EVENT: Start with the library "c5".

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                              ;;
;;                        EXACT-TIME LEMMA SIGNAL CASE                          ;;
;;                                                                              ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

EVENT: Enable labelledp.

THEOREM: instrs-unlabel
(unlabel (cons ('push-constant, *y*)) = cons ('push-constant, *y*))
∧ (unlabel (cons ('pop-global, *y*)) = cons ('pop-global, *y*))
∧ (unlabel (cons ('jump, *y*)) = cons ('jump, *y*))

THEOREM: signal-meaning-r-2
(car (*stmt*) = 'signal-mg)
→ (mg-meaning-r (*stmt*, *proc-list*, *mg-state*, *n*, *sizes*)
= **if** *n* ≃ 0 **then** signal-system-error (*mg-state*, 'timed-out)
elseif ¬ normal (*mg-state*) **then** *mg-state*
elseif resources-inadequatep (*stmt*, *proc-list*, *sizes*)
then signal-system-error (*mg-state*, 'resource-error)
else set-condition (*mg-state*, signalled-condition (*stmt*)) **endif**)

THEOREM: signal-translation2
(car (*stmt*) = 'signal-mg)
→ (translate (*cinfo*, *cond-list*, *stmt*, *proc-list*)
= make-cinfo (append (code (*cinfo*),
list (list ('push-constant,
mg-cond-to-p-nat (signalled-condition (*stmt*),
cond-list)),
list ('pop-global, 'c-c),
list ('jump,
fetch-label (signalled-condition (*stmt*),
label-alist (*cinfo*))))),
label-alist (*cinfo*),
label-cnt (*cinfo*)))

THEOREM: exact-time-lemma-signal-case-steps-1-3
((*n* ≄ 0)
∧ (¬ resources-inadequatep (*stmt*,
proc-list,
list (length (*temp-stk*),

p-ctrl-stk-size (*ctrl-stk*))))
$\wedge$ (car (*stmt*) $\neq$ 'no-op-mg)
$\wedge$ (car (*stmt*) = 'signal-mg)
$\wedge$ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
$\wedge$ ok-mg-def-plistp (*proc-list*)
$\wedge$ ok-translation-parameters (*cinfo*, *t-cond-list*, *stmt*, *proc-list*, *code2*)
$\wedge$ ok-mg-statep (*mg-state*, *r-cond-list*)
$\wedge$ cond-subsetp (*r-cond-list*, *t-cond-list*)
$\wedge$ (code (translate-def-body (assoc (*subr*, *proc-list*), *proc-list*))
= append (code (translate (*cinfo*, *t-cond-list*, *stmt*, *proc-list*)),
code2))
$\wedge$ user-defined-procp (*subr*, *proc-list*)
$\wedge$ plistp (*temp-stk*)
$\wedge$ listp (*ctrl-stk*)
$\wedge$ mg-vars-list-ok-in-p-state (mg-alist (*mg-state*),
bindings (top (*ctrl-stk*)),
temp-stk)
$\wedge$ no-p-aliasing (bindings (top (*ctrl-stk*)), mg-alist (*mg-state*))
$\wedge$ signatures-match (mg-alist (*mg-state*), *name-alist*)
$\wedge$ normal (*mg-state*)
$\wedge$ all-cars-unique (mg-alist (*mg-state*))
$\wedge$ ($\neg$ resource-errorp (mg-meaning-r (*stmt*,
proc-list,
mg-state,
n,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*))))))
$\rightarrow$ (p-step (p-step (p-step (map-down (*mg-state*,
proc-list,
ctrl-stk,
temp-stk,
tag ('pc,
cons (*subr*, length (code (*cinfo*)))),
t-cond-list))))
= p-state (tag ('pc,
cons (*subr*,
if normal (mg-meaning-r (*stmt*,
proc-list,
mg-state,
n,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*))))
then length (code (translate (*cinfo*,
t-cond-list,

stmt,
proc-list)))
else find-label (fetch-label (cc (mg-meaning-r (*stmt*,
proc-list,
mg-state,
n,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*)))),
label-alist (translate (*cinfo*,
t-cond-list,
stmt,
proc-list))),
append (code (translate (*cinfo*,
t-cond-list,
stmt,
proc-list)),
code2)) **endif**)),
ctrl-stk,
map-down-values (mg-alist (mg-meaning-r (*stmt*,
proc-list,
mg-state,
n,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*)))),
bindings (top (*ctrl-stk*)),
temp-stk),
translate-proc-list (*proc-list*),
list (list ('c-c,
mg-cond-to-p-nat (cc (mg-meaning-r (*stmt*,
proc-list,
mg-state,
n,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*)))),
t-cond-list))),
MG-MAX-CTRL-STK-SIZE,
MG-MAX-TEMP-STK-SIZE,
MG-WORD-SIZE,
'run))

EVENT: Disable exact-time-lemma-signal-case-steps-1-3.

