```
THEOREM: fetch-label-0-case-2
 \operatorname{cdr}\left(\operatorname{assoc}\left(\operatorname{cc},\operatorname{cons}\left(\mathsf{'}\left(\operatorname{routineerror}\right.0\right),\operatorname{make-label-alist}\left(\operatorname{lst},0\right)\right)\right)\right)
EVENT: Disable fetch-label-0-case-2.
THEOREM: mg-cond-to-p-nat-p-objectp-type-nat
 ((length (cond-list) < (((exp(2, MG-WORD-SIZE) - 1) - 1))
    \land (p-word-size (state) = MG-WORD-SIZE))
              p-objectp-type ('nat, mg-cond-to-p-nat (cc, cond-list), state)
EVENT: Disable mg-cond-to-p-nat-p-objectp-type-nat.
Theorem: mg-cond-to-p-nat-index-lessp
 (length(lst) < n)
 \rightarrow ((untag (mg-cond-to-p-nat (c, lst)) < (1 + (1 + n))) = \mathbf{t})
EVENT: Disable mg-cond-to-p-nat-index-lessp.
THEOREM: set-alist-value-map-down-values-length-doesnt-shrink
 (\text{mg-vars-list-ok-in-p-state}(x, bindings, temp-stk) \land \text{mg-alistp}(x))
 \rightarrow ((length (map-down-values (set-alist-value (name, value, x),
                                                                                          bindings,
                                                                                          temp-stk))
                    < length (map-down-values (x, bindings, temp-stk)))
EVENT: Disable set-alist-value-map-down-values-length-doesnt-shrink.
\label{theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theorem:theor
 (\operatorname{car}(x) \not\in \operatorname{listcars}(y))
 \rightarrow (formal-types-preserved (y, \cos(x, z)) = formal-types-preserved (y, z))
Theorem: not-simple-identifiers-array-identifiers
 (definedp(x, alist))
    \land mg-alistp (alist)
    \land \quad (\neg \text{ simple-identifierp } (x, alist)))
  \rightarrow array-identifierp (x, alist)
THEOREM: mg-meaning-preserves-signatures-match2
 plistp (alist)
```

EVENT: Start with the library "c-predefined4".

```
signatures-match (alist,
                   mg-alist (mg-meaning (stmt,
                                      proc-list,
                                      mg-state (cc, alist, psw),
                                      n)))
PROCEDURE CALLS
;;
                                                                             ;;
;;
                                                                             ;;
EVENT: Disable make-cond-list.
EVENT: Disable make-call-var-alist.
EVENT: Disable mg-var-ok-in-p-state.
EVENT: Enable proc-call-code.
Theorem: proc-call-meaning-r-2
(car(stmt) = 'proc-call-mg)
    (mg-meaning-r(stmt, proc-list, mg-state, n, sizes)
     = if n \simeq 0 then signal-system-error (mg-state, 'timed-out)
         elseif \neg normal (mg-state) then mg-state
        elseif resources-inadequatep (stmt, proc-list, sizes)
        then signal-system-error (mg-state, 'resource-error)
         else map-call-effects (mg-meaning-r (def-body (fetch-called-def (stmt,
                                                             proc-list)),
                                       proc-list,
                                       make-call-environment (mg-state,
                                                           stmt,
                                                           fetch-called-def (stmt,
                                                                         proc-list)),
                                       n-1,
                                       list (t-size (sizes)
                                           + data-length (def-locals (fetch-called-def (stmt,
                                                                                proc-list))),
                                           c-size (sizes)
                                           + (2
                                               + length (def-locals (fetch-called-def (stmt,
```

```
proc-list)))
                                                                length (def-formals (fetch-called-def (stmt,
                                                                                                        proc-list))))
                                  fetch-called-def (stmt, proc-list),
                                  stmt,
                                  mg-state) endif)
Theorem: call-translation-2
(car(stmt) = 'proc-call-mg)
 \rightarrow (translate (cinfo, cond-list, stmt, proc-list)
      = make-cinfo (append (code (cinfo),
                                 proc-call-code (cinfo,
                                                 stmt,
                                                 cond-list,
                                                 def-locals (fetch-called-def(stmt,
                                                                              proc-list)),
                                                 length (def-cond-locals (fetch-called-def (stmt,
                                                                                            proc-list))))),
                        label-alist (cinfo),
                        label-cnt (cinfo)
                        + (1 + (1 + length (call-conds (stmt))))))
Theorem: locals-pointers-bigger0
all-cars-unique (locals)
     all-pointers-bigger (collect-pointers (map-call-locals (locals, n), locals),
                          n)
Theorem: no-p-aliasing-locals
((n \in \mathbf{N}) \land \text{all-cars-unique}(locals))
     no-p-aliasing (map-call-locals (locals, n), locals)
Theorem: map-call-formals-all-pointers-smaller3
(ok-actual-params-list (actuals, mq-vars)
     data-param-lists-match (actuals, formals, mg-vars)
     all-cars-unique (formals)
 \land mg-alistp (mg-vars)
 \land mg-vars-list-ok-in-p-state (mg-vars, bindings, temp-stk))
     all-pointers-smaller (collect-pointers (map-call-formals (formals,
                                                                actuals,
                                                                bindings),
                                             make-call-param-alist (formals,
                                                                     actuals,
                                                                     mq-vars)),
                           length(temp-stk))
```

;; These got lost somewhere

```
THEOREM: array-alist-element-lengths-match
(mg-alistp (mq-vars))
 \land definedp (x, mq\text{-}vars)
 \land (¬ simple-mg-type-refp (cadr (assoc (x, mg-vars)))))
    (length (caddr (assoc (x, mg-vars))))
       = \operatorname{array-length} \left( \operatorname{cadr} \left( \operatorname{assoc} \left( x, \, mg\text{-}vars \right) \right) \right)
EVENT: Disable array-alist-element-lengths-match.
EVENT: Enable no-p-aliasing.
Theorem: extra-bindings-dont-affect-no-p-aliasing
no-duplicates (listcars (append (bindings1, lst)))
    (no-p-aliasing (append (bindings1, bindings2), lst)
       = no-p-aliasing (bindings2, lst))
Theorem: extra-bindings-dont-affect-no-p-aliasing2
no-duplicates (listcars (append (bindings2, lst)))
      (no-p-aliasing (append (bindings1, bindings2), lst)
       = no-p-aliasing (bindings1, lst)
THEOREM: actual-pointers-distinct
(ok-actual-params-list (actuals, mq-vars)
 \land defined-identifierp (x, mg\text{-}vars)
     data-param-lists-match (actuals, formals, mg-vars)
 \wedge
 ∧ ok-mg-formal-data-params-plistp (formals)
 \land all-cars-unique (mq-vars)
 \land no-duplicates (cons (x, actuals))
 \land no-p-aliasing (bindings, mg-vars)
 \land all-cars-unique (formals)
 \land mg-alistp (mg-vars))
 \rightarrow (untag (cdr (assoc (x, bindings)))
       ∉ collect-pointers (map-call-formals (formals, actuals, bindings),
                             make-call-param-alist (formals,
                                                       actuals,
                                                       mq-vars)))
Theorem: actual-pointers-distinct2
(ok-actual-params-list (actuals, mg-vars)
 \land defined-identifierp (x, mq\text{-}vars)
     data-param-lists-match (actuals, formals, mg-vars)
```

```
ok-mg-formal-data-params-plistp (formals)
 \land mg-vars-list-ok-in-p-state (mg-vars, bindings, temp-stk)
 \land all-cars-unique (mq-vars)
 \land no-duplicates (cons (x, actuals))
 \land no-p-aliasing (bindings, mq-vars)
 \land all-cars-unique (formals)
 \land mg-alistp (mg-vars)
     (\neg \text{ simple-mg-type-refp} (\text{cadr} (\text{assoc}(x, mg-vars)))))
     disjoint (n-successive-pointers (cdr (assoc(x, bindings))),
                                        \operatorname{array-length}\left(\operatorname{cadr}\left(\operatorname{assoc}\left(x, mq\text{-}vars\right)\right)\right)\right)
                collect-pointers (map-call-formals (formals, actuals, bindings),
                                  make-call-param-alist (formals,
                                                            mg-vars)))
THEOREM: no-p-aliasing-formals
(ok-actual-params-list (actuals, mg-vars)
 ∧ data-param-lists-match (actuals, formals, mg-vars)
 \land ok-mg-formal-data-params-plistp (formals)
 \land mg-vars-list-ok-in-p-state (mg-vars, bindings, temp-stk)
 \land all-cars-unique (mg-vars)
 \land no-duplicates (actuals)
 \land no-p-aliasing (bindings, mq-vars)
 \land all-cars-unique (formals)
 \land mg-alistp (mg-vars))
      no-p-aliasing (map-call-formals (formals, actuals, bindings),
                      make-call-param-alist (formals, actuals, mg-vars))
Theorem: actual-params-list-ok-in-mg-alist
((\operatorname{car}(stmt) = \operatorname{"proc-call-mg}))
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-statep (mg-state, r-cond-list)
     signatures-match (mg-alist (mg-state), name-alist))
      ok-actual-params-list (call-actuals (stmt), mg-alist (mg-state))
Theorem: data-param-lists-match-in-mg-alist
((\operatorname{car}(stmt) = \operatorname{"proc-call-mg"})
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land signatures-match (mg-alist (mg-state), name-alist))
     data-param-lists-match (call-actuals (stmt)),
                                  def-formals (fetch-called-def (stmt, proc-list)),
                                  mg-alist (mq-state))
```

Theorem: call-local-names-unique

```
((\operatorname{car}(stmt) = \operatorname{'proc-call-mg})
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list))
 \rightarrow no-duplicates (append (listcars (def-locals (fetch-called-def (stmt),
                                                                        proc-list))),
                                listcars (def-formals (fetch-called-def (stmt,
                                                                          proc-list))))))
THEOREM: no-p-aliasing-in-call-environment
((\operatorname{car}(stmt) = \texttt{'proc-call-mg'})
 \land \quad \text{ok-mg-statement} \ (\textit{stmt}, \ \textit{r-cond-list}, \ \textit{name-alist}, \ \textit{proc-list})
 \land ok-mg-def-plistp (proc-list)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land user-defined-procp (subr, proc-list)
 \land plistp (temp-stk)
 \land listp (ctrl-stk)
     mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                    bindings (top (ctrl-stk)),
                                    temp-stk)
 \land no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
 \land signatures-match (mg-alist (mg-state), name-alist)
 \land all-cars-unique (mg-alist (mg-state)))
 \rightarrow no-p-aliasing (make-frame-alist (fetch-called-def (stmt, proc-list),
                                          stmt.
                                          ctrl-stk,
                                          temp-stk),
                      make-call-var-alist (mg-alist (mg-state),
                                            stmt,
                                            fetch-called-def(stmt, proc-list)))
Theorem: call-exact-time-translation-parameters-ok
((car(stmt) = 'proc-call-mg))
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
     ok-mg-def-plistp (proc-list)
 \land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
 \land ok-mg-statep (mg-state, r-cond-list))
     ok-translation-parameters (make-cinfo (nil,
                                                   cons('(routineerror . 0),
                                                         make-label-alist (make-cond-list (fetch-called-def (stmt,
                                                                                                                 proc-list))
                                                                            0)),
                                                   1).
                                     make-cond-list (fetch-called-def (stmt,
```

