Modular Proof: The Fundamental Theorem of Calculus'. In: M. Kaufmann, P. Manolios, and J. S. Moore (eds.): *Computer-Aided Reasoning: ACL2 Case Studies*. Kluwer Academic Press, Chapt. 6.
9. Kaufmann, M.: 2005, 'Defchoose in ACL2r'. Private communication.
10. Kaufmann, M., P. Manolios, and J. S. Moore: 2000, *Computer-Aided Reasoning: An Approach*. Kluwer Academic Press.
11. Kaufmann, M. and J. S. Moore, 'ACL2 Documentation Topic on Conservativity of Defchoose'. Available in the ACL2 distribution since version 2.9.2 with :DOC CONSERVATIVITY-OF-DEFCHOOSE.
12. Kaufmann, M. and J. S. Moore, 'A Precise Description of the ACL2 Logic'. `http://www.cs.utexas.edu/users/moore/publications/km97a.ps.Z`.
13. Kaufmann, M. and J. S. Moore: 2001, 'Structured Theory Development for a Mechanized Logic'. *Journal of Automated Reasoning* **26**(2), 161–203.
14. M. Gameiro and P. Manolios: 2004, 'Formally Verifying an Algorithm Based on Interval Arithmetic for Checking Transversality'. In: *Fifth International Workshop on the ACL2 Theorem Prover and Its Applications (ACL2-2004)*.
15. Nelson, E., 'On-Line Books: Internal Set Theory'. Available on the world-wide web at `http://www.math.princeton.edu/\~nelson/books.html`.
16. Nelson, E.: 1977, 'Internal Set Theory: A New Approach to Nonstandard Analysis'. *Bulletin of the American Mathematical Society* **83**, 1165–1198.
17. Robert, A.: 1988, *Non-Standard Analysis*. John Wiley.
18. Robinson, A.: 1996, *Non-Standard Analysis*. Princeton University Press.
19. Sawada, J. and R. Gamboa: 2002, 'Mechanical Verification of a Square Root Algorithm using Taylor's Theorem'. In: *Formal Methods in Computer-Aided Design (FMCAD'02)*.