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

Definition:
$\mathrm{h}(n,\ \mathit{from},\ \mathit{to},\ \mathit{temp})$
$=$ **if** $n \simeq \mathtt{0}$ **then nil**
else $\mathrm{append}(\mathrm{h}(n-1,\ \mathit{from},\ \mathit{temp},\ \mathit{to}),$
$\mathrm{cons}(\mathrm{cons}(\mathit{from},\ \mathit{to}),\ \mathrm{h}(n-1,\ \mathit{temp},\ \mathit{to},\ \mathit{from})))$ **endif**

Definition: $\mathrm{hanoi}(n) = \mathrm{h}(n,\ \mathtt{0},\ \mathtt{1},\ \mathtt{2})$

Definition:
$\mathrm{nth}(i,\ l)$
$=$ **if** $i \simeq \mathtt{0}$ **then** $\mathrm{car}(l)$
else $\mathrm{nth}(i-1,\ \mathrm{cdr}(l))$ **endif**

Definition:
$\text{set-nth}(i,\ l,\ v)$
$=$ **if** $i \simeq \mathtt{0}$ **then** $\mathrm{cons}(v,\ \mathrm{cdr}(l))$
else $\mathrm{cons}(\mathrm{car}(l),\ \text{set-nth}(i-1,\ \mathrm{cdr}(l),\ v))$ **endif**

Definition:
$\mathrm{legal}(\mathit{move},\ \mathit{st})$
$=$ $(\mathrm{listp}(\mathit{move})$
$\land$ $(\mathrm{car}(\mathit{move}) \in \mathtt{'(0\ 1\ 2)})$
$\land$ $(\mathrm{cdr}(\mathit{move}) \in \mathtt{'(0\ 1\ 2)})$
$\land$ $\mathrm{listp}(\mathrm{nth}(\mathrm{car}(\mathit{move}),\ \mathit{st}))$
$\land$ $((\mathrm{nth}(\mathrm{cdr}(\mathit{move}),\ \mathit{st}) \simeq \mathbf{nil})$
$\lor$ $(\mathrm{car}(\mathrm{nth}(\mathrm{car}(\mathit{move}),\ \mathit{st})) < \mathrm{car}(\mathrm{nth}(\mathrm{cdr}(\mathit{move}),\ \mathit{st})))))$

Definition:
$\mathrm{step}(\mathit{move},\ \mathit{st})$
$=$ $\text{set-nth}(\mathrm{car}(\mathit{move}),$
$\text{set-nth}(\mathrm{cdr}(\mathit{move}),$
$\mathit{st},$
$\mathrm{cons}(\mathrm{car}(\mathrm{nth}(\mathrm{car}(\mathit{move}),\ \mathit{st})),\ \mathrm{nth}(\mathrm{cdr}(\mathit{move}),\ \mathit{st}))),$
$\mathrm{cdr}(\mathrm{nth}(\mathrm{car}(\mathit{move}),\ \mathit{st})))$

Event: Disable legal.

Definition:
$\mathrm{play}(\mathit{moves},\ \mathit{st})$
$=$ **if** $\mathrm{listp}(\mathit{moves})$
then if $\mathrm{legal}(\mathrm{car}(\mathit{moves}),\ \mathit{st})$
then $\mathrm{play}(\mathrm{cdr}(\mathit{moves}),\ \mathrm{step}(\mathrm{car}(\mathit{moves}),\ \mathit{st}))$
else f endif
else st **endif**

EVENT: Enable legal.