proc-list)),

```
def-body (fetch-called-def (stmt, proc-list)),
                                  proc-list,
                                 cons('(dl 0 nil (no-op)),
                                       cons (list ('pop*,
                                                  data-length (def-locals (fetch-called-def (stmt,
                                                                                           proc-list)))),
                                              '((ret))))
THEOREM: call-body-rewrite
code (translate-def-body (assoc (call-name (stmt), proc-list), proc-list))
    append (code (translate (make-cinfo (nil,
                                           cons('(routineerror . 0),
                                                 make-label-alist (make-cond-list (fetch-called-def (stmt,
                                                                                                    proc-list)),
                                                                   0)),
                                           1),
                              make-cond-list (fetch-called-def (stmt, proc-list)),
                              def-body (fetch-called-def (stmt, proc-list)),
                              proc-list)),
              cons('(dl 0 nil (no-op)),
                    cons (list ('pop*,
                              data-length (def-locals (fetch-called-def (stmt,
                                                                        proc-list)))),
                          '((ret))))
DEFINITION:
mg-locals-list-ok-induction-hint (locals, temp-stk)
   if locals \simeq nil then t
    elseif simple-mg-type-refp (cadar (locals))
    then mg-locals-list-ok-induction-hint (cdr (locals),
                                            cons (mg-to-p-simple-literal (caddar (locals)),
                                                  temp-stk))
    else mg-locals-list-ok-induction-hint (cdr (locals),
                                           append (reverse (mg-to-p-simple-literal-list (caddar (locals))),
                                                    temp-stk)) endif
THEOREM: definedp-caar
defined (x, \cos(\cos(x, y), z))
THEOREM: tag-length-plistp-2
length-plistp (tag (x, y), 2)
Theorem: mg-locals-list-ok-in-call-environment
(all-cars-unique (locals) \land ok-mg-local-data-plistp (locals))
\rightarrow mg-vars-list-ok-in-p-state (locals,
```

```
append (map-call-locals (locals,
                                                            length(temp-stk)),
                                  append (reverse (mg-to-p-local-values (locals)),
                                           temp-stk)
Theorem: simple-formal-ok-for-actual2
(definedp (actual, mg-alist))
     data-params-match (actual, formal, mg-alist)
     mg-vars-list-ok-in-p-state (mg-alist, bindings, temp-stk)
 \land simple-mg-type-refp (cadr (formal)))
     ok-temp-stk-index (cdr (assoc (actual, bindings)), temp-stk)
mg-formals-list-ok-induction-hint (formals, actuals)
    if formals \simeq nil then t
    else mg-formals-list-ok-induction-hint (cdr (formals),
                                              cdr(actuals)) endif
Theorem: array-formal-ok-for-actual
(defined-identifierp (actual, mg-alist)
 \land data-params-match (actual, formal, mg-alist)
 \land mg-vars-list-ok-in-p-state (mg-alist, bindings, temp-stk)
 \land (\neg simple-mg-type-refp (cadr (formal))))
 \rightarrow ok-temp-stk-array-index (cdr (assoc (actual, bindings)),
                                 append (local-values, temp-stk),
                                 \operatorname{array-length}\left(\operatorname{cadr}\left(formal\right)\right)\right)
Theorem: mg-formals-list-ok-in-call-environment0
(all-cars-unique (formals)
 \land ok-actual-params-list (actuals, mg-alist)
 \land data-param-lists-match (actuals, formals, mg-alist)
 \land mg-vars-list-ok-in-p-state (mg-alist, bindings, temp-stk))
 → mg-vars-list-ok-in-p-state (make-call-param-alist (formals,
                                                          actuals,
                                                          mq-alist),
                                  map-call-formals (formals, actuals, bindings),
                                  append (local-values, temp-stk))
Theorem: mg-formals-list-ok-in-call-environment1
(all-cars-unique (formals)
 \land mg-alistp (mg-alist)
 \land ok-actual-params-list (actuals, mg-alist)
 ∧ ok-mg-formal-data-params-plistp (formals)
 \land data-param-lists-match (actuals, formals, mg-alist)
```

```
mg-vars-list-ok-in-p-state (mg-alist, bindings, temp-stk))
     mg-vars-list-ok-in-p-state (make-call-param-alist (formals,
                                                           actuals,
                                                           mq-alist),
                                  map-call-formals (formals, actuals, bindings),
                                  append (local-values,
                                           map-down-values (mg-alist,
                                                               bindings,
                                                               temp-stk)))
THEOREM: mg-vars-list-ok-in-call-environment
((n \not\simeq 0)
 \land (¬ resources-inadequatep (stmt,
                                  proc-list,
                                  list (length (temp-stk),
                                       p-ctrl-stk-size (ctrl-stk))))
     (car(stmt) = 'proc-call-mg)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 \land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land cond-subsetp (r-cond-list, t-cond-list)
 \land (code (translate-def-body (assoc (subr, proc-list), proc-list))
          append (code (translate (cinfo, t-cond-list, stmt, proc-list)),
                     code2))
 \land user-defined-procp (subr, proc-list)
 \land plistp (temp-stk)
 \land listp (ctrl-stk)
 \land mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                  temp-stk)
 \land no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
     signatures-match (mg-alist (mg-state), name-alist)
 \wedge
 \land normal (mg-state)
 \land all-cars-unique (mg-alist (mg-state)))
 \rightarrow mg-vars-list-ok-in-p-state (make-call-var-alist (mg-alist (mg-state)),
                                                       stmt,
                                                       fetch-called-def (stmt,
                                                                         proc-list)),
                                  append (map-call-locals (def-locals (fetch-called-def (stmt,
                                                                                          proc-list)),
                                                             length(temp-stk)),
                                           map-call-formals (def-formals (fetch-called-def (stmt,
                                                                                              proc-list)),
```

```
call-actuals (stmt),
                                                                 bindings (top (ctrl-stk)))),
                                   append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                                            proc-list)))),
                                             map-down-values (mg-alist (mg-state),
                                                                  bindings (top (ctrl-stk)),
                                                                  temp-stk)))
THEOREM: proc-call-doesnt-halt
((\operatorname{car}(stmt) = \operatorname{'proc-call-mg})
 \land normal (mg\text{-}state)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 \land ok-mg-statep (mg-state, r-cond-list)
     mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                   bindings (top (ctrl-stk)),
                                    temp-stk)
      (\neg resource-errorp (mg-meaning-r (stmt,
                                             proc-list,
                                             mg-state,
                                             n,
                                             list (length (temp-stk),
                                                  p-ctrl-stk-size (ctrl-stk))))))
     (\neg \text{ resource-errorp } (\text{mg-meaning-r} (\text{def-body } (\text{fetch-called-def} (stmt,
                                                                          proc-list)),
                                            proc-list,
                                             make-call-environment (mg-state,
                                                                        stmt,
                                                                        fetch-called-def (stmt,
                                                                                          proc-list)),
                                             n-1,
                                            list (length (append (reverse (mg-to-p-local-values (def-locals (fetch-call
                                                                    map-down-values (mg-alist (mg-state),
                                                                                        bindings (top (ctrl-stk)),
                                                                                         temp-stk))),
                                                  p-ctrl-stk-size (cons (p-frame (make-frame-alist (fetch-called-def (s
                                                                                                       stmt,
                                                                                                       ctrl-stk.
                                                                                                       temp-stk),
                                                                                   tag('pc,
```

cons(subr,

1 + (length (code (cinfo)))

```
length (def-local
                                                                                                        length (call-actu
                                                                        ctrl-stk)))))))
Theorem: proc-call-doesnt-halt2
((\operatorname{car}(stmt) = \operatorname{'proc-call-mg})
 \land normal (mg-state)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 \land ok-mg-statep (mg-state, r-cond-list)
     mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                   bindings (top (ctrl-stk)),
                                    temp-stk)
     (\neg \text{ resource-errorp } (\text{mg-meaning-r } (stmt,
                                             proc-list,
                                             mg-state,
                                             n,
                                             list (length (temp-stk),
                                                  p-ctrl-stk-size (ctrl-stk))))))
      (¬ resource-errorp (mg-meaning-r (def-body (fetch-called-def (stmt,
                                                                          proc-list)),
                                             proc-list,
                                             make-call-environment (mg-state,
                                                                       fetch-called-def (stmt,
                                                                                          proc-list)),
                                             n-1,
                                            list (length (temp-stk)
                                                  + data-length (def-locals (fetch-called-def (stmt,
                                                                                                   proc-list))),
                                                 p-ctrl-stk-size (ctrl-stk)
                                                      length (def-locals (fetch-called-def (stmt,
                                                                                             proc-list)))
                                                      length (def-formals (fetch-called-def (stmt,
                                                                                               proc-list)))))))))
Theorem: proc-call-exact-time-hyps
((n \not\simeq 0)
 \land (\neg resources-inadequatep (stmt,
                                   proc-list,
```

data-length (def-

```
list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
    (car(stmt) = 'proc-call-mg)
   ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
\land ok-mg-def-plistp (proc-list)
\land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
\land 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)
\wedge
    listp (ctrl-stk)
\wedge
    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))
\land signatures-match (mg-alist (mg-state), name-alist)
\land normal (mq\text{-}state)
\land all-cars-unique (mg-alist (mg-state))
   (\neg \text{ resource-errorp (mg-meaning-r }) stmt,
                                         proc-list,
                                         mg-state,
                                         list (length (temp-stk),
                                              p-ctrl-stk-size(ctrl-stk))))))
    (ok-mg-statement (def-body (fetch-called-def (stmt, proc-list)),
                        make-cond-list (fetch-called-def (stmt, proc-list)),
                        make-name-alist (fetch-called-def (stmt, proc-list)),
                        proc-list)
        ok-translation-parameters (make-cinfo (nil,
                                                   cons('(routineerror
                                                            . 0).
                                                         make-label-alist (make-cond-list (fetch-called-def (stmt,
                                                                            0)),
                                                   1),
                                      make-cond-list (fetch-called-def (stmt,
                                                                         proc-list)),
                                      def-body (fetch-called-def (stmt,
                                                                   proc-list)),
                                      proc-list,
                                      cons('(dl 0 nil (no-op)),
```

```
cons (list ('pop*,
                                               data-length (def-locals (fetch-called-def (stmt,
                                                                                          proc-list)))),
                                          '((ret))))
ok-mg-statep (make-call-environment (mg-state,
                                          stmt,
                                          fetch-called-def (stmt,
                                                            proc-list)),
                {\it make-cond-list} \, ({\it fetch-called-def} \, ({\it stmt}, \, {\it proc-list})))
cond-subsetp (make-cond-list (fetch-called-def (stmt, proc-list)),
                make-cond-list (fetch-called-def (stmt, proc-list)))
(code (translate-def-body (assoc (call-name (stmt), proc-list),
                             proc-list))
      append (code (translate (make-cinfo (nil,
                                             cons('(routineerror
                                                       . 0),
                                                    make-label-alist (make-cond-list (fetch-called-def (str
                                                                      0)),
                                             1),
                                make-cond-list (fetch-called-def (stmt,
                                                                   proc-list)),
                                def-body (fetch-called-def (stmt,
                                                             proc-list)),
                                proc-list)),
               cons('(dl 0 nil (no-op)),
                     cons (list ('pop*,
                                data-length (def-locals (fetch-called-def (stmt,
                                                                           proc-list)))),
                            '((ret)))))
user-defined-procp (call-name (stmt), proc-list)
plistp (append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                              proc-list)))),
                 map-down-values (mg-alist (mg-state),
                                     bindings (top (ctrl-stk)),
                                     temp-stk)))
listp (cons (p-frame (make-frame-alist (fetch-called-def (stmt,
                                                           proc-list),
                                          stmt,
                                          ctrl-stk,
                                          temp-stk),
                      tag('pc,
                           \cos(subr,
                                 1 + (length (code (cinfo)))
```

```
data-length (def-locals (fetch-called-def (stmt,
                                                                                     proc-list)))
                                           length (def-locals (fetch-called-def (stmt,
                                                                                proc-list)))
                                           length (call-actuals (stmt)))))),
            ctrl-stk))
mg-vars-list-ok-in-p-state (mg-alist (make-call-environment (mg-state,
                                                                fetch-called-def (stmt,
                                                                                 proc-list))),
                            bindings (top (cons (p-frame (make-frame-alist (fetch-called-def (stmt,
                                                                                                proc-list),
                                                                               stmt,
                                                                               ctrl-stk,
                                                                               temp-stk),
                                                           tag('pc,
                                                                cons(subr,
                                                                      1 + (length (code (cinfo)))
                                                                                data-length (def-locals (fe
                                                                                length (def-locals (fetch-
                                                                                length (call-actuals (stmt
                                                  ctrl-stk))),
                            append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                                  proc-list
                                      map-down-values (mg-alist (mg-state),
                                                         bindings (top (ctrl-stk)),
                                                          temp-stk)))
no-p-aliasing (bindings (top (cons (p-frame (make-frame-alist (fetch-called-def (stmt,
                                                                                   proc-list),
                                                                  stmt,
                                                                  ctrl-stk,
                                                                  temp-stk),
                                              tag('pc,
                                                   \cos(subr,
                                                         1 + (length (code (cinfo)))
                                                                   data-length (def-locals (fetch-called-d
                                                                   length (def-locals (fetch-called-def (st
                                                                   length (call-actuals (stmt)))))),
                                     ctrl-stk))),
               mg-alist (make-call-environment (mg-state,
```

```
stmt,
                                                   fetch-called-def (stmt,
                                                                     proc-list))))
signatures-match (mg-alist (make-call-environment (mg-state,
                                                        stmt,
                                                        fetch-called-def (stmt,
                                                                         proc-list))),
                    make-name-alist (fetch-called-def (stmt,
                                                         proc-list)))
normal (make-call-environment (mg-state,
                                   fetch-called-def(stmt, proc-list)))
all-cars-unique (mg-alist (make-call-environment (mg-state,
                                                     stmt,
                                                     fetch-called-def (stmt,
                                                                       proc-list))))
(\neg \text{ resource-errorp } (\text{mg-meaning-r} (\text{def-body } (\text{fetch-called-def} (stmt,
                                                                  proc-list)),
                                     proc-list,
                                     make-call-environment (mg-state,
                                                                stmt,
                                                                fetch-called-def (stmt,
                                                                                 proc-list)),
                                     n-1,
                                     list (length (append (reverse (mg-to-p-local-values (def-locals (fetch
                                                            map-down-values (mg-alist (mg-state),
                                                                                bindings (top (ctrl-stk)),
                                                                                temp-stk))),
                                          p-ctrl-stk-size (cons (p-frame (make-frame-alist (fetch-called-o
                                                                                              stmt,
                                                                                              ctrl-stk,
                                                                                              temp-stk),
                                                                          tag('pc,
                                                                               \cos(subr,
                                                                                     1 + (length (code (ci
                                                                                               data-length
                                                                                                length (def-
```

