Adding APPLY to ACL2 – Work in Progress

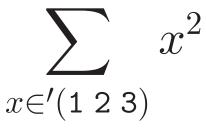
Matt Kaufmann J Strother Moore

Department of Computer Science University of Texas at Austin

January, 2016

Motivation

Iterative constructs are common in all programming languages — except ACL2.



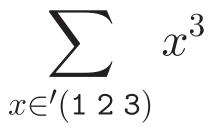
(loop for x in '(1 2 3) sum (sq x))

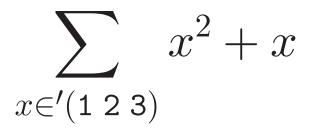
(sumlist '(1 2 3) 'sq)

But in ACL2...

(sum-sq '(1 2 3))

Now Write These in ACL2





$$\sum_{x \in '(1\ 2\ 3)} x^2 + 2x + 1$$

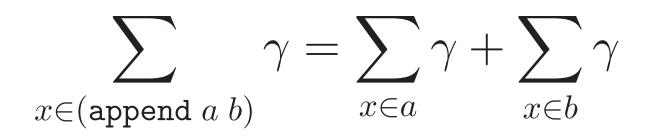
4

Each requires a different ACL2 function, sum-sq, sum-cubes, sum-sq+x, sum-yet-another-poly.

Two Beautiful Things about Iterative Notation

Succinct: Many different computations can be described with the same control structure.

General: Lemmas can be proved about the control structure independent of the particulars.



(sum-sq (append a b))
= (+ (sum-sq a) (sum-sq b))

$$(sum-sq+x (append a b))$$

= (+ (sum-sq+x a) (sum-sq+x b))

Goals

Make it possible to define such functions as:

to prove and use such lemmas as:

```
(defthm sumlist-append
  (equal (sumlist (append a b) fn)
                     (+ (sumlist a fn)
                          (sumlist b fn))))
```

and to reason about and execute such terms as

```
(sumlist lst 'sq)
(sumlist lst 'cube)
(sumlist lst '(lambda (x) (+ (* x x) x)))
(sumlist lst '(lambda (x) (+ (* x x) (* 2 x) 1)))
```