EVENT: Start with the initial **nqthm** theory.

DEFINITION:
bubble1 (l)
$=$ **if** $\neg$ listp (l) **then** l
elseif $\neg$ listp (cdr (l)) **then** l
elseif car $(l) <$ car (bubble1 (cdr (l)))
then cons (car (l), bubble1 (cdr (l)))
else cons (car (bubble1 (cdr (l))),
cons (car (l), cdr (bubble1 (cdr (l))))) **endif**

DEFINITION:
length (lst)
$=$ **if** listp (lst) **then** $1 +$ length (cdr (lst))
else 0 **endif**

THEOREM: bubble1-preserves-length
length (bubble1 (l)) $=$ length (l)

THEOREM: stupid-lemma-1
listp $(l) \rightarrow$ (length (cdr (bubble1 (l))) $<$ length (l))

DEFINITION:
bubblesort (l)
$=$ **if** $\neg$ listp (l) **then nil**
else cons (car (bubble1 (l)), bubblesort (cdr (bubble1 (l)))) **endif**

DEFINITION:
sortedp (l)
$=$ **if** $\neg$ listp (l) **then t**
elseif $\neg$ listp (cdr (l)) **then t**
else (car $(l) \leq$ cadr (l)) $\wedge$ sortedp (cdr (l)) **endif**

THEOREM: member-cdr-bubble1-not-lessp
$(x \in$ cdr (bubble1 (l))) $\rightarrow$ (($x <$ car (bubble1 (l))) $=$ **f**)

THEOREM: bubblesort-is-sorted
sortedp (bubblesort (l))

THEOREM: bubblesort-is-sortedp
sortedp (bubblesort (l))

```
; the last of these seems to be new
                                  ; somehow, 2/20/88
```

EVENT: Introduce the function symbol *amb* of 2 arguments.

AXIOM: amb-axiom
$(\text{amb}(x,\ y) = x) \vee (\text{amb}(x,\ y) = y)$

DEFINITION:
state (i)

$$
\begin{aligned}
= \quad & \textbf{if } i \in \mathbf{N} \\
& \textbf{then if } i \simeq \mathtt{0} \ \textbf{ then } \text{cons}(\mathtt{0},\ \mathtt{1}) \\
& \qquad \textbf{elseif } \text{car}(\text{state}(i - 1)) \simeq \mathtt{0} \\
& \qquad \textbf{then } \text{amb}(\text{cons}(\mathtt{1},\ \text{cdr}(\text{state}(i - 1))), \\
& \qquad\qquad\qquad \text{cons}(\mathtt{0},\ 1 + \text{cdr}(\text{state}(i - 1)))) \\
& \qquad \textbf{else } \text{cons}(\text{car}(\text{state}(i - 1)),\ \text{cdr}(\text{state}(i - 1)) - 1) \ \textbf{endif} \\
& \textbf{else } \text{cons}(\mathtt{0},\ \mathtt{0}) \ \textbf{endif}
\end{aligned}
$$

EVENT: Introduce the function symbol *constant* of 0 arguments.

AXIOM: constant-axiom

$$
\begin{aligned}
& ((\text{CONSTANT} \in \mathbf{N}) \wedge (\text{car}(\text{state}(\text{CONSTANT})) = \mathtt{1})) \\
\vee \quad & (\neg ((x \in \mathbf{N}) \wedge (\text{car}(\text{state}(x)) = \mathtt{1})))
\end{aligned}
$$

DEFINITION:
zz (x)

$$
\begin{aligned}
= \quad & \textbf{if } x \simeq \mathtt{0} \ \textbf{ then } \mathbf{t} \\
& \textbf{else } \text{zz}(x - 1) \ \textbf{endif}
\end{aligned}
$$

THEOREM: car-of-state-is-numberp
$\text{car}(\text{state}(x)) \in \mathbf{N}$

THEOREM: cdr-of-state-is-numberp
$\text{cdr}(\text{state}(x)) \in \mathbf{N}$

DEFINITION:
thm-ind $(x,\ y)$

$$
\begin{aligned}
= \quad & \textbf{if } y \simeq \mathtt{0} \ \textbf{ then } \mathbf{t} \\
& \textbf{else } \text{thm-ind}(1 + x,\ y - 1) \ \textbf{endif}
\end{aligned}
$$

THEOREM: main-base
t

THEOREM: main-thm

$$
\begin{aligned}
& ((x \in \mathbf{N}) \wedge (\text{car}(\text{state}(x)) = \mathtt{1}) \wedge (\text{cdr}(\text{state}(x)) = y)) \\
\rightarrow \quad & (\text{state}(x + y) = \mathtt{'(1\ .\ 0)})
\end{aligned}
$$

