Knuth's Generalization of McCarthy's 91 Function

by

John Cowles
University of Wyoming

On sabbatical at
Advanced Technology Center
Rockwell-Collins
Cedar Rapids, Iowa

McCarthy's 91 Function

$$M(x) \stackrel{\text{def}}{\Leftarrow} \text{ if } \; x > 100 \; \text{ then } \; x - 10 \\ \qquad\qquad\qquad\qquad\qquad \text{ else } \; M(M(x+11)).$$

For integer inputs x,

$$M(x) \; = \; \text{if } \; x > 100 \; \text{ then } \; x - 10 \\ \qquad\qquad\qquad\qquad \text{ else } \; 91.$$

So for all integers $x \leq 101$, $M(x) = 91$.

$$M(x) \stackrel{\mathsf{def}}{\Leftarrow} \text{ if } x > 100 \text{ then } x - 10 \\ \text{else } M(M(x+11)).$$

Hard for ACL2 to verify the recursion terminates.

$$M(x) \stackrel{\text{def}}{\Leftarrow} \text{ if } \; x > 100 \; \text{ then } \; x - 10$$
$$\text{else } \; 91.$$

ACL2 Theorem.

$$M(x) \; = \; \text{if } \; x > 100 \; \text{ then } \; x - 10$$
$$\text{else } \; M(M(x + 11)).$$

Knuth's Generalization

$$K(x) \stackrel{\text{def}}{\Leftarrow} \text{if} \quad x > a \quad \text{then} \quad x - b \quad \text{else} \quad K^c(x+d).$$

- Real a
- Real $b > 0$
- Integer $c > 0$
- Real $d > 0$

The generalized 91 recursion with parameters (a, b, c, d)

Knuth's Theorem

Theorem 1 *The generalized 91 recursion with parameters* (a, b, c, d) *defines a total function on the integers iff* $(c-1) \cdot b < d$.
In such a case the values of $K(x)$ *also obey the simpler recurrence*

$$\begin{aligned} K(x) \;\; = \;\; \textit{if} \;\; x > a \;\; & \textit{then} \;\; x - b \\ & \textit{else} \;\; K(x + d - (c-1)b). \end{aligned}$$

Knuth's Challenge

Prove Theorem 1 by computer.

Knuth's Proof

$$K(x) \;=\; \text{if} \;\; x > a \;\; \text{then} \;\; x - b \;\; \text{else} \;\; K^c(x + d).$$

1. If $(c-1) \cdot b \geq d$, then the recursion does not terminate for some inputs $x \leq a$.

2. If $(c-1) \cdot b < d$, then the recursion always terminates.

3. Any total function that satisfies the generalized 91 recursion (above) must also satisfy the simpler recursion (below).

$$K(x) \;=\; \text{if} \;\; x > a \;\; \text{then} \;\; x - b \;\; \text{else} \;\; K(x + d - (c-1)b).$$

(Meta)Corollaries to ACL2 Results

0. For all choices of Real a, Real $b > 0$, Integer $c > 0$, and Real $d > 0$, there are total functions on the Reals that satisfy the generalized 91 recursion.

1. If $(c - 1) \cdot b \geq d$, then there are *many* total functions on the reals that satisfy the generalized 91 recursion.

2. If $(c - 1) \cdot b < d$, then there is *only one* total function on the reals that satisfies the generalized 91 recursion.

3. Any total function that satisfies the generalized 91 recursion must also satisfy the simpler recursion.

Total Functions and Recursive Definitions

Question. How many total functions, over the nonnegative integers, satisfy the recursive equation?

Example 1

$$
\begin{aligned}
F(x) \;=\; & \textit{if} \;\; x = 0 \;\; \textit{then} \;\; 0 \\
& \qquad\qquad\quad \textit{else} \;\; F(x-1).
\end{aligned}
$$

Example 2

$$
\begin{aligned}
G(x) \;=\; & \textit{if} \;\; x = 0 \;\; \textit{then} \;\; 0 \\
& \qquad\qquad\quad \textit{else} \;\; G(x)+1.
\end{aligned}
$$

Example 3

$$
\begin{aligned}
H(x) \;=\; & \textit{if} \;\; x = 0 \;\; \textit{then} \;\; 0 \\
& \qquad\qquad\quad \textit{else} \;\; H(x+1).
\end{aligned}
$$

Total Functions and Recursive Definitions

Example 1

$$\begin{aligned} F(x) \;\; = \;\; \textit{if} \;\; x = 0 \;\; \textit{then} \;\; & 0 \\ \textit{else} \;\; & F(x-1). \end{aligned}$$

Only one function, over the nonnegative integers, satisfies the recursive equation.

$$F(x) \stackrel{\text{def}}{\Leftarrow} 0$$

Total Functions and Recursive Definitions

Example 2

$$G(x) = \textit{if } x = 0 \textit{ then } 0 \textit{ else } G(x) + 1.$$

No function, over the nonnegative integers, satisfies the recursive equation.

For $x > 0$,

$$G(x) = G(x) + 1$$

Total Functions and Recursive Definitions

Example 3

$$\begin{array}{rcl} H(x) & = & \textit{if} \;\; x = 0 \;\; \textit{then} \;\; 0 \\ & & \qquad\qquad\qquad \textit{else} \;\; H(x+1). \end{array}$$

Many functions, over the nonnegative integers, satisfy the recursive equation.

