

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))))
^ (car (stmt) ≠ 'no-op-mg)
^ (car (stmt) = 'signal-mg)
^ ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
^ ok-mg-def-plistp (proc-list)
^ ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
^ ok-mg-statep (mg-state, r-cond-list)
^ cond-subsetp (r-cond-list, t-cond-list)
^ (code (translate-def-body (assoc (subr, proc-list), proc-list))
    = append (code (translate (cinfo, t-cond-list, stmt, proc-list)),
              code2))
^ user-defined-procp (subr, proc-list)
^ plistp (temp-stk)
^ listp (ctrl-stk)
^ mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                             bindings (top (ctrl-stk)),
                             temp-stk)
^ no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
^ signatures-match (mg-alist (mg-state), name-alist)
^ normal (mg-state)
^ all-cars-unique (mg-alist (mg-state))
^ (¬ resource-errorp (mg-meaning-r (stmt,
                                   proc-list,
                                   mg-state,
                                   n,
                                   list (length (temp-stk),
                                           p-ctrl-stk-size (ctrl-stk)))))
→ (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
 $((n \neq 0)$

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

```

                                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

all-cars-unique, 2, 4
bindings, 2–5
cc, 3, 5
clock, 4
code, 1–5
cond-subsetp, 2, 4
exact-time-lemma-signal-case, 3
exact-time-lemma-signal-case-step
 s-1-3, 1
fetch-label, 1, 3, 5
find-label, 3, 5
instrs-unlabel, 1
label-alist, 1, 3, 5
label-cnt, 1
length, 1–5
make-cinfo, 1
map-down, 2, 4
map-down-values, 3, 5
mg-alist, 2–5
mg-cond-to-p-nat, 1, 3, 5
mg-max-ctrl-stk-size, 3, 5
mg-max-temp-stk-size, 3, 5
mg-meaning-r, 1–5
mg-vars-list-ok-in-p-state, 2, 4
mg-word-size, 3, 5
no-p-aliasing, 2, 4
normal, 1, 2, 4, 5
ok-mg-def-plistp, 2, 4
ok-mg-statement, 2, 4
ok-mg-statep, 2, 4
ok-translation-parameters, 2, 4
p, 4
p-ctrl-stk-size, 2–5
p-state, 3, 5
p-step, 2
plistp, 2, 4
resource-errorp, 2, 4
resources-inadequatep, 1, 2, 4
set-condition, 1
signal-meaning-r-2, 1
signal-system-error, 1
signal-translation2, 1
signalled-condition, 1
signatures-match, 2, 4
tag, 2–5
top, 2–5
translate, 1–5
translate-def-body, 2, 4
translate-proc-list, 3, 5
unlabel, 1
user-defined-procp, 2, 4