DEFINITION:
tower (n)
$=$ **if** $n \simeq 0$ **then nil**
else append (tower $(n-1)$, list (n)) **endif**

DEFINITION:
permp (i, j, k)
$= ((i \in$ '(0 1 2)$)$
$\land \quad (j \in$ '(0 1 2)$)$
$\land \quad (k \in$ '(0 1 2)$)$
$\land \quad (i \neq j)$
$\land \quad (i \neq k)$
$\land \quad (j \neq k))$

DEFINITION:
statep (s) = (listp (s) $\land$ listp (cdr (s)) $\land$ listp (cddr (s)))

DEFINITION:
slessp (n, s)
$=$ (**if** nth (0, s) $\simeq$ **nil then t**
else $n <$ car (nth (0, s)) **endif**
$\land$ **if** nth (1, s) $\simeq$ **nil then t**
else $n <$ car (nth (1, s)) **endif**
$\land$ **if** nth (2, s) $\simeq$ **nil then t**
else $n <$ car (nth (2, s)) **endif**)

```
; We wish to prove:

;(equal (play (hanoi n) (list (tower n) nil nil))
;       (list nil (tower n) nil))

; i.e.

;(equal (play (h n 0 1 2) (list (tower n) nil nil))
;       (list nil (tower n) nil))

; But to do so we go for a more general fact:

; (implies
;   (and (statep s)
;        (slessp n s)
;        (permp i j k))
```

```
;   (and (play (h n i j k)
;              (set-nth i s
;                       (append (tower n) (nth i s))))
;        (equal (play (h n i j k)
;                     (set-nth i s
;                              (append (tower n) (nth i s))))
;               (set-nth j s
;                        (append (tower n) (nth j s))))))

; The proof should go roughly by induction along the definition of
; h, but with s instantiated appropriately.
```

DEFINITION:
main-ind $(i,\ j,\ k,\ n,\ s)$
$=$ **if** $n \simeq 0$ **then nil**
else main-ind $(i,\ k,\ j,\ n-1,$ set-nth $(i,\ s,$ cons $(n,$ nth $(i,\ s))))$
$\wedge$ main-ind $(k,\ j,\ i,\ n-1,$ set-nth $(j,\ s,$ cons $(n,$ nth $(j,\ s))))$ **endif**

```
; The following is used to conclude the proof of the base case.
```

THEOREM: set-nth-nth
$((i \in$ '(0 1 2)$) \wedge$ statep $(s)) \rightarrow ($set-nth $(i,\ s,$ nth $(i,\ s)) = s)$

DEFINITION:
hyps $(i,\ j,\ k,\ n,\ s) = ($statep $(s) \wedge$ slessp $(n,\ s) \wedge$ permp $(i,\ j,\ k))$

DEFINITION:
conc $(i,\ j,\ k,\ n,\ s)$
$=$ (play (h $(n,\ i,\ j,\ k)$, set-nth $(i,\ s,$ append (tower (n), nth $(i,\ s))))$
$\wedge$ (play (h $(n,\ i,\ j,\ k)$, set-nth $(i,\ s,$ append (tower (n), nth $(i,\ s))))$
$=$ set-nth $(j,\ s,$ append (tower (n), nth $(j,\ s)))))$

DEFINITION:
main-thm $(i,\ j,\ k,\ n,\ s) = ($hyps $(i,\ j,\ k,\ n,\ s) \rightarrow$ conc $(i,\ j,\ k,\ n,\ s))$

THEOREM: base-case
$(n \simeq 0) \rightarrow$ main-thm $(i,\ j,\ k,\ n,\ s)$

THEOREM: nth-of-nlistp
$(\neg$ listp $(x)) \rightarrow ($nth $(n,\ x) = 0)$

THEOREM: play-append
play (append $(x,\ y),\ s) =$ play $(y,$ play $(x,\ s))$

Theorem: statep-set-nth
$\text{statep}(s) \rightarrow \text{statep}(\text{set-nth}(i,\ s,\ \text{cons}(n,\ \text{nth}(i,\ s))))$

```
; Got to here, and then aborted out as follows:
```

Theorem: nth-of-set-nth
$(\text{statep}(s) \land (i \in \texttt{'(0 1 2)}) \land (j \in \texttt{'(0 1 2)}))$
$\rightarrow \quad (\text{nth}(i,\ \text{set-nth}(j,\ s,\ v))$
$\quad = \quad \textbf{if } i = j \textbf{ then } v$
$\quad\quad \textbf{else } \text{nth}(i,\ s) \textbf{ endif})$