length (call

ctrl-stk)))))))))

```
THEOREM: simple-typed-literal-listp
simple-typed-literal (exp, type) \rightarrow \text{listp} (mg-to-p-simple-literal } (exp))
DEFINITION:
push-array-value-induction-hint (array-value, code, temp-stk)
= if array-value \simeq nil then t
    else push-array-value-induction-hint (cdr (array-value),
                                           append (code,
                                                   list(list('push-constant,
                                                            mg-to-p-simple-literal (car (array-value))))),
                                           push (mg-to-p-simple-literal (car (array-value)),
                                                 temp-stk)) endif
Theorem: push-local-array-values-code-effect
(((length(temp-stk) + length(array-value)) < MG-MAX-TEMP-STK-SIZE))
 \land simple-mg-type-refp (array-elemtype)
 ∧ simple-typed-literal-plistp (array-value, array-elemtype)
 \land (cdddr (assoc (subr, proc-list-code))
      = append (code,
                   append (push-local-array-values-code (array-value),
                            code2))))
     (p (p-state (tag ('pc, cons (subr, length (code)))),
                 ctrl-stk,
                 temp-stk,
                 proc-list-code,
                 data-segment,
                 max-ctrl,
                 MG-MAX-TEMP-STK-SIZE,
                 word-size,
                 'run),
        length(array-value))
      = p-state(tag('pc,
                       cons(subr, length(code) + length(array-value))),
                   append (reverse (mg-to-p-simple-literal-list (array-value)),
                            temp-stk),
                   proc-list-code,
                   data-segment,
                   max-ctrl,
                   MG-MAX-TEMP-STK-SIZE,
                   word-size,
                   'run))
```

```
DEFINITION:
locals-values-induction-hint (locals, code, temp-stk)
    if locals \simeq nil then t
    elseif simple-mg-type-refp (cadr (car (locals)))
    then locals-values-induction-hint (cdr (locals),
                                            append (code,
                                                      list(list('push-constant,
                                                                mg-to-p-simple-literal (caddar (locals))))),
                                            \operatorname{push}(\operatorname{mg-to-p-simple-literal}(\operatorname{caddr}(\operatorname{car}(\operatorname{locals}))),
                                                   temp-stk))
    else locals-values-induction-hint (cdr (locals),
                                           append (code,
                                                     push-local-array-values-code (caddr (car (locals)))),
                                           append (reverse (mg-to-p-simple-literal-list (caddr (car (locals)))),
                                                     temp-stk)) endif
Theorem: call-push-locals-values-effect
(((length(temp-stk) + data-length(locals)) < MG-MAX-TEMP-STK-SIZE)
     ok-mg-local-data-plistp (locals)
      (\operatorname{cdddr}(\operatorname{assoc}(\operatorname{subr}, \operatorname{proc-list-code})))
       = append (code, append (push-locals-values-code (locals), code2))))
      (p (p-state (tag ('pc, cons (subr, length (code)))),
                   ctrl-stk,
                   temp-stk,
                   proc-list-code,
                   data-segment,
                   max-ctrl,
                   MG-MAX-TEMP-STK-SIZE,
                   word-size.
                   'run),
          data-length(locals)
       = p-state(tag('pc,
                          cons(subr, data-length(locals) + length(code))),
                     ctrl-stk,
                     append (reverse (mg-to-p-local-values (locals)), temp-stk),
                     proc-list-code,
                     data-segment,
                     max-ctrl,
                     MG-MAX-TEMP-STK-SIZE,
                     word-size.
                     'run))
```

locals-addresses-induction-hint (locals, code, temp-stk, n)

DEFINITION:

```
if locals \simeq nil then t
    elseif simple-mg-type-refp (cadr(car(locals)))
    then locals-addresses-induction-hint (cdr (locals),
                                              append (code,
                                                        list (list ('push-temp-stk-index,
                                                                 n))),
                                              push (tag ('nat,
                                                          (length(temp-stk) - n) - 1),
                                                     temp-stk),
                                              n)
    else locals-addresses-induction-hint (cdr (locals),
                                             append (code,
                                                      list(list('push-temp-stk-index,
                                                                n))),
                                             push (tag('nat,
                                                         (length(temp-stk))
                                                         - n) - 1),
                                                    temp-stk),
                                             1 + (n - \text{array-length}(\text{cadr}(\text{car}(locals))))) endif
Theorem: call-push-locals-addresses-effect
(((length(temp-stk) + length(locals)) < MG-MAX-TEMP-STK-SIZE)
     ok-mg-local-data-plistp (locals)
      (\operatorname{cdddr}(\operatorname{assoc}(\operatorname{subr}, \operatorname{proc-list-code})))
       = append (code,
                     append (push-locals-addresses-code (locals, n), code2)))
     (n < \text{length}(temp-stk))
     (n \not< (\text{data-length}(locals) - 1)))
     (p (p-state (tag ('pc, cons (subr, length (code)))),
                   ctrl-stk,
                   temp-stk,
                  proc-list-code,
                   data-segment,
                   max-ctrl,
                  MG-MAX-TEMP-STK-SIZE,
                   word-size,
                   'run),
         length (locals))
       = p-state (tag ('pc, cons (subr, length (code) + length (locals))),
                    append (reverse (ascending-local-address-sequence (locals,
                                                                             (length (temp-stk))
                                                                              - n) - 1)),
                              temp-stk),
```

```
proc-list-code,
                     data-segment,
                     max-ctrl,
                    {\tt MG\text{-}MAX\text{-}TEMP\text{-}STK\text{-}SIZE},
                     word-size,
                     'run))
DEFINITION:
call-push-actuals-induction-hint (actuals, code, temp-stk, ctrl-stk)
    if actuals \simeq nil then t
    else call-push-actuals-induction-hint (cdr (actuals),
                                              append (code,
                                                        list (list ('push-local,
                                                                 car(actuals)))),
                                              cons (cdr (assoc (car (actuals),
                                                                bindings (top (ctrl-stk)))),
                                                     temp-stk),
                                              ctrl-stk) endif
Theorem: call-push-actuals-effect
(((length(temp-stk) + length(actuals)) < MG-MAX-TEMP-STK-SIZE)
     (\operatorname{cdddr}(\operatorname{assoc}(\operatorname{subr}, \operatorname{proc-list-code}))
       = append (code, append (push-actuals-code (actuals), code2))))
     (p (p-state (tag ('pc, cons (subr, length (code))),
                   ctrl-stk,
                   temp-stk,
                  proc-list-code,
                   data-segment,
                   max-ctrl,
                  MG-MAX-TEMP-STK-SIZE,
                   word-size,
                   'run),
         length (actuals))
       = p-state (tag ('pc, cons (subr, length (code) + length (actuals))),
                    append (reverse (mg-actuals-to-p-actuals (actuals,
                                                                  bindings (top(ctrl-stk))),
                              temp-stk),
                    proc-list-code,
                     data-segment,
                    max-ctrl,
                    MG-MAX-TEMP-STK-SIZE,
                     word-size,
                     'run))
```

```
Theorem: call-push-parameters-effect1
(((length (temp-stk))
   + data-length (locals)
   + length (locals)
   + length (actuals))
  < MG-MAX-TEMP-STK-SIZE)
 \land ok-mg-local-data-plistp (locals)
     (\operatorname{cdddr}(\operatorname{assoc}(\operatorname{subr}, \operatorname{proc-list-code}))
          append (code,
                     append (push-parameters-code (locals, actuals), code2))))
     (p (p-state (tag ('pc, cons (subr, length (code)))),
                  ctrl-stk,
                  temp-stk,
                  proc-list-code,
                  data-segment,
                  max-ctrl,
                  MG-MAX-TEMP-STK-SIZE,
                  word-size,
                  'run),
         data-length(locals) + length(locals) + length(actuals))
       = p-state(tag('pc,
                         \cos(subr,
                               length(code)
                               + data-length (locals)
                               + length (locals)
                               + length (actuals))),
                    ctrl-stk.
                    append (reverse (mg-actuals-to-p-actuals (actuals,
                                                                 bindings (top(ctrl-stk))),
                             append (reverse (ascending-local-address-sequence (locals,
                                                                                    length(temp-stk))),
                                       append (reverse (mg-to-p-local-values (locals)),
                                                temp-stk))),
                    proc-list-code,
                    data-segment,
                    max-ctrl,
                    MG-MAX-TEMP-STK-SIZE,
                    word-size,
                    'run))
Theorem: call-push-parameters-effect
((\operatorname{car}(stmt) = \operatorname{'proc-call-mg}))
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
     ok-mg-def-plistp (proc-list)
```

```
ok-mg-statep (mg-state, r-cond-list)
    (code (translate-def-body (assoc (subr, proc-list), proc-list))
         append (code (translate (cinfo, t-cond-list, stmt, proc-list)),
                   code2))
Λ
    user-defined-procp (subr, proc-list)
    mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                bindings (top (ctrl-stk)),
                                temp-stk)
    (\neg \text{ resources-inadequatep} (stmt,
                                proc-list,
                                list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk)))))
    (p (map-down (mg-state,
                    proc-list,
                    ctrl-stk,
                    temp-stk.
                    tag('pc, cons(subr, length(code(cinfo)))),
                    t-cond-list),
       data-length (def-locals (fetch-called-def (stmt, proc-list)))
       + length (def-locals (fetch-called-def (stmt, proc-list)))
       + length (call-actuals (stmt)))
        p-state (tag ('pc,
                       \cos(subr,
                             length (code (cinfo))
                             + data-length (def-locals (fetch-called-def (stmt,
                                                                            proc-list)))
                                 length (def-locals (fetch-called-def (stmt,
                                                                      proc-list)))
                                length (call-actuals (stmt)))),
                  ctrl-stk,
                  append (reverse (mg-actuals-to-p-actuals (call-actuals (stmt),
                                                               bindings (top(ctrl-stk))),
                           append (reverse (ascending-local-address-sequence (def-locals (fetch-called-def (stmt
                                                                                 length(temp-stk))),
                                     append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                              map-down-values (mg-alist (mg-state),
                                                                 bindings (top (ctrl-stk)),
                                                                  temp-stk)))),
                  translate-proc-list (proc-list),
                  list (list ('c-c,
                           mg-cond-to-p-nat (cc (mq-state), t-cond-list))),
```

proc

proc-list

MG-MAX-CTRL-STK-SIZE,

```
MG-MAX-TEMP-STK-SIZE,
                    MG-WORD-SIZE,
                    'run))
THEOREM: call-call-step
((n \not\simeq 0)
 \land (\neg resources-inadequatep (stmt,
                                  list (length (temp-stk),
                                       p-ctrl-stk-size (ctrl-stk))))
     (car(stmt) = 'proc-call-mg)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 \land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land 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)
 \land listp (ctrl-stk)
 \land mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                  temp-stk)
 \land no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
 \land signatures-match (mg-alist (mg-state), name-alist)
 \land normal (mg-state)
 \land all-cars-unique (mg-alist (mq-state))
     (\neg \text{ resource-errorp (mg-meaning-r }) stmt,
                                           proc-list,
                                           mg-state,
                                           n,
                                           list (length (temp-stk),
                                                p-ctrl-stk-size (ctrl-stk))))))
     (p-step (p-state (tag ('pc,
                            \cos(subr,
                                  length (code (cinfo))
                                       data-length (def-locals (fetch-called-def (stmt,
                                                                                  proc-list)))
                                       length (def-locals (fetch-called-def (stmt,
                                                                            proc-list)))
                                       length (call-actuals (stmt)))),
                       ctrl-stk,
```

```
append (reverse (mg-actuals-to-p-actuals (call-actuals (stmt),
                                                                                                                                           bindings (top(ctrl-stk))),
                                                     append (reverse (ascending-local-address-sequence (def-locals (fetch-called-def (s
                                                                                                                                                                                          length(temp-stk))),
                                                                           append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                                  map-down-values (mg-alist (mq-state),
                                                                                                                                                  bindings (top (ctrl-stk)),
                                                                                                                                                  temp-stk)))),
                              translate-proc-list (proc-list),
                              list (list ('c-c,
                                                    mg-cond-to-p-nat (cc (mg-state), t-cond-list))),
                              MG-MAX-CTRL-STK-SIZE,
                              MG-MAX-TEMP-STK-SIZE,
                              MG-WORD-SIZE,
                              'run))
p-state (tag ('pc, cons (call-name (stmt), 0)),
                      push (p-frame (append (pairlist (cadr (assoc (call-name (stmt)),
                                                                                                                                         translate-proc-list (proc-list))),
                                                                                                          append (ascending-local-address-sequence (def-locals (fetch-
                                                                                                                                                                                                                          length (temp-stk
                                                                                                                                mg-actuals-to-p-actuals (call-actuals (stmt),
                                                                                                                                                                                                 bindings (top(ctrl-stk))))
                                                                                    pair-temps-with-initial-values (caddr (assoc (call-name (stmt),
                                                                                                                                                                                                       translate-proc-list (proc
                                                             tag('pc,
                                                                         \cos(subr,
                                                                                        length (code (cinfo))
                                                                                                  data-length (def-locals (fetch-called-def (stmt,
                                                                                                                                                                                                           proc-list)))
                                                                                                   length (def-locals (fetch-called-def (stmt,
                                                                                                                                                                                             proc-list)))
                                                                                                   length (call-actuals (stmt))
                                                                                                   1))),
                                      ctrl-stk),
                      {\it append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (\it stmt, \it water))))} and {\it to-p-local-values (def-locals (fetch-called-def (\it stmt, \it water)))))}. The contraction of the contra
                                                                                                                                                                                                 proc-list)))),
                                            map-down-values (mg-alist (mg-state),
                                                                                             bindings (top (ctrl-stk)),
                                                                                             temp-stk)),
                      translate-proc-list (proc-list),
                      list (list ('c-c,
```