For each integer $n \geq 0$,

$$\begin{array}{rcl} H_n(x) & \overset{\text{def}}{\Leftarrow} & \text{if} \;\; x = 0 \;\; \text{then} \;\; 0 \\ & & \qquad\qquad\qquad \text{else} \;\; n. \end{array}$$

$$\begin{array}{rcl} H_n(0) & = & 0 \\ H_n(x) & = & n = H_n(x+1) \text{ for } x > 0. \end{array}$$

Total Functions and Recursive Definitions

Example 4

$$L(x,y) \;=\; \textit{if} \;\; x=0 \;\; \textit{then} \;\; 0 \;\; \textit{else} \;\; L(x-1, L(x,y)).$$

- *Only one* function, over the nonnegative integers, satisfies the recursive equation.

$$L(x,y) \stackrel{\text{def}}{\Leftarrow} 0$$

- Recursion does *not* terminate.
  Assuming left-to-right, innermost-first evaluation order.

The Reals

Theorem 2 *There is (up to ordered field isomorphism) exactly one complete ordered field.*

This unique field is known as the field of *real numbers* or the *Reals*.

Completeness Every nonempty set of field elements bounded above has a *least upper bound*.

Example $\{x \in \text{Reals} \mid x^2 \leq 2\}$ has least upper bound, $\sqrt{2}$.

Achimedean Ordered Fields

An ordered field is *Archimedean* iff every field element is bounded above by some positive integer.

The Archimedean fields are exactly the subfields of the Reals.

- Rationals are Archimedean.
- Reals are Archimedean.

Archimedean fields have ACL2 axioms that ACL2 can show are consistent.

Prove the claims about the *Generalized 91 Recursion* over arbitrary Archimedean fields (instead of just over the Reals).

Axiomatize Archimedean Fields in ACL2

- Constrain with encapsulate
  Witness: Rationals

- Archimedean Axioms
  - Function: `Nat-Int-Bound`
  - `Nat-Int-Bound`(x) is an Integer
  - `Nat-Int-Bound`$(x) \geq_a x$, if $x \geq_a 0$
  - Witness: `numerator`

- Define useful function:
  `Least-Nat-Bound`(x) returns the least Natural *larger* than x

Axiomatize the Parameters in ACL2

Use ecapsulate to constrain:

- `(a)`, `(b)` $>_a 0$, and `(d)` $>_a 0$ are elements of an Archimedean field.

- Integer `(c)` > 0.

Define ACL2 Functions `K1` and `K2`

- Prove `K1` and `K2` satisfy the Generalized 91 Recursion with parameters $(\texttt{(a)}, \texttt{(b)}, \texttt{(c)}, \texttt{(d)})$.

- Prove $\texttt{K1} \neq \texttt{K2}$ when $(\texttt{(c)} - 1) \cdot_a \texttt{(b)} \geq_a \texttt{(d)}$.

- Use Simpler Recursion to help define `K1` and `K2`.

$$
\begin{array}{rcl}
K(x) & = & \text{if } \; x > a \\
 & & \qquad \text{then } \; x - b \\
 & & \qquad \text{else } \; K(x + d - (c-1)b).
\end{array}
$$

Definitions for `K1` and `K2`

$$
\begin{aligned}
K1(x) \stackrel{\text{def}}{\Leftarrow}\ & \text{if } x > a \\
& \quad \text{then } x - b \\
& \quad \text{else if } (c-1)\cdot b < d \\
& \qquad \text{then } K1(x + d - (c-1)b) \\
& \qquad \text{else } a.
\end{aligned}
$$

$$
\begin{aligned}
K2(x) \stackrel{\text{def}}{\Leftarrow}\ & \text{if } x > a \\
& \quad \text{then } x - b \\
& \quad \text{else if } (c-1)\cdot b < d \\
& \qquad \text{then } K2(x + d - (c-1)b) \\
& \qquad \text{else } a - b.
\end{aligned}
$$

Axiomatize ACL2 Function `K`

- Constrain `K` to satisfy Generalized 91 Recursion with parameters (`(a)`, `(b)`, `(c)`, `(d)`).
  - Axiom

$$K(x) \;=\; \text{if } \; x > a \;\; \text{then } \; x - b \;\; \text{else } \; K^{c}(x + d).$$

  - Use encapsulate
    Witness: `K1`

- Prove `K` satisfies Simpler Recursion.

$$K(x) \;=\; \text{if } \; x > a \;\; \text{then } \; x - b \;\; \text{else } \; K(x + d - (c - 1)b).$$

- Prove `K` $=$ `K1` when $(\texttt{(c)} - 1) \cdot_a \texttt{(b)} <_a \texttt{(d)}$.

ACL2 Proof of $\texttt{K} = \texttt{K1}$ when $\cdots$

- Constrain ACL2 function `G1` to satisfy Simpler Recursion.
  - Axiom

    $$G1(x) \;=\; \text{if } \; x > a \quad \text{then } \; x - b \quad \text{else } \; G1(x + d - (c-1)b).$$

  - Use encapsulate
    Witness: `K1`

- Prove $\texttt{G1} = \texttt{K1}$ when $((\texttt{c}) - 1) \cdot_a (\texttt{b}) <_a (\texttt{d})$.

- Replace `G1` with `K` in above item.
  Use Functional Instantiation

Archimedean ordered field operations extend operations on rationals.

- `Rationalp` extended by `aofp`
- $+$ extended by $+_a$
- $*$ extended by $*_a$
- $<$ extended by $<_a$
- etc.

Life would be easier:

- Extend linear arithmetic to $<_a$.
- Improved ordered arithmetic.