Definition:
$\text{statep-tower-ind}(i,\ n,\ s)$
$= \quad \textbf{if } n \simeq \texttt{0} \textbf{ then t}$
$\quad \textbf{else } \text{statep-tower-ind}(i,\ n-1,\ \text{set-nth}(i,\ s,\ \text{cons}(n,\ \text{nth}(i,\ s)))) \textbf{ endif}$

Theorem: append-assoc
$\text{append}(\text{append}(x,\ y),\ z) = \text{append}(x,\ \text{append}(y,\ z))$

Theorem: set-nth-twice
$\text{set-nth}(i,\ \text{set-nth}(i,\ s,\ v1),\ v2) = \text{set-nth}(i,\ s,\ v2)$

Theorem: slessp-preserved
$((n \in \mathbf{N}) \land (n \neq \texttt{0}) \land \text{statep}(s) \land (i \in \texttt{'(0 1 2)}) \land \text{slessp}(n,\ s))$
$\rightarrow \quad \text{slessp}(n-1,\ \text{set-nth}(i,\ s,\ \text{cons}(n,\ \text{nth}(i,\ s))))$

Theorem: statep-tower
$((n \in \mathbf{N}) \land (n \neq \texttt{0}) \land \text{statep}(s) \land \text{slessp}(n,\ s) \land (i \in \texttt{'(0 1 2)}))$
$\rightarrow \quad \text{statep}(\text{set-nth}(i,\ s,\ \text{append}(\text{tower}(n),\ \text{nth}(i,\ s))))$

Theorem: play-is-non-f-bookmark
t

Theorem: permp-preserved-1
$\text{permp}(i,\ j,\ k) \rightarrow \text{permp}(i,\ k,\ j)$

Event: Disable permp-preserved-1.

Theorem: used-first-ind-hyp
t

Theorem: set-nth-different-twice
$((i \in \texttt{'(0 1 2)}) \land (j \in \texttt{'(0 1 2)}) \land (i \neq j) \land \text{statep}(s))$
$\rightarrow \quad (\text{set-nth}(i,\ \text{set-nth}(j,\ s,\ v),\ w) = \text{set-nth}(j,\ \text{set-nth}(i,\ s,\ w),\ v))$

THEOREM: slessp-sub1
$\text{slessp}(n, s) \rightarrow \text{slessp}(n - 1, s)$

THEOREM: permp-preserved-2
$\text{permp}(i, j, k) \rightarrow \text{permp}(k, j, i)$

EVENT: Disable permp-preserved-2.

THEOREM: statep-preserved-by-set-nth
$(\text{statep}(s) \land (i \in \texttt{'(0 1 2)})) \rightarrow \text{statep}(\text{set-nth}(i, s, v))$

THEOREM: all-but-legal
t

THEOREM: ind-step
$$
\begin{array}{ll}
((n \not\simeq \texttt{0}) & \\
\ \land & \text{main-thm}(i, k, j, n - 1, \text{set-nth}(i, s, \text{cons}(n, \text{nth}(i, s)))) \\
\ \land & \text{main-thm}(k, j, i, n - 1, \text{set-nth}(j, s, \text{cons}(n, \text{nth}(j, s))))) \\
\rightarrow & \text{main-thm}(i, j, k, n, s)
\end{array}
$$

THEOREM: main-theorem-abbreviated
$\text{main-thm}(i, j, k, n, s)$

THEOREM: main-theorem
$$
\begin{array}{ll}
(\text{statep}(s) \land \text{slessp}(n, s) \land \text{permp}(i, j, k)) & \\
\rightarrow & (\text{play}(\text{h}(n, i, j, k), \text{set-nth}(i, s, \text{append}(\text{tower}(n), \text{nth}(i, s)))) \\
& \ = \ \text{set-nth}(j, s, \text{append}(\text{tower}(n), \text{nth}(j, s))))
\end{array}
$$

THEOREM: correctness-of-hanoi
$\text{play}(\text{hanoi}(n), \text{cons}(\text{tower}(n), \texttt{'(nil nil)}))$
$= \ \text{cons}(\textbf{nil}, \text{cons}(\text{tower}(n), \texttt{'(nil)}))$

Index