proc-

```
mg-cond-to-p-nat (cc (mg-state), t-cond-list))),
                    MG-MAX-CTRL-STK-SIZE,
                    MG-MAX-TEMP-STK-SIZE,
                    MG-WORD-SIZE,
                    'run))
THEOREM: call-steps-to-body
((n \not\simeq 0)
 \land (\neg resources-inadequatep (stmt,
                                  list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
     (car(stmt) = 'proc-call-mg)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 \land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land 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))
 \land user-defined-procp (subr, proc-list)
     plistp(temp-stk)
 \wedge
 \wedge
    listp(ctrl-stk)
 \land mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                  temp-stk)
 \land no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
 \land signatures-match (mg-alist (mg-state), name-alist)
 \land normal (mg\text{-}state)
 \land all-cars-unique (mg-alist (mg-state))
 \land (¬ resource-errorp (mg-meaning-r (stmt,
                                           proc-list,
                                           mg-state,
                                           n,
                                           list (length (temp-stk),
                                               p-ctrl-stk-size (ctrl-stk))))))
     (p (map-down (mg-state,
                     proc-list,
                      ctrl-stk,
                      temp-stk,
                      tag('pc, cons(subr, length(code(cinfo)))),
                      t-cond-list),
         data-length (def-locals (fetch-called-def (stmt, proc-list)))
```

```
+ length (call-actuals (stmt))
         p-state (tag ('pc, cons (call-name (stmt), 0)),
                   push (p-frame (append (pairlist (cadr (assoc (call-name (stmt),
                                                                 translate-proc-list(proc-list))),
                                                     append (ascending-local-address-sequence (def-locals (fetch-
                                                                                                  length (temp-stk
                                                              mg-actuals-to-p-actuals (call-actuals (stmt),
                                                                                        bindings (top(ctrl-stk))))
                                            pair-temps-with-initial-values (caddr (assoc (call-name (stmt),
                                                                                          translate-proc-list (proc
                                   tag('pc,
                                        \cos(subr,
                                              length (code (cinfo))
                                                  data-length (def-locals (fetch-called-def (stmt,
                                                                                            proc-list)))
                                                  length (def-locals (fetch-called-def (stmt,
                                                                                      proc-list)))
                                                  length (call-actuals (stmt))
                                                  1))),
                          ctrl-stk),
                   append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                        proc-list)))),
                            map-down-values (mg-alist (mg-state),
                                                bindings (top (ctrl-stk)),
                                                temp-stk)),
                   translate-proc-list (proc-list),
                   list (list ('c-c,
                            mg-cond-to-p-nat (cc (mg-state), t-cond-list))),
                   MG-MAX-CTRL-STK-SIZE,
                   MG-MAX-TEMP-STK-SIZE,
                   MG-WORD-SIZE,
                    'run))
EVENT: Disable deposit-alist-value.
Theorem: call-local-var-lists-match1
(k \in \mathbf{N})
   (pairlist (listcars (locals), ascending-local-address-sequence (locals, k))
          map-call-locals (locals, k))
THEOREM: call-formal-var-lists-match
```

length (def-locals (fetch-called-def (stmt, proc-list)))

```
data-param-lists-match (actuals, formals, alist)
     (pairlist (listcars (formals), mg-actuals-to-p-actuals (actuals, bindings))
          map-call-formals (formals, actuals, bindings))
DEFINITION:
map-down-locals-doesnt-affect-formals-induction-hint (formals,
                                                       actuals,
                                                       temp-stk,
                                                       mg-alist,
                                                       bindings)
    if formals \simeq nil then t
    else map-down-locals-doesnt-affect-formals-induction-hint (cdr (formals),
                                                                 \operatorname{cdr}(actuals),
                                                                 deposit-alist-value (list (caar (formals),
                                                                                         cadar (formals).
                                                                                         caddr (assoc (car (actu
                                                                                                       mq-alist)
                                                                                     cdr (assoc (car (
                                                                                          map-call-formals (cdr
                                                                                     temp-stk),
                                                                 mg-alist,
                                                                 bindings) endif
Theorem: map-down-locals-doesnt-affect-formals
all-cars-unique (append (formals, locals))
     (map-down-values (make-call-param-alist (formals, actuals, mg-alist),
                         append (map-call-locals (locals, n),
                                  map-call-formals (formals, actuals, bindings)),
                         temp-stk)
          map-down-values (make-call-param-alist (formals,
                                                     actuals.
                                                     mg-alist),
                             map-call-formals (formals, actuals, bindings),
                             temp-stk))
DEFINITION:
map-down-formals-doesnt-affect-locals-induction-hint (locals, temp-stk, n, lst)
   if locals \simeq nil then t
    elseif simple-mg-type-refp (cadr(car(locals)))
    then map-down-formals-doesnt-affect-locals-induction-hint (cdr (locals),
                                                                  deposit-alist-value (car (locals),
```

bind

cdrbin

```
append (map-call-locals (l
                                                                                                              lst),
                                                                                                    temp-stk),
                                                                            1 + n,
                                                                             lst)
     else map-down-formals-doesnt-affect-locals-induction-hint (cdr (locals),
                                                                           deposit-alist-value (car(locals),
                                                                                                  append (map-call-locals (lo
                                                                                                             lst).
                                                                                                   temp-stk),
                                                                           \operatorname{array-length}\left(\operatorname{cadr}\left(\operatorname{car}\left(\operatorname{locals}\right)\right)\right)
                                                                           + n,
                                                                           lst) endif
Theorem: map-down-formals-doesnt-affect-locals
all-cars-unique (locals)
      (map-down-values (locals,
                             append (map-call-locals (locals, n), lst),
                             temp-stk)
            map-down-values (locals, map-call-locals (locals, n), temp-stk))
Theorem: map-down-skips-non-referenced-segment
(\text{mg-alistp}(vars) \land \text{mg-vars-list-ok-in-p-state}(vars, bindings, temp-stk))
      (map-down-values (vars, bindings, append (lst, temp-stk))
       = append (lst, map-down-values (vars, bindings, temp-stk)))
DEFINITION:
map-down-locals-induction-hint (locals, n, temp-stk)
= if locals \simeq nil then t
     elseif simple-mg-type-refp (cadr (car (locals)))
     then map-down-locals-induction-hint (cdr (locals),
                                                   1 + n,
                                                   cons (mg-to-p-simple-literal (caddar (locals)),
                                                          temp-stk))
     else map-down-locals-induction-hint (cdr (locals),
                                                  \operatorname{array-length}\left(\operatorname{cadr}\left(\operatorname{car}\left(\operatorname{locals}\right)\right)\right)
                                                  append (reverse (mg-to-p-simple-literal-list (caddr (car (locals)))),
                                                            temp-stk)) endif
Theorem: map-down-locals-equals-reverse-values
```