THEOREM: pnueli-1
$((\text{CONSTANT} \in \mathbf{N}) \wedge (\text{car}\,(\text{state}\,(\text{CONSTANT})) = 1))$
$\rightarrow \quad (\text{cdr}\,(\text{state}\,(\text{CONSTANT} + \text{cdr}\,(\text{state}\,(\text{CONSTANT})))) = 0)$

THEOREM: car-of-state-is-zero-or-1
$(\text{car}\,(\text{state}\,(x)) \neq 1) \rightarrow (\text{car}\,(\text{state}\,(x)) = 0)$

THEOREM: pnueli-2
$(\neg\,((x \in \mathbf{N}) \wedge (\text{car}\,(\text{state}\,(x)) = 1))) \rightarrow (\text{car}\,(\text{state}\,(x)) \simeq 0)$

THEOREM: pnueli-thm
$(\text{cdr}\,(\text{state}\,(\text{CONSTANT} + \text{cdr}\,(\text{state}\,(\text{CONSTANT})))) = 0)$
$\vee \quad (\text{car}\,(\text{state}\,(x)) \simeq 0)$

```
; Begin commutativity of times
```

DEFINITION:
double-ind $(x,\, y)$
$=$ **if** $(x \simeq 0) \vee (y \simeq 0)$ **then t**
else double-ind $(x,\, y - 1)$
$\wedge$ double-ind $(x - 1,\, y)$
$\wedge$ double-ind $(x - 1,\, y - 1)$ **endif**

```
; the following has USE-GOAL and hence is omitted, 2/88
#|
(PROVE-LEMMA TIMES-COMM
     (REWRITE)
     (EQUAL (TIMES X Y) (TIMES Y X))
     ((INSTRUCTIONS (INDUCT (DOUBLE-IND X Y))
    S SPLIT SUBV S
    (PUSH TIMES-0)
    PROVE SUBV S USE-GOAL
    (HIDE-HYPS 1 2 3)
    (PUSH TIMES-NOT-NUMBERP)
    PROVE USE-GOAL S DIVE PUSH DIVE
    (DIVE 1)
    X
    (DIVE 2)
    (= * (TIMES Y (SUB1 X)) T)
    X UP NX X
    (DIVE 2)
    (= * (TIMES X (SUB1 Y)) T)
    X UP UP
    (DIVE 1 2 2)
```

```
    (= * (TIMES (SUB1 X) (SUB1 Y)) T)
    TOP
    (GENERALIZE (((TIMES (SUB1 X) (SUB1 Y)) ?)))
    PROVE)))
|#
```

DEFINITION:
flatten (x)
$=$ **if** $x \simeq$ **nil** **then** list (x)
else append (flatten (car (x)), flatten (cdr (x))) **endif**

DEFINITION:
mcflatten (x, y)
$=$ **if** $x \simeq$ **nil** **then** cons (x, y)
else mcflatten (car (x), mcflatten (cdr (x), y)) **endif**

DEFINITION:
fmfind (x, y)
$=$ **if** $x \simeq$ **nil** **then** **t**
else fmfind (car (x), mcflatten (cdr (x), y)) $\wedge$ fmfind (cdr (x), y) **endif**

THEOREM: bookmark
t

THEOREM: assoc-of-append
append (append (x, y), z) $=$ append (x, append (y, z))

THEOREM: flatten-mcflatten
mcflatten (x, y) $=$ append (flatten (x), y)

THEOREM: plus-assoc
$(x + y + z) = ((x + y) + z)$

THEOREM: assoc-plus
$(x + y + z) = ((x + y) + z)$

DEFINITION:
rev (l)
$=$ **if** $l \simeq$ **nil** **then** **nil**
else append (rev (cdr (l)), list (car (l))) **endif**

DEFINITION:
plistp (l)
$=$ **if** $l \simeq$ **nil** **then** $l =$ **nil**
else plistp (cdr (l)) **endif**

THEOREM: rev-rev
plistp $(l) \rightarrow (\text{rev}(\text{rev}(l)) = l)$

THEOREM: rev-plistp
plistp $(\text{rev}(l))$

THEOREM: rev-append-1
rev $(y) = \text{append}(\text{rev}(y),\ \mathbf{nil})$

THEOREM: rev-append
rev $(\text{append}(x,\ y)) = \text{append}(\text{rev}(y),\ \text{rev}(x))$

Index