THEOREM: exact-time-lemma-signal-case
((*n* ≠ 0)

$$
\begin{array}{ll}
\wedge & (\neg\ \text{resources-inadequatep}\,(\textit{stmt},\\
& \qquad\qquad\qquad\qquad \textit{proc-list},\\
& \qquad\qquad\qquad\qquad \text{list}\,(\text{length}\,(\textit{temp-stk}),\\
& \qquad\qquad\qquad\qquad\quad \text{p-ctrl-stk-size}\,(\textit{ctrl-stk}))))\\
\wedge & (\text{car}\,(\textit{stmt}) \neq \texttt{'no-op-mg})\\
\wedge & (\text{car}\,(\textit{stmt}) = \texttt{'signal-mg})\\
\wedge & \text{ok-mg-statement}\,(\textit{stmt},\ \textit{r-cond-list},\ \textit{name-alist},\ \textit{proc-list})\\
\wedge & \text{ok-mg-def-plistp}\,(\textit{proc-list})\\
\wedge & \text{ok-translation-parameters}\,(\textit{cinfo},\ \textit{t-cond-list},\ \textit{stmt},\ \textit{proc-list},\ \textit{code2})\\
\wedge & \text{ok-mg-statep}\,(\textit{mg-state},\ \textit{r-cond-list})\\
\wedge & \text{cond-subsetp}\,(\textit{r-cond-list},\ \textit{t-cond-list})\\
\wedge & (\text{code}\,(\text{translate-def-body}\,(\text{assoc}\,(\textit{subr},\ \textit{proc-list}),\ \textit{proc-list}))\\
& \quad = \quad \text{append}\,(\text{code}\,(\text{translate}\,(\textit{cinfo},\ \textit{t-cond-list},\ \textit{stmt},\ \textit{proc-list})),\\
& \qquad\qquad \textit{code2}))\\
\wedge & \text{user-defined-procp}\,(\textit{subr},\ \textit{proc-list})\\
\wedge & \text{plistp}\,(\textit{temp-stk})\\
\wedge & \text{listp}\,(\textit{ctrl-stk})\\
\wedge & \text{mg-vars-list-ok-in-p-state}\,(\text{mg-alist}\,(\textit{mg-state}),\\
& \qquad\qquad\qquad\qquad \text{bindings}\,(\text{top}\,(\textit{ctrl-stk})),\\
& \qquad\qquad\qquad\qquad \textit{temp-stk})\\
\wedge & \text{no-p-aliasing}\,(\text{bindings}\,(\text{top}\,(\textit{ctrl-stk})),\ \text{mg-alist}\,(\textit{mg-state}))\\
\wedge & \text{signatures-match}\,(\text{mg-alist}\,(\textit{mg-state}),\ \textit{name-alist})\\
\wedge & \text{normal}\,(\textit{mg-state})\\
\wedge & \text{all-cars-unique}\,(\text{mg-alist}\,(\textit{mg-state}))\\
\wedge & (\neg\ \text{resource-errorp}\,(\text{mg-meaning-r}\,(\textit{stmt},\\
& \qquad\qquad\qquad\qquad\qquad\qquad \textit{proc-list},\\
& \qquad\qquad\qquad\qquad\qquad\qquad \textit{mg-state},\\
& \qquad\qquad\qquad\qquad\qquad\qquad n,\\
& \qquad\qquad\qquad\qquad\qquad\qquad \text{list}\,(\text{length}\,(\textit{temp-stk}),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{p-ctrl-stk-size}\,(\textit{ctrl-stk})))))\\
\rightarrow & (\text{p}\,(\text{map-down}\,(\textit{mg-state},\\
& \qquad\qquad\quad \textit{proc-list},\\
& \qquad\qquad\quad \textit{ctrl-stk},\\
& \qquad\qquad\quad \textit{temp-stk},\\
& \qquad\qquad\quad \text{tag}\,(\texttt{'pc},\ \text{cons}\,(\textit{subr},\ \text{length}\,(\text{code}\,(\textit{cinfo})))),\\
& \qquad\qquad\quad \textit{t-cond-list}),\\
& \qquad \text{clock}\,(\textit{stmt},\ \textit{proc-list},\ \textit{mg-state},\ n))\\
& \quad = \quad \text{p-state}\,(\text{tag}\,(\texttt{'pc},\\
& \qquad\qquad\qquad\quad \text{cons}\,(\textit{subr},\\
& \qquad\qquad\qquad\qquad\quad \textbf{if}\ \text{normal}\,(\text{mg-meaning-r}\,(\textit{stmt},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \textit{proc-list},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \textit{mg-state},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad n,\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{list}\,(\text{length}\,(\textit{temp-stk}),
\end{array}
$$

```
                                                    p-ctrl-stk-size (ctrl-stk))))
            then length (code (translate (cinfo,
                                          t-cond-list,
                                          stmt,
                                          proc-list)))
            else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                            proc-list,
                                                            mg-state,
                                                            n,
                                                            list (length (temp-stk),
                                                                  p-ctrl-stk-size (ctrl-stk)))),
                                              label-alist (translate (cinfo,
                                                                      t-cond-list,
                                                                      stmt,
                                                                      proc-list))),
                                    append (code (translate (cinfo,
                                                             t-cond-list,
                                                             stmt,
                                                             proc-list)),
                                            code2)) endif)),
        ctrl-stk,
        map-down-values (mg-alist (mg-meaning-r (stmt,
                                                 proc-list,
                                                 mg-state,
                                                 n,
                                                 list (length (temp-stk),
                                                       p-ctrl-stk-size (ctrl-stk)))),
                                   bindings (top (ctrl-stk)),
                                   temp-stk),
        translate-proc-list (proc-list),
        list (list ('c-c,
                    mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                        proc-list,
                                                        mg-state,
                                                        n,
                                                        list (length (temp-stk),
                                                              p-ctrl-stk-size (ctrl-stk)))),
                                         t-cond-list))),
        MG-MAX-CTRL-STK-SIZE,
        MG-MAX-TEMP-STK-SIZE,
        MG-WORD-SIZE,
        'run))
```

EVENT: Disable exact-time-lemma-signal-case.

EVENT: Make the library `"c-signal"`.

Index