((n = length(temp-stk))) $\land \quad \text{all-cars-unique}(locals)$

```
ok-mg-local-data-plistp (locals)
 \land plistp (temp-stk))
     (map-down-values (locals,
                          map-call-locals (locals, n),
                          append (reverse (mg-to-p-local-values (locals)),
                                   temp-stk)
           append (reverse (mg-to-p-local-values (locals)), temp-stk))
Theorem: map-down-again-preserves-values
(mg-vars-list-ok-in-p-state (mg-vars, bindings, temp-stk)
 \land ok-actual-params-list (actuals, mg-vars)
 ∧ data-param-lists-match (actuals, formals, mg-vars)
 \land all-cars-unique (mq-vars)
 \land plistp (temp-stk)
 \land no-p-aliasing (bindings, mq-vars)
 \land all-cars-unique (formals)
 \land mg-alistp (mq\text{-}vars)
     (map-down-values (make-call-param-alist (formals, actuals, mg-vars),
                          map-call-formals (formals, actuals, bindings),
                          map-down-values (mg-vars, bindings, temp-stk))
           map-down-values (mq-vars, bindings, temp-stk))
EVENT: Enable deposit-temp.
Theorem: array-formal-ok-for-actual2
(definedp(actual, mg-alist)
 ∧ data-params-match (actual, formal, mg-alist)
 \land mg-vars-list-ok-in-p-state (mg-alist, bindings, temp-stk)
 \land (\neg simple-mg-type-refp (cadr (formal))))
   ok-temp-stk-array-index (cdr (assoc (actual, bindings)),
                                map-down-values (mq-alist, bindings, temp-stk),
                                \operatorname{array-length}\left(\operatorname{cadr}\left(formal\right)\right)\right)
EVENT: Disable array-formal-ok-for-actual2.
Theorem: call-environment-mg-vars-list-ok1
(no-p-aliasing (bindings, mg-vars)
 \land mg-alistp (mq-vars)
 \wedge all-cars-unique (formals)
 \land ok-actual-params-list (actuals, mg-vars)
 ∧ ok-mg-formal-data-params-plistp (formals)
 ∧ data-param-lists-match (actuals, formals, mg-vars)
 \land mg-vars-list-ok-in-p-state (mg-vars, bindings, temp-stk))
 → mg-vars-list-ok-in-p-state (make-call-param-alist (formals,
```

```
actuals,
                                                           mg-vars),
                                  map-call-formals (formals, actuals, bindings),
                                  map-down-values (mg-vars, bindings, temp-stk))
Theorem: map-down-preserves-references
((n \not\simeq 0)
 \land (\neg resources-inadequatep (stmt,
                                  proc-list,
                                  list (length (temp-stk),
                                       p-ctrl-stk-size(ctrl-stk))))
     (car(stmt) = 'proc-call-mg)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 \land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land cond-subsetp (r-cond-list, t-cond-list)
 \land (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)
 \wedge
     listp(ctrl-stk)
 \wedge
     mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                  temp-stk)
 \land no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
 \land signatures-match (mg-alist (mg-state), name-alist)
 \land normal (mq\text{-}state)
 \land all-cars-unique (mg-alist (mg-state))
     (\neg \text{ resource-errorp (mg-meaning-r }) stmt,
                                           proc-list,
                                           mg-state,
                                           n,
                                           list (length (temp-stk),
                                                p-ctrl-stk-size (ctrl-stk))))))
     (append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                             proc-list)))),
                map-down-values (mg-alist (mg-state),
                                    bindings (top (ctrl-stk)),
                                    temp-stk)
           map-down-values (make-call-var-alist (mg-alist (mg-state),
                                                    fetch-called-def (stmt,
```

```
proc-list)),
                               make-frame-alist (fetch-called-def (stmt,
                                                                     proc-list),
                                                   stmt,
                                                   ctrl-stk.
                                                   temp-stk),
                               append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                                       proc-list)))),
                                         map-down-values (mg-alist (mg-state),
                                                             bindings (top (ctrl-stk)),
                                                             temp-stk))))
Theorem: call-step-initial-equals-state1
((n \not\simeq 0)
 \land (\neg resources-inadequatep (stmt,
                                   proc-list,
                                  list (length (temp-stk),
                                       p-ctrl-stk-size (ctrl-stk))))
     (car(stmt) = 'proc-call-mg)
 \land ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 \land ok-mg-def-plistp (proc-list)
 ∧ ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
 \land ok-mg-statep (mg-state, r-cond-list)
 \land 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))
 \land user-defined-procp (subr, proc-list)
      plistp (temp-stk)
 \land listp (ctrl-stk)
     mg-vars-list-ok-in-p-state (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                   temp-stk)
 \land no-p-aliasing (bindings (top (ctrl-stk)), mg-alist (mg-state))
 \land signatures-match (mg-alist (mg-state), name-alist)
 \land normal (mg\text{-}state)
 \land all-cars-unique (mg-alist (mg-state))
     (\neg \text{ resource-errorp } (\text{mg-meaning-r } (stmt,
                                            proc-list,
                                            mg-state,
                                            n,
                                            list (length (temp-stk),
                                                p-ctrl-stk-size (ctrl-stk))))))
     (p-state(tag('pc, cons(call-name(stmt), 0)),
```

```
push (p-frame (append (pairlist (cadr (assoc (call-name (stmt)),
                                                   translate-proc-list(proc-list))),
                                      append (ascending-local-address-sequence (def-locals (fetch-calle
                                                                                   length(temp-stk)),
                                               mg-actuals-to-p-actuals (call-actuals (stmt),
                                                                         bindings (top(ctrl-stk)))),
                             pair-temps-with-initial-values (caddr (assoc (call-name (stmt)),
                                                                            translate-proc-list (proc-list
                    tag('pc,
                         \cos(subr,
                               length (code (cinfo))
                                   data-length (def-locals (fetch-called-def (stmt,
                                                                             proc-list)))
                                   length (def-locals (fetch-called-def (stmt,
                                                                        proc-list)))
                                   length (call-actuals (stmt))
                                   1))),
           ctrl-stk),
     append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                         proc-list)))),
              map-down-values (mg-alist (mg-state),
                                 bindings (top (ctrl-stk)),
                                 temp-stk)),
     translate-proc-list (proc-list),
     list (list ('c-c, mg-cond-to-p-nat (cc (mg-state), t-cond-list))),
     MG-MAX-CTRL-STK-SIZE,
     MG-MAX-TEMP-STK-SIZE,
     MG-WORD-SIZE,
     'run)
map-down (make-call-environment (mg-state,
                                      stmt,
                                      fetch-called-def (stmt,
                                                       proc-list)),
             proc-list,
             cons (p-frame (make-frame-alist (fetch-called-def (stmt,
                                                                proc-list),
                                               stmt,
                                               ctrl-stk,
                                               temp-stk),
                            tag('pc,
                                 \cos(subr,
                                       1 + (length (code (cinfo)))
                                                data-length (def-locals (fetch-called-def (stmt,
```

```
proc-list)))
                                                    length (def-locals (fetch-called-def (stmt,
                                                                                    proc-list)))
                                                    length (call-actuals (stmt)))))),
                         ctrl-stk),
                    append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                proc-list)))),
                            map-down-values (mg-alist (mg-state),
                                             bindings (top (ctrl-stk)),
                                             temp-stk)),
                    tag('pc,
                        cons(call-name(stmt),
                             length (code (make-cinfo (nil,
                                                     \cos ('(routineerror
                                                          make-label-alist (make-cond-list (fetch-called-d
                                                                          0)),
                                                     1))))),
                     make-cond-list (fetch-called-def(stmt, proc-list)))) 
;; This lemma characterizes the bevavior all of the way through the call-body
;; This lemma characterizes the bevavior all of the way through the call-body
(prove-lemma call-step-initial-to-state2 (rewrite)
      (IMPLIES
       (AND (NOT (ZEROP N))
    (NOT (RESOURCES-INADEQUATEP STMT PROC-LIST
(LIST (LENGTH TEMP-STK)
       (P-CTRL-STK-SIZE CTRL-STK))))
    (EQUAL (CAR STMT) 'PROC-CALL-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)
    (EQUAL (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))
    (NOT (RESOURCE-ERRORP (MG-MEANING-R STMT PROC-LIST MG-STATE N
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK)))))
            (EQUAL
      (P
       (MAP-DOWN
                                                               ;; state1
(MAKE-CALL-ENVIRONMENT MG-STATE STMT
       (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
(CONS
 (P-FRAME
  (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
   STMT CTRL-STK TEMP-STK)
  (TAG 'PC
       (CONS SUBR
     (ADD1
      (PLUS (LENGTH (CODE CINFO))
    (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
    (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
    (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
(APPEND
 (REVERSE
  (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
 (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
  (BINDINGS (TOP CTRL-STK))
 TEMP-STK))
(TAG 'PC
     (CONS
      (CALL-NAME STMT)
      (LENGTH
       (CODE
```

```
(MAKE-CINFO NIL
    (CONS
     '(ROUTINEERROR . 0)
     (MAKE-LABEL-ALIST
      (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
     0))
   1)))))
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))
       (CLOCK (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
     PROC-LIST
      (MAKE-CALL-ENVIRONMENT MG-STATE STMT
     (FETCH-CALLED-DEF STMT PROC-LIST))
      (SUB1 N)))
      (P-STATE
                                                    ;; state2
       (TAG 'PC
    (CONS
     (CALL-NAME STMT)
     (IF
      (NORMAL
       (MG-MEANING-R
(DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
(MAKE-CALL-ENVIRONMENT MG-STATE STMT
       (FETCH-CALLED-DEF STMT PROC-LIST))
(SUB1 N)
(LIST
 (LENGTH
  (APPEND
   (REVERSE
    (MG-TO-P-LOCAL-VALUES
     (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
   (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
    (BINDINGS (TOP CTRL-STK))
   TEMP-STK)))
 (P-CTRL-STK-SIZE
  (CONS
   (P-FRAME
    (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
     STMT CTRL-STK TEMP-STK)
    (TAG 'PC
 (CONS SUBR
       (ADD1
(PLUS
 (LENGTH (CODE CINFO))
```

```
(DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))))
      (LENGTH
       (CODE
(TRANSLATE
 (MAKE-CINFO NIL
     (CONS
      '(ROUTINEERROR . 0)
      (MAKE-LABEL-ALIST
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
       0))
     1)
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST)))
      (FIND-LABEL
       (FETCH-LABEL
(CC
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
   (LENGTH
    (APPEND
     (REVERSE
      (MG-TO-P-LOCAL-VALUES
                                (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
      (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
STMT CTRL-STK TEMP-STK)
     (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
```

```
(LENGTH (CODE CINFO))
   (DATA-LENGTH
    (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH
    (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
(LABEL-ALIST
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
      1)
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST)))
       (APPEND
(CODE
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
      '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
      1)
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST))
(CONS
 '(DL O NIL (NO-OP))
 (CONS
  (LIST 'POP* (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
 '((RET)))))))))
       (CONS
(P-FRAME
 (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
  STMT CTRL-STK TEMP-STK)
 (TAG 'PC
      (CONS SUBR
    (ADD1
```

```
(PLUS (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
       (MAP-DOWN-VALUES
(MG-ALIST
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
   (LENGTH
    (APPEND
     (REVERSE
      (MG-TO-P-LOCAL-VALUES
       (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
      (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
     (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
STMT CTRL-STK TEMP-STK)
      (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
    CTRL-STK)))))
(BINDINGS
 (TOP
  (CONS
   (P-FRAME
    (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
     STMT CTRL-STK TEMP-STK)
    (TAG 'PC
 (CONS SUBR
```

```
(ADD1
(PLUS (LENGTH (CODE CINFO))
      (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
      (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
      (LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))
(APPEND
 (REVERSE
  (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
 (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
 (BINDINGS (TOP CTRL-STK))
 TEMP-STK)))
       (TRANSLATE-PROC-LIST PROC-LIST)
       (LIST
(LIST 'C-C
      (MG-COND-TO-P-NAT
       (CC
(MG-MEANING-R
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
 (MAKE-CALL-ENVIRONMENT MG-STATE STMT
(FETCH-CALLED-DEF STMT PROC-LIST))
 (SUB1 N)
 (LIST
 (LENGTH
  (APPEND
   (REVERSE
     (MG-TO-P-LOCAL-VALUES
      (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
    (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
    TEMP-STK)))
  (P-CTRL-STK-SIZE
  (CONS
    (P-FRAME
     (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
      STMT CTRL-STK TEMP-STK)
     (TAG 'PC
  (CONS SUBR
(ADD1
 (PLUS
 (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
```

```
(LENGTH (CALL-ACTUALS STMT)))))))
    CTRL-STK)))))
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
       (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
       'RUN)))
       (EQUAL
(P (MAP-DOWN MG-STATE PROC-LIST CTRL-STK TEMP-STK
     (TAG 'PC
  (CONS SUBR (LENGTH (CODE CINFO))))
     T-COND-LIST)
           (plus (data-length (def-locals (fetch-called-def stmt proc-list)))
 (length (def-locals (fetch-called-def stmt proc-list)))
 (length (call-actuals stmt))
 (CLOCK (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
(MAKE-CALL-ENVIRONMENT MG-STATE STMT
       (FETCH-CALLED-DEF STMT PROC-LIST))
(SUB1 N))))
      (P-STATE
                                                    ;; state2
       (TAG 'PC
    (CONS
     (CALL-NAME STMT)
     (IF
      (NORMAL
       (MG-MEANING-R
(DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
(MAKE-CALL-ENVIRONMENT MG-STATE STMT
       (FETCH-CALLED-DEF STMT PROC-LIST))
(SUB1 N)
(LIST
 (LENGTH
  (APPEND
   (REVERSE
    (MG-TO-P-LOCAL-VALUES
     (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
   (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
    (BINDINGS (TOP CTRL-STK))
   TEMP-STK)))
 (P-CTRL-STK-SIZE
  (CONS
   (P-FRAME
    (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
```

```
STMT CTRL-STK TEMP-STK)
    (TAG 'PC
 (CONS SUBR
       (ADD1
(PLUS
 (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))))
      (LENGTH
       (CODE
(TRANSLATE
 (MAKE-CINFO NIL
     (CONS
      '(ROUTINEERROR . 0)
      (MAKE-LABEL-ALIST
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
      0))
    1)
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST)))
      (FIND-LABEL
       (FETCH-LABEL
(CC
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
  (LENGTH
    (APPEND
     (REVERSE
     (MG-TO-P-LOCAL-VALUES
                                (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
```

```
(MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
STMT CTRL-STK TEMP-STK)
      (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
(LABEL-ALIST
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
     1)
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST)))
       (APPEND
(CODE
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST))
(CONS
 '(DL O NIL (NO-OP))
 (CONS
  (LIST 'POP* (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
  '((RET))))))))))
       (CONS
(P-FRAME
 (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
```

```
STMT CTRL-STK TEMP-STK)
 (TAG 'PC
      (CONS SUBR
    (ADD1
     (PLUS (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
       (MAP-DOWN-VALUES
(MG-ALIST
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
   (LENGTH
    (APPEND
     (REVERSE
     (MG-TO-P-LOCAL-VALUES
       (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
      (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
STMT CTRL-STK TEMP-STK)
      (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
    CTRL-STK)))))
(BINDINGS
 (TOP
  (CONS
   (P-FRAME
```

```
(MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
     STMT CTRL-STK TEMP-STK)
    (TAG 'PC
 (CONS SUBR
       (ADD1
(PLUS (LENGTH (CODE CINFO))
      (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
      (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
      (LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))
(APPEND
 (REVERSE
  (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
 (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
  (BINDINGS (TOP CTRL-STK))
 TEMP-STK)))
       (TRANSLATE-PROC-LIST PROC-LIST)
(LIST 'C-C
      (MG-COND-TO-P-NAT
       (CC
(MG-MEANING-R
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
 (MAKE-CALL-ENVIRONMENT MG-STATE STMT
(FETCH-CALLED-DEF STMT PROC-LIST))
 (SUB1 N)
 (LIST
  (LENGTH
   (APPEND
    (REVERSE
     (MG-TO-P-LOCAL-VALUES
      (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
    (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
  (P-CTRL-STK-SIZE
   (CONS
    (P-FRAME
     (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
      STMT CTRL-STK TEMP-STK)
     (TAG 'PC
  (CONS SUBR
(ADD1
```

```
(PLUS
(LENGTH (CODE CINFO))
(DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
(LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
(LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))))
      (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
     (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
     'RUN)))
((INSTRUCTIONS
 (ADD-ABBREVIATION @INITIAL
                    (MAP-DOWN MG-STATE PROC-LIST CTRL-STK TEMP-STK
                              (TAG 'PC
                                   (CONS SUBR (LENGTH (CODE CINFO))))
                              T-COND-LIST))
 (ADD-ABBREVIATION @STATE1
   (MAP-DOWN
   (MAKE-CALL-ENVIRONMENT MG-STATE STMT
                           (FETCH-CALLED-DEF STMT PROC-LIST))
   PROC-LIST
   (CONS
     (P-FRAME
     (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                        STMT CTRL-STK TEMP-STK)
     (TAG 'PC
      (CONS SUBR
       (ADD1
            (PLUS (LENGTH (CODE CINFO))
                  (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                  (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                  (LENGTH (CALL-ACTUALS STMT)))))))
    CTRL-STK)
   (APPEND
     (REVERSE
        (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                      (BINDINGS (TOP CTRL-STK))
                      TEMP-STK))
   (TAG 'PC
     (CONS
      (CALL-NAME STMT)
     (LENGTH
      (CODE
       (MAKE-CINFO NIL
```

```
(CONS
        '(ROUTINEERROR . 0)
        (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                          0))
      1)))))
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))))
(ADD-ABBREVIATION @STATE2
 (P-STATE
 (TAG 'PC
   (CONS
   (CALL-NAME STMT)
   (IF
     (NORMAL
      (MG-MEANING-R
       (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
      PROC-LIST
       (MAKE-CALL-ENVIRONMENT MG-STATE STMT
                              (FETCH-CALLED-DEF STMT PROC-LIST))
       (SUB1 N)
       (LIST
        (LENGTH
         (APPEND
          (REVERSE
           (MG-TO-P-LOCAL-VALUES
                            (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
          (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                           (BINDINGS (TOP CTRL-STK))
                           TEMP-STK)))
        (P-CTRL-STK-SIZE
         (CONS
          (P-FRAME
           (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                             STMT CTRL-STK TEMP-STK)
           (TAG 'PC
            (CONS SUBR
             (ADD1
              (PLUS
                (LENGTH (CODE CINFO))
                (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                (LENGTH (CALL-ACTUALS STMT)))))))
         CTRL-STK)))))
     (LENGTH
      (CODE
```

```
(TRANSLATE
   (MAKE-CINFO NIL
   (CONS
    '(ROUTINEERROR . 0)
     (MAKE-LABEL-ALIST
                     (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                     0))
   1)
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST)))
(FIND-LABEL
(FETCH-LABEL
 (CC
   (MG-MEANING-R
   (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
   PROC-LIST
   (MAKE-CALL-ENVIRONMENT MG-STATE STMT
                           (FETCH-CALLED-DEF STMT PROC-LIST))
   (SUB1 N)
   (LIST
    (LENGTH
      (APPEND
       (REVERSE
       (MG-TO-P-LOCAL-VALUES
                       (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
       (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                        (BINDINGS (TOP CTRL-STK))
                        TEMP-STK)))
     (P-CTRL-STK-SIZE
      (CONS
       (P-FRAME
        (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                          STMT CTRL-STK TEMP-STK)
        (TAG 'PC
         (CONS SUBR
          (ADD1
           (PLUS
            (LENGTH (CODE CINFO))
            (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
            (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
            (LENGTH (CALL-ACTUALS STMT)))))))
      CTRL-STK)))))
 (LABEL-ALIST
```

```
(TRANSLATE
       (MAKE-CINFO NIL
        (CONS
         '(ROUTINEERROR . 0)
         (MAKE-LABEL-ALIST
                        (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                        0))
       1)
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
       (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
      PROC-LIST)))
   (APPEND
     (CODE
      (TRANSLATE
       (MAKE-CINFO NIL
        (CONS
         '(ROUTINEERROR . 0)
         (MAKE-LABEL-ALIST
                        (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
       1)
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
       (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
      PROC-LIST))
     (CONS
      '(DL O NIL (NO-OP))
      (CONS
       (LIST 'POP* (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
       '((RET))))))))))
(CONS
 (P-FRAME
 (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                    STMT CTRL-STK TEMP-STK)
 (TAG 'PC
   (CONS SUBR
   (ADD1
        (PLUS (LENGTH (CODE CINFO))
              (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
              (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
              (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
(MAP-DOWN-VALUES
 (MG-ALIST
 (MG-MEANING-R
```

```
(DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
                         (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
  (LENGTH
   (APPEND
     (REVERSE
     (MG-TO-P-LOCAL-VALUES
                         (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                      (BINDINGS (TOP CTRL-STK))
                      TEMP-STK)))
  (P-CTRL-STK-SIZE
   (CONS
     (P-FRAME
      (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                        STMT CTRL-STK TEMP-STK)
      (TAG 'PC
      (CONS SUBR
        (ADD1
         (PLUS
             (LENGTH (CODE CINFO))
             (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
             (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
             (LENGTH (CALL-ACTUALS STMT)))))))
    CTRL-STK)))))
(BINDINGS
(TOP
  (CONS
  (P-FRAME
   (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                      STMT CTRL-STK TEMP-STK)
   (TAG 'PC
     (CONS SUBR
      (ADD1
       (PLUS (LENGTH (CODE CINFO))
             (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
             (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
             (LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))
(APPEND
(REVERSE
```

```
(MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
   (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                     (BINDINGS (TOP CTRL-STK))
                     TEMP-STK)))
  (TRANSLATE-PROC-LIST PROC-LIST)
  (LIST
   (LIST 'C-C
   (MG-COND-TO-P-NAT
    (CC
      (MG-MEANING-R
       (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
      PROC-LIST
       (MAKE-CALL-ENVIRONMENT MG-STATE STMT
                              (FETCH-CALLED-DEF STMT PROC-LIST))
       (SUB1 N)
       (LIST
       (LENGTH
         (APPEND
          (REVERSE
           (MG-TO-P-LOCAL-VALUES
                            (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
          (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                           (BINDINGS (TOP CTRL-STK))
                           TEMP-STK)))
        (P-CTRL-STK-SIZE
         (CONS
          (P-FRAME
           (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                             STMT CTRL-STK TEMP-STK)
           (TAG 'PC
            (CONS SUBR
             (ADD1
              (PLUS
                (LENGTH (CODE CINFO))
                (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                (LENGTH (CALL-ACTUALS STMT)))))))
         CTRL-STK)))))
     (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
  (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
  'RUN))
(ADD-ABBREVIATION @BODY-TIME
            (CLOCK (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
                  PROC-LIST
```

```
(MAKE-CALL-ENVIRONMENT MG-STATE STMT
                                              (FETCH-CALLED-DEF STMT PROC-LIST))
                      (SUB1 N)))
   (ADD-ABBREVIATION @TIME-TO-STATE1
             (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                   (LENGTH (CALL-ACTUALS STMT))
                   1))
  PROMOTE
   (DIVE 1 2)
   (= (PLUS (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                  (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                  (LENGTH (CALL-ACTUALS STMT))
                  1)
            @BODY-TIME))
  UP
   (REWRITE P-PLUS-LEMMA)
   (DIVE 1)
   (REWRITE CALL-STEPS-TO-BODY)
   (REWRITE CALL-STEP-INITIAL-EQUALS-STATE1)
  UP UP
   (DEMOTE 19)
  S-PROP)))
EVENT: Enable make-call-environment.
(prove-lemma call-state2-step1-effect (rewrite)
      (IMPLIES
       (AND (NOT (ZEROP N))
    (NOT (RESOURCES-INADEQUATEP STMT PROC-LIST
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK))))
    (EQUAL (CAR STMT) 'PROC-CALL-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)
    (EQUAL (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))
    (NOT (RESOURCE-ERRORP (MG-MEANING-R STMT PROC-LIST MG-STATE N
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK))))))
     (equal
      (p-step
              (P-STATE
                                                            ;; state2
       (TAG 'PC
    (CONS
     (CALL-NAME STMT)
     (IF
      (NORMAL
       (MG-MEANING-R
(DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
(MAKE-CALL-ENVIRONMENT MG-STATE STMT
       (FETCH-CALLED-DEF STMT PROC-LIST))
(SUB1 N)
(LIST
 (LENGTH
  (APPEND
   (REVERSE
    (MG-TO-P-LOCAL-VALUES
     (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
   (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
    (BINDINGS (TOP CTRL-STK))
   TEMP-STK)))
 (P-CTRL-STK-SIZE
  (CONS
   (P-FRAME
    (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
     STMT CTRL-STK TEMP-STK)
    (TAG 'PC
```

```
(CONS SUBR
       (ADD1
(PLUS
 (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
   CTRL-STK)))))
      (LENGTH
       (CODE
(TRANSLATE
 (MAKE-CINFO NIL
     (CONS
      '(ROUTINEERROR . 0)
      (MAKE-LABEL-ALIST
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
       0))
     1)
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST)))
      (FIND-LABEL
       (FETCH-LABEL
(CC
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
   (LENGTH
    (APPEND
     (REVERSE
      (MG-TO-P-LOCAL-VALUES
                                (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
      (BINDINGS (TOP CTRL-STK))
      TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
STMT CTRL-STK TEMP-STK)
```

```
(TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH
    (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH
    (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
(LABEL-ALIST
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
     1)
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST)))
       (APPEND
(CODE
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST))
(CONS
 '(DL O NIL (NO-OP))
 (CONS
  (LIST 'POP* (Data-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
  '((RET)))))))))
       (CONS
(P-FRAME
 (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
```

```
STMT CTRL-STK TEMP-STK)
 (TAG 'PC
      (CONS SUBR
    (ADD1
     (PLUS (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
       (MAP-DOWN-VALUES
(MG-ALIST
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
  (MAKE-CALL-ENVIRONMENT MG-STATE STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
  (SUB1 N)
  (LIST
   (LENGTH
    (APPEND
     (REVERSE
     (MG-TO-P-LOCAL-VALUES
       (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
      (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
STMT CTRL-STK TEMP-STK)
      (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
    CTRL-STK)))))
(BINDINGS
 (TOP
  (CONS
   (P-FRAME
```

```
(MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
     STMT CTRL-STK TEMP-STK)
    (TAG 'PC
 (CONS SUBR
       (ADD1
(PLUS (LENGTH (CODE CINFO))
      (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
      (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
      (LENGTH (CALL-ACTUALS STMT)))))))
  CTRL-STK)))
(APPEND
 (REVERSE
  (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
 (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
  (BINDINGS (TOP CTRL-STK))
 TEMP-STK)))
       (TRANSLATE-PROC-LIST PROC-LIST)
(LIST 'C-C
      (MG-COND-TO-P-NAT
       (CC
(MG-MEANING-R
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
PROC-LIST
 (MAKE-CALL-ENVIRONMENT MG-STATE STMT
(FETCH-CALLED-DEF STMT PROC-LIST))
 (SUB1 N)
 (LIST
  (LENGTH
   (APPEND
    (REVERSE
     (MG-TO-P-LOCAL-VALUES
      (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
    (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
  (P-CTRL-STK-SIZE
   (CONS
    (P-FRAME
     (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
      STMT CTRL-STK TEMP-STK)
     (TAG 'PC
  (CONS SUBR
(ADD1
```

```
(PLUS
  (LENGTH (CODE CINFO))
  (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
  (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
  (LENGTH (CALL-ACTUALS STMT)))))))
   CTRL-STK)))))
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
       (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
       'RUN))
       (P-STATE (TAG 'PC
     (CONS
      (CALL-NAME STMT)
      (PLUS
       (LENGTH
(CODE
 (TRANSLATE
  (MAKE-CINFO NIL
      (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
     1)
  (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST)))
       1)))
(CONS
 (P-FRAME
  (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (LENGTH TEMP-STK))
  (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
    (CALL-ACTUALS STMT)
    (BINDINGS (TOP CTRL-STK))))
  (TAG 'PC
       (CONS SUBR
     (ADD1 (PLUS (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
(MAP-DOWN-VALUES
 (MG-ALIST
  (MG-MEANING-R
   (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
```

```
PROC-LIST
  (MG-STATE (CC MG-STATE)
     (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 STMT
  (FETCH-CALLED-DEF STMT PROC-LIST))
     (MG-PSW MG-STATE))
  (SUB1 N)
  (LIST (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
       (LENGTH TEMP-STK))
 (PLUS 2
       (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
       (LENGTH (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST)))
       (P-CTRL-STK-SIZE CTRL-STK)))))
 (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (LENGTH TEMP-STK))
 (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (CALL-ACTUALS STMT)
  (BINDINGS (TOP CTRL-STK))))
 (APPEND
  (REVERSE
  (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
  (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
  (BINDINGS (TOP CTRL-STK))
  TEMP-STK)))
(TRANSLATE-PROC-LIST PROC-LIST)
(LIST
 (LIST 'C-C
       (MG-COND-TO-P-NAT
(CC
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
 (MG-STATE (CC MG-STATE)
   (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 (FETCH-CALLED-DEF STMT PROC-LIST))
   (MG-PSW MG-STATE))
  (SUB1 N)
  (LIST
  (PLUS
   (LENGTH
     (MG-TO-P-LOCAL-VALUES
      (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
   (LENGTH (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
```

```
(BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (APPEND
       (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
(LENGTH TEMP-STK))
       (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
 (CALL-ACTUALS STMT)
 (BINDINGS (TOP CTRL-STK))))
      (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
                (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
'RUN)))
 ((INSTRUCTIONS
   (ENABLE LENGTH-CONS)
  PROMOTE
   (DIVE 1)
   (S LEMMAS)
   (DIVE 1 1 1)
   (= *
    (LENGTH
     (CODE
      (TRANSLATE
       (MAKE-CINFO NIL
        (CONS
           '(ROUTINEERROR . 0)
           (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                             0))
       1)
       (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
       (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
      PROC-LIST)))
   0)
```

```
UP
(DIVE 2)
(REWRITE TRANSLATE-DEF-BODY-REWRITE
 (($CINFO
   (MAKE-CINFO NIL
    (CONS
        (CONS 'ROUTINEERROR O)
        (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                          0))
    1))
  ($T-COND-LIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))
  ($CODE2
   (LIST '(DL O NIL (NO-OP))
         (LIST 'POP*
               (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
         '(RET)))
  ($STMT (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST)))))
(REWRITE GET-LENGTH-CAR)
(S LEMMAS)
UP X UP X
(S LEMMAS)
(S LEMMAS)
(DIVE 1 2 2 1)
(= *
 (LENGTH
  (CODE
   (TRANSLATE
    (MAKE-CINFO NIL
     (CONS
        '(ROUTINEERROR . 0)
        (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                          0))
     1)
    (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
    (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
    PROC-LIST)))
 0)
UP UP UP UP
(DIVE 3 1 1 5 1 1)
(REWRITE LENGTH-MG-TO-P-LOCAL-VALUES)
(REWRITE MAP-DOWN-VALUES-PRESERVES-LENGTH)
```

```
UP NX
   (DIVE 1)
   (S LEMMAS)
   (S-PROP P-FRAME-SIZE)
   (S LEMMAS)
  TOP S SPLIT PROVE PROVE
   (REWRITE OK-MG-STATEP-MG-ALIST-MG-ALISTP)
   (REWRITE CALLED-DEF-FORMALS-OK)
  S-PROP SPLIT
   (DIVE 1 1)
   (REWRITE FETCH-LABEL-0-CASE-2)
   (REWRITE FIND-LABEL-APPEND)
  UP DROP
   (PROVE (ENABLE UNLABEL))
   (DIVE 1)
   (REWRITE FIND-LABELP-MONOTONIC-LESSP)
   TOP S
   (S LEMMAS)
   (S LEMMAS)
   (PROVE (ENABLE OK-MG-STATEMENT OK-PROC-CALL))
   (DIVE 1 1)
  X UP
   (S LEMMAS)
   (PROVE (ENABLE FETCH-CALLED-DEF FETCH-DEF))
  S-PROP SPLIT
   (DIVE 1 1)
   (REWRITE FETCH-LABEL-O-CASE-2)
   (REWRITE FIND-LABEL-APPEND)
  TOP DROP
   (PROVE (ENABLE UNLABEL))
   (DIVE 1)
   (REWRITE FIND-LABELP-MONOTONIC-LESSP)
  UP S
   (S LEMMAS)
   (S LEMMAS))))
Theorem: call-body-mg-vars-list-ok1
((n \not\simeq 0)
 \land (¬ resources-inadequatep (stmt,
                            proc-list,
```

```
list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
    (car(stmt) = 'proc-call-mg)
   ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
\land ok-mg-def-plistp (proc-list)
\land ok-translation-parameters (cinfo, t-cond-list, stmt, proc-list, code2)
\land 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)
\wedge
    listp (ctrl-stk)
\wedge
    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)
\land normal (mq\text{-}state)
\land all-cars-unique (mg-alist (mg-state))
   (\neg \text{ resource-errorp (mg-meaning-r }) stmt,
                                          proc-list,
                                          mg-state,
                                          n,
                                          list (length (temp-stk),
                                               p-ctrl-stk-size(ctrl-stk))))))
    mg-vars-list-ok-in-p-state (mg-alist (mg-meaning-r (def-body (fetch-called-def (stmt,
                                                                                        proc-list)),
                                                           proc-list,
                                                           mg-state (cc (mg-state),
                                                                      make-call-var-alist (mg-alist (mg-state),
                                                                                            stmt.
                                                                                            fetch-called-def (stmt,
                                                                                                             proc-lis
                                                                      mg\text{-psw}(mg\text{-}state)),
                                                           n-1,
                                                           list (data-length (def-locals (fetch-called-def (stmt,
                                                                                                           proc-list)
                                                                     length(temp-stk),
                                                                2
                                                                    length (def-locals (fetch-called-def (stmt,
                                                                    length (def-formals (fetch-called-def (stmt,
```

```
+ p-ctrl-stk-size (ctrl-stk)))),
                             append (map-call-locals (def-locals (fetch-called-def (stmt,
                                                                            proc-list)),
                                                   length(temp-stk)),
                                     map-call-formals (def-formals (fetch-called-def (stmt,
                                                                               proc-list)),
                                                     call-actuals (stmt),
                                                     bindings (top (ctrl-stk)))),
                             append (reverse (mg-to-p-local-values (def-locals (fetch-called-def (stmt,
                                                                                        proc-list)))),
                                     map-down-values (mg-alist (mg-state),
                                                      bindings (top (ctrl-stk)),
                                                      temp-stk)))
;; (pop* (data-length locals))
;; (pop* (data-length locals))
(prove-lemma call-state2-step2-effect (rewrite)
      (IMPLIES
       (AND (NOT (ZEROP N))
    (NOT (RESOURCES-INADEQUATEP STMT PROC-LIST
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK))))
    (EQUAL (CAR STMT) 'PROC-CALL-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)
    (EQUAL (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)
```

proc-list

```
(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))
    (NOT (RESOURCE-ERRORP (MG-MEANING-R STMT PROC-LIST MG-STATE N
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK))))))
     (equal
       (p-step
 (P-STATE (TAG 'PC
     (CONS
      (CALL-NAME STMT)
      (PLUS
       (LENGTH
(CODE
 (TRANSLATE
  (MAKE-CINFO NIL
     (CONS
       '(ROUTINEERROR . 0)
       (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
0))
     1)
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST)))
      1)))
(CONS
 (P-FRAME
 (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (LENGTH TEMP-STK))
  (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
    (CALL-ACTUALS STMT)
    (BINDINGS (TOP CTRL-STK))))
  (TAG 'PC
       (CONS SUBR
     (ADD1 (PLUS (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
CTRL-STK)
(MAP-DOWN-VALUES
 (MG-ALIST
```

```
(MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST
  (MG-STATE (CC MG-STATE)
     (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 STMT
  (FETCH-CALLED-DEF STMT PROC-LIST))
     (MG-PSW MG-STATE))
  (SUB1 N)
  (LIST (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
       (LENGTH TEMP-STK))
 (PLUS 2
       (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
       (LENGTH (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST)))
       (P-CTRL-STK-SIZE CTRL-STK)))))
 (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (LENGTH TEMP-STK))
 (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
   (CALL-ACTUALS STMT)
   (BINDINGS (TOP CTRL-STK))))
 (APPEND
  (REVERSE
  (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
  (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
  (BINDINGS (TOP CTRL-STK))
  TEMP-STK)))
(TRANSLATE-PROC-LIST PROC-LIST)
(LIST
 (LIST 'C-C
       (MG-COND-TO-P-NAT
(CC
 (MG-MEANING-R
 (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
 PROC-LIST
 (MG-STATE (CC MG-STATE)
   (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 (FETCH-CALLED-DEF STMT PROC-LIST))
    (MG-PSW MG-STATE))
  (SUB1 N)
 (LIST
  (PLUS
   (LENGTH
     (MG-TO-P-LOCAL-VALUES
```

```
(DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
    (LENGTH (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
    TEMP-STK)))
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (APPEND
       (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
(LENGTH TEMP-STK))
       (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
 (CALL-ACTUALS STMT)
 (BINDINGS (TOP CTRL-STK))))
      (TAG 'PC
   (CONS SUBR
 (ADD1
  (PLUS
   (LENGTH (CODE CINFO))
   (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
   (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
(MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
(MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
'RUN))
        (P-STATE
 (TAG 'PC
      (CONS
       (CALL-NAME STMT)
       (PLUS
(LENGTH
 (CODE
  (TRANSLATE
   (MAKE-CINFO NIL
       (CONS
'(ROUTINEERROR . 0)
(MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
 0))
       1)
   (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
   (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST)))
2)))
 (CONS
```

```
(P-FRAME
  (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
   (LENGTH TEMP-STK))
  (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
     (CALL-ACTUALS STMT)
     (BINDINGS (TOP CTRL-STK))))
  (TAG 'PC
(CONS SUBR
      (ADD1 (PLUS (LENGTH (CODE CINFO))
  (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
  (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
  (LENGTH (CALL-ACTUALS STMT)))))))
 CTRL-STK)
 (POPN (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
  (MAP-DOWN-VALUES
  (MG-ALIST
   (MG-MEANING-R
    (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
    PROC-LIST
    (MG-STATE (CC MG-STATE)
       (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
    (FETCH-CALLED-DEF STMT PROC-LIST))
       (MG-PSW MG-STATE))
     (SUB1 N)
    (LIST (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH TEMP-STK))
  (PLUS 2
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (P-CTRL-STK-SIZE CTRL-STK)))))
  (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
   (LENGTH TEMP-STK))
   (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
     (CALL-ACTUALS STMT)
     (BINDINGS (TOP CTRL-STK))))
   (APPEND
    (REVERSE
     (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
    (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
    (BINDINGS (TOP CTRL-STK))
    TEMP-STK))))
 (TRANSLATE-PROC-LIST PROC-LIST)
 (LIST
```

```
(LIST 'C-C
(MG-COND-TO-P-NAT
  (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST
  (MG-STATE (CC MG-STATE)
     (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 STMT
  (FETCH-CALLED-DEF STMT PROC-LIST))
     (MG-PSW MG-STATE))
  (SUB1 N)
  (LIST
   (PLUS
     (LENGTH
      (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (LENGTH (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
   (P-CTRL-STK-SIZE
     (CONS
     (P-FRAME
       (APPEND
(MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
 (LENGTH TEMP-STK))
(MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (CALL-ACTUALS STMT)
  (BINDINGS (TOP CTRL-STK))))
       (TAG 'PC
   (CONS SUBR
  (ADD1
   (PLUS (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
         (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
 'RUN)))
 ((INSTRUCTIONS PROMOTE
  (DIVE 1)
  Х
  (S LEMMAS)
  (DIVE 1 1 2)
```

```
(REWRITE TRANSLATE-DEF-BODY-REWRITE
 (($CINFO
   (MAKE-CINFO NIL
    (CONS
        (CONS 'ROUTINEERROR O)
        (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                          0))
    1))
  ($T-COND-LIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))
  ($CODE2
   (LIST '(DL O NIL (NO-OP))
         (LIST 'POP*
               (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
         '(RET)))
  ($STMT (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST)))))
(REWRITE GET-LENGTH-PLUS)
(S LEMMAS)
UP X UP X
(S LEMMAS)
(DIVE 1)
X UP S
(S LEMMAS)
UP S-PROP
(DIVE 1)
(DIVE 1)
(REWRITE MAP-DOWN-VALUES-PRESERVES-LENGTH)
(S LEMMAS)
(DIVE 1)
(REWRITE LENGTH-MG-TO-P-LOCAL-VALUES)
UP UP
(= * F)
(REWRITE CALLED-DEF-FORMALS-OK)
(REWRITE CALL-BODY-MG-VARS-LIST-OK1)
(DIVE 1 1)
(REWRITE MG-MEANING-EQUIVALENCE)
UP UP
(REWRITE MG-MEANING-PRESERVES-MG-ALISTP
        (($R-COND-LIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))
         ($NAME-ALIST (MAKE-NAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
(DIVE 1)
(= *
   (MAKE-CALL-ENVIRONMENT MG-STATE STMT
```

```
(FETCH-CALLED-DEF STMT PROC-LIST))
   0)
(REWRITE PROC-CALL-EXACT-TIME-HYPS)
S PROVE
(REWRITE CALL-EXACT-TIME-HYPS1)
(DIVE 1)
S UP
(REWRITE CALL-SIGNATURES-MATCH2)
(DIVE 1)
(REWRITE MORE-RESOURCES-PRESERVES-NOT-RESOURCE-ERRORP
 (($T-SIZE1
   (LENGTH
    (APPEND
     (REVERSE
      (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
                      (BINDINGS (TOP CTRL-STK))
                      TEMP-STK))))
  ($C-SIZE1
   (P-CTRL-STK-SIZE
    (CONS
     (P-FRAME
      (MAKE-FRAME-ALIST (FETCH-CALLED-DEF STMT PROC-LIST)
                        STMT CTRL-STK TEMP-STK)
      (TAG 'PC
       (CONS SUBR
        (ADD1
          (PLUS (LENGTH (CODE CINFO))
                (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
                (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
UP S
(S LEMMAS)
(DIVE 1 1 1)
(REWRITE LENGTH-MG-TO-P-LOCAL-VALUES)
(REWRITE MAP-DOWN-VALUES-PRESERVES-LENGTH)
TOP DROP PROVE
(REWRITE OK-MG-STATEP-MG-ALIST-MG-ALISTP)
(REWRITE CALLED-DEF-FORMALS-OK)
(DIVE 1 1)
Χ
```

```
(S LEMMAS)
   (S-PROP P-FRAME-SIZE)
   (S LEMMAS)
  UP TOP DROP PROVE
   (DIVE 1 1 3)
   (= *
      (MAKE-CALL-ENVIRONMENT MG-STATE STMT
                             (FETCH-CALLED-DEF STMT PROC-LIST))
     0)
  TOP
   (DIVE 1)
   (REWRITE PROC-CALL-DOESNT-HALT)
   TOP S
   (S-PROP MAKE-CALL-ENVIRONMENT)
   (PROVE (ENABLE OK-MG-STATEMENT OK-PROC-CALL))
   (REWRITE PROC-CALL-EXACT-TIME-HYPS)
   (DIVE 1 1)
  X TOP
   (PROVE (ENABLE FETCH-CALLED-DEF FETCH-DEF)))))
;; (ret)
;; (ret)
(prove-lemma call-state2-step3-effect (rewrite)
      (IMPLIES
       (AND (NOT (ZEROP N))
    (NOT (RESOURCES-INADEQUATEP STMT PROC-LIST
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK))))
    (EQUAL (CAR STMT) 'PROC-CALL-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)
    (EQUAL (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))
    (NOT (RESOURCE-ERRORP (MG-MEANING-R STMT PROC-LIST MG-STATE N
(LIST (LENGTH TEMP-STK)
      (P-CTRL-STK-SIZE CTRL-STK))))))
     (equal
       (p-step
        (P-STATE
 (TAG 'PC
      (CONS
       (CALL-NAME STMT)
       (PLUS
(LENGTH
 (CODE
  (TRANSLATE
   (MAKE-CINFO NIL
       (CONS
'(ROUTINEERROR . 0)
(MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
 0))
       1)
   (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
   (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST)))
2)))
 (CONS
  (P-FRAME
   (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
    (LENGTH TEMP-STK))
   (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
     (CALL-ACTUALS STMT)
     (BINDINGS (TOP CTRL-STK))))
   (TAG 'PC
(CONS SUBR
      (ADD1 (PLUS (LENGTH (CODE CINFO))
```

```
(DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
 CTRL-STK)
 (POPN (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (MAP-DOWN-VALUES
  (MG-ALIST
   (MG-MEANING-R
    (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
    PROC-LIST
    (MG-STATE (CC MG-STATE)
      (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
   STMT
   (FETCH-CALLED-DEF STMT PROC-LIST))
       (MG-PSW MG-STATE))
    (SUB1 N)
    (LIST (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH TEMP-STK))
  (PLUS 2
(LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (P-CTRL-STK-SIZE CTRL-STK))))
  (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
   (LENGTH TEMP-STK))
  (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
    (CALL-ACTUALS STMT)
    (BINDINGS (TOP CTRL-STK))))
  (APPEND
   (REVERSE
    (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
    (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
    (BINDINGS (TOP CTRL-STK))
    TEMP-STK))))
 (TRANSLATE-PROC-LIST PROC-LIST)
 (LIST
 (LIST 'C-C
(MG-COND-TO-P-NAT
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST
  (MG-STATE (CC MG-STATE)
    (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 STMT
```

```
(FETCH-CALLED-DEF STMT PROC-LIST))
     (MG-PSW MG-STATE))
   (SUB1 N)
  (LIST
    (PLUS
     (LENGTH
     (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (LENGTH (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
    (P-CTRL-STK-SIZE
     (CONS
      (P-FRAME
       (APPEND
(MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
 (LENGTH TEMP-STK))
(MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (CALL-ACTUALS STMT)
  (BINDINGS (TOP CTRL-STK))))
       (TAG 'PC
   (CONS SUBR
  (ADD1
   (PLUS (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
 (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
 'RUN))
      (P-STATE
          (TAG 'PC
       (CONS SUBR
     (ADD1 (PLUS (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT))))))
        ctrl-stk
 (POPN (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (MAP-DOWN-VALUES
  (MG-ALIST
    (MG-MEANING-R
     (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
    PROC-LIST
```

```
(MG-STATE (CC MG-STATE)
      (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
   (FETCH-CALLED-DEF STMT PROC-LIST))
      (MG-PSW MG-STATE))
     (SUB1 N)
     (LIST (PLUS (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH TEMP-STK))
  (PLUS 2
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
(LENGTH (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (P-CTRL-STK-SIZE CTRL-STK)))))
  (APPEND (MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
   (LENGTH TEMP-STK))
  (MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
     (CALL-ACTUALS STMT)
     (BINDINGS (TOP CTRL-STK))))
  (APPEND
   (REVERSE
    (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
   (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
    (BINDINGS (TOP CTRL-STK))
    TEMP-STK))))
(TRANSLATE-PROC-LIST PROC-LIST)
(LIST
 (LIST 'C-C
(MG-COND-TO-P-NAT
(CC
 (MG-MEANING-R
  (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST))
  PROC-LIST
  (MG-STATE (CC MG-STATE)
    (MAKE-CALL-VAR-ALIST (MG-ALIST MG-STATE)
 STMT
 (FETCH-CALLED-DEF STMT PROC-LIST))
    (MG-PSW MG-STATE))
  (SUB1 N)
  (LIST
   (PLUS
    (LENGTH
     (MG-TO-P-LOCAL-VALUES (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
     (LENGTH (MAP-DOWN-VALUES (MG-ALIST MG-STATE)
     (BINDINGS (TOP CTRL-STK))
     TEMP-STK)))
```

```
(P-CTRL-STK-SIZE
     (CONS
      (P-FRAME
       (APPEND
(MAP-CALL-LOCALS (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))
(LENGTH TEMP-STK))
(MAP-CALL-FORMALS (DEF-FORMALS (FETCH-CALLED-DEF STMT PROC-LIST))
  (CALL-ACTUALS STMT)
  (BINDINGS (TOP CTRL-STK))))
       (TAG 'PC
   (CONS SUBR
  (ADD1
  (PLUS (LENGTH (CODE CINFO))
 (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST)))
 (LENGTH (CALL-ACTUALS STMT)))))))
     CTRL-STK)))))
 (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))))
 (MG-MAX-CTRL-STK-SIZE) (MG-MAX-TEMP-STK-SIZE) (MG-WORD-SIZE)
 'RUN)))
 ((INSTRUCTIONS PROMOTE
  (DIVE 1)
  Х
  (S LEMMAS)
  (DIVE 1 1 2)
   (REWRITE TRANSLATE-DEF-BODY-REWRITE
   (($CINFO
      (MAKE-CINFO NIL
       (CONS
           (CONS 'ROUTINEERROR 0)
           (MAKE-LABEL-ALIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST))
                             0))
     ($T-COND-LIST (MAKE-COND-LIST (FETCH-CALLED-DEF STMT PROC-LIST)))
     ($CODE2
      (LIST
       '(DL O NIL (NO-OP))
       (LIST 'POP* (DATA-LENGTH (DEF-LOCALS (FETCH-CALLED-DEF STMT PROC-LIST))))
       '(RET)))
     ($STMT (DEF-BODY (FETCH-CALLED-DEF STMT PROC-LIST)))))
  (REWRITE GET-LENGTH-PLUS)
  (S LEMMAS)
  UP X UP X
```

```
(S LEMMAS)
X
(S LEMMAS)
UP prove
(PROVE (ENABLE OK-MG-STATEMENT OK-PROC-CALL))
(DIVE 1 1)
X TOP
(PROVE (ENABLE FETCH-CALLED-DEF FETCH-DEF)))))
```

EVENT: Make the library "c-proc-call1".

Index

IIIdeA	
actual-params-list-ok-in-mg-ali st, 5 actual-pointers-distinct, 4 actual-pointers-distinct2, 4 all-cars-unique, 3–9, 12, 15, 22, 24, 26–30, 61 all-pointers-bigger, 3 all-pointers-smaller, 3 array-alist-element-lengths-mat ch, 4	call-push-parameters-effect 1, 20 call-step-initial-equals-state 1, 30 call-steps-to-body, 24 call-translation-2, 3 cc, 21, 23–25, 31, 61 code, 3, 7, 9, 10, 12–15, 21–25, 29–32, 61 collect-pointers, 3–5 cond-subsetp, 9, 12, 13, 22, 24, 29, 30, 61
array-formal-ok-for-actual, 8 array-formal-ok-for-actual2, 28 array-identifierp, 1 array-length, 4, 5, 8, 18, 27, 28 ascending-local-address-sequence, 18, 20, 21, 23, 25, 31 bindings, 6, 9–15, 19–25, 29–32, 61,	data-length, 2, 7, 11, 13–15, 17, 18, 20–25, 31, 32, 61 data-param-lists-match, 3–5, 8, 26, 28 data-param-lists-match-in-mg-ali st, 5 data-params-match, 8, 28
62 c-size, 2 call-actuals, 5, 10, 11, 14, 15, 21–23, 25, 31, 32, 62	def-body, 2, 7, 10–13, 15, 61 def-cond-locals, 3 def-formals, 3, 5, 6, 9, 11, 62 def-locals, 2, 3, 6, 7, 9–11, 13–15, 21–25, 29–32, 61, 62
call-body-mg-vars-list-ok1, 60 call-body-rewrite, 7 call-call-step, 22 call-conds, 3 call-environment-mg-vars-list-o k1, 28	defined-identifierp, 4, 8 definedp, 1, 4, 7, 8, 28 definedp-caar, 7 deposit-alist-value, 26, 27 disjoint, 5
call-exact-time-translation-par ameters-ok, 6 call-formal-var-lists-match, 26 call-local-names-unique, 6 call-local-var-lists-match1, 25 call-name, 7, 13, 23, 25, 30–32 call-push-actuals-effect, 19 call-push-actuals-induction-hint, 19 call-push-locals-addresses-effe ct, 18 call-push-locals-values-effect, 17	exp, 1 extra-bindings-doesnt-affect-fo rmal-types-preserved, 1 extra-bindings-dont-affect-no-p -aliasing, 4 -aliasing2, 4 fetch-called-def, 2, 3, 5–7, 9–15, 21– 25, 29–32, 61, 62 fetch-label-0-case-2, 1 formal-types-preserved, 1
call-push-parameters-effect, 20	label-alist, 3

```
label-cnt, 3
                                             mg-actuals-to-p-actuals, 19-21, 23,
length, 1, 3, 4, 8-25, 27, 29-32, 61,
                                                      25, 26, 31
                                             mg-alist, 2, 5, 6, 9–15, 21–25, 29–32,
                                                      61, 62
length-plistp, 7
listcars, 1, 4, 6, 25, 26
                                             mg-alistp, 1, 3–5, 8, 27, 28
locals-addresses-induction-hint, 17,
                                             mg-cond-to-p-nat, 1, 21, 23-25, 31
                                             mg-cond-to-p-nat-index-lessp, 1
locals-pointers-bigger0, 3
                                             mg-cond-to-p-nat-p-objectp-type
locals-values-induction-hint, 17
                                                 -nat. 1
                                             mg-formals-list-ok-in-call-envi
make-call-environment, 2, 10, 11, 13-
                                                 ronment0, 8
         15.31
                                                 ronment1, 8
make-call-param-alist, 3-5, 8, 9, 26,
                                             mg-formals-list-ok-induction-hi
         28, 29
                                                 nt, 8
make\text{-call-var-alist},\,6,\,9,\,30,\,61
                                             mg-locals-list-ok-in-call-envir
make-cinfo, 3, 6, 7, 12, 13, 32
                                                 onment, 7
make-cond-list, 6, 7, 12, 13, 32
                                             mg-locals-list-ok-induction-hint, 7
make-frame-alist, 6, 10, 13–15, 30,
                                             mg-max-ctrl-stk-size, 21, 23-25, 31
         31
                                             mg-max-temp-stk-size, 16-20, 22-25,
make-label-alist, 1, 6, 7, 12, 13, 32
make-name-alist, 12, 15
                                             mg-meaning, 2
map-call-effects, 3
                                             mg-meaning-preserves-signatures
map-call-formals, 3-5, 8-10, 26, 28,
                                                 -match2, 1
         29, 62
                                             mg-meaning-r, 2, 3, 10-12, 15, 22,
map-call-formals-all-pointers-s
                                                      24, 29, 30, 61, 62
    maller3, 3
                                             mg-psw, 61
map-call-locals, 3, 8, 9, 25–28, 62
                                             mg-state, 2, 61
map-down, 21, 24, 32
                                             mg-to-p-local-values, 8, 10, 13–15,
map-down-again-preserves-values, 28
                                                      17, 20, 21, 23, 25, 28–32,
map-down-formals-doesnt-affect-
    locals, 27
                                             mg-to-p-simple-literal, 7, 16, 17, 27
    locals-induction-hint, 26, 27
                                             mg-to-p-simple-literal-list, 7, 16, 17,
map-down-locals-doesnt-affect-f
                                                      27
    ormals, 26
                                             mg-vars-list-ok-in-call-environ
    ormals-induction-hint, 26
                                                 ment, 9
map-down-locals-equals-reverse-v
                                             mg-vars-list-ok-in-p-state, 1, 3, 5, 6,
    alues, 27
                                                      8-12, 14, 21, 22, 24, 27-30,
map-down-locals-induction-hint, 27
                                                      61, 62
map-down-preserves-references, 29
                                             mg-word-size, 1, 22–25, 31
map-down-skips-non-referenced-se
    gment, 27
                                             n-successive-pointers, 5
map-down-values, 1, 9, 10, 13–15,
                                             no-duplicates, 4–6
         21, 23, 25-32, 62
                                             no-p-aliasing, 3-6, 9, 12, 15, 22, 24,
                                                      28-30, 61
```

no-p-aliasing-formals, 5	push-array-value-induction-hint, 16
no-p-aliasing-in-call-environme	push-local-array-values-code, 16, 17
nt, 6	push-local-array-values-code-ef
no-p-aliasing-locals, 3	fect, 16
normal, 2, 9–12, 15, 22, 24, 29, 30,	push-locals-addresses-code, 18
61	push-locals-values-code, 17
not-simple-identifiers-array-ide	push-parameters-code, 20
ntifiers, 1	pasii parameters code, 20
numers, 1	resource-errorp, 10–12, 15, 22, 24,
ok-actual-params-list, 3–5, 8, 28	29, 30, 61
ok-mg-def-plistp, 6, 9–12, 20, 22, 24,	resources-inadequatep, 2, 9, 12, 21,
29, 30, 61	22, 24, 29, 30, 61
ok-mg-formal-data-params-plistp, 4,	reverse, 7, 8, 10, 13–21, 23, 25, 27–
5, 8, 28	32, 62
ok-mg-local-data-plistp, 7, 17, 18,	92, 02
20, 28	set-alist-value, 1
ok-mg-statement, 5, 6, 9–12, 20, 22,	set-alist-value-map-down-values
24, 29, 30, 61	-length-doesnt-shrink, 1
ok-mg-statep, 5, 6, 9–13, 21, 22, 24,	signal-system-error, 2
29, 30, 61	signatures-match, 2, 5, 6, 9, 12, 15,
ok-temp-stk-array-index, 8, 28	22, 24, 29, 30, 61
ok-temp-stk-index, 8	simple-formal-ok-for-actual2, 8
ok-translation-parameters, 6, 7, 9,	simple-identifierp, 1
12, 13, 22, 24, 29, 30, 61	simple-mg-type-refp, 4, 5, 7, 8, 16-
12, 10, 22, 24, 25, 50, 61	18, 26–28
p, 16–21, 25	simple-typed-literal-listp, 16
p-ctrl-stk-size, 9–12, 15, 21, 22, 24,	simple-typed-literal-plistp, 16
29, 30, 61, 62	simple-typed-literalp, 16
p-frame, 11, 14, 15, 23, 25, 31, 32	
p-objectp-type, 1	t-size, 2
p-state, 16–20, 22–25, 31	tag, 7, 11, 14-25, 30-32
p-step, 23	tag-length-plistp-2, 7
p-word-size, 1	top,6,915,1925,2932,61,62
pair-temps-with-initial-values, 23, 25,	translate, 3, 7, 9, 12, 13, 21, 22, 24,
31	29, 30, 61
plistp, 1, 6, 9, 12, 13, 22, 24, 28–30,	translate-def-body, 7, 9, 12, 13, 21,
61	22, 24, 29, 30, 61
proc-call-code, 3	translate-proc-list, 21 , 23 , 25 , 31
proc-call-doesnt-halt, 10	
proc-call-doesnt-halt2, 11	untag, 1, 4
proc-call-exact-time-hyps, 11	user-defined-procp, 6, 9, 12, 13, 21,
proc-call-meaning-r-2, 2	22, 24, 29, 30, 61
push, 16–18, 23, 25, 31	
nuch actuals code 10	

push-actuals-code, 19