EVENT: Start with the library "c-predefined2".

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                            ;;
;;                       EXACT-TIME LEMMA MG-INTEGER-ADD                       ;;
;;                                                                            ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

THEOREM: mg-integer-add-args-have-simple-mg-type-refps
((car (*stmt*) = `'predefined-proc-call-mg`)
 ∧ (call-name (*stmt*) = `'mg-integer-add`)
 ∧ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
 ∧ ok-mg-statep (*mg-state*, *r-cond-list*)
 ∧ signatures-match (mg-alist (*mg-state*), *name-alist*))
→ (int-identifierp (car (call-actuals (*stmt*)), mg-alist (*mg-state*))
  ∧ int-identifierp (cadr (call-actuals (*stmt*)), mg-alist (*mg-state*))
  ∧ int-identifierp (caddr (call-actuals (*stmt*)), mg-alist (*mg-state*)))

THEOREM: mg-integer-add-args-definedp
((car (*stmt*) = `'predefined-proc-call-mg`)
 ∧ (call-name (*stmt*) = `'mg-integer-add`)
 ∧ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
 ∧ ok-mg-statep (*mg-state*, *r-cond-list*)
 ∧ signatures-match (mg-alist (*mg-state*), *name-alist*))
→ (definedp (car (call-actuals (*stmt*)), mg-alist (*mg-state*))
  ∧ definedp (cadr (call-actuals (*stmt*)), mg-alist (*mg-state*))
  ∧ definedp (caddr (call-actuals (*stmt*)), mg-alist (*mg-state*)))

THEOREM: mg-integer-add-steps-1-3
(($n \not\simeq 0$)
 ∧ (¬ resources-inadequatep (*stmt*,
   proc-list,
   list (length (*temp-stk*),
   p-ctrl-stk-size (*ctrl-stk*))))
 ∧ (car (*stmt*) = `'predefined-proc-call-mg`)
 ∧ (call-name (*stmt*) = `'mg-integer-add`)
 ∧ 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))
 ∧ user-defined-procp (*subr*, *proc-list*)

```
 ∧   listp (ctrl-stk)
 ∧   all-cars-unique (mg-alist (mg-state))
 ∧   signatures-match (mg-alist (mg-state), name-alist)
 ∧   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))
 ∧   normal (mg-state))
 →   (p-step (p-step (p-step (map-down (mg-state,
                                        proc-list,
                                        ctrl-stk,
                                        temp-stk,
                                        tag ('pc,
                                             cons (subr, length (code (cinfo)))),
                                        t-cond-list))))
       =   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                    ctrl-stk,
                    push (value (caddr (call-actuals (stmt)),
                                 bindings (top (ctrl-stk))),
                          push (value (cadr (call-actuals (stmt)),
                                       bindings (top (ctrl-stk))),
                                push (value (car (call-actuals (stmt)),
                                             bindings (top (ctrl-stk))),
                                      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))
```

THEOREM: mg-integer-add-step-4

```
((n ≄ 0)
 ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
 ∧   (car (stmt) = 'predefined-proc-call-mg)
 ∧   (call-name (stmt) = 'mg-integer-add)
 ∧   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))
 ∧   user-defined-procp (subr, proc-list)
 ∧   listp (ctrl-stk)
 ∧   all-cars-unique (mg-alist (mg-state))
 ∧   signatures-match (mg-alist (mg-state), name-alist)
 ∧   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))
 ∧   normal (mg-state))
 →   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                       ctrl-stk,
                       push (value (caddr (call-actuals (stmt)),
                                    bindings (top (ctrl-stk))),
                             push (value (cadr (call-actuals (stmt)),
                                          bindings (top (ctrl-stk))),
                                   push (value (car (call-actuals (stmt)),
                                                bindings (top (ctrl-stk))),
                                         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))
       =   p-state (tag ('pc, '(mg-integer-add . 0)),
                    push (p-frame (cons (cons ('ans,
                                               value (car (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk)))),
                                         cons (cons ('y,
                                                     value (cadr (call-actuals (stmt)),
                                                            bindings (top (ctrl-stk)))),
                                               cons (cons ('z,
                                                           value (caddr (call-actuals (stmt)),
                                                                  bindings (top (ctrl-stk)))),
                                                     '((t1 int 0))))),
                                   tag ('pc,
                                        cons (subr, length (code (cinfo))
                                              +  4))),
                    ctrl-stk),
                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))
```

THEOREM: mg-integer-add-steps-5-9

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-add)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→  (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                          '(mg-integer-add
                                                            . 0)),
                                                     push (p-frame (cons (cons ('ans,
                                                                                value (car (call-actuals (stmt)),
                                                                                       bindings (top (ctrl-stk)))),
                                                                          cons (cons ('y,
                                                                                      value (cadr (call-actuals (st
                                                                                                    bindings (top (ctrl-s
                                                                                     cons (cons ('z,
                                                                                                   value (caddr (call-act
                                                                                                           bindings (top (
                                                                                            '((t1
                                                                                                 int
                                                                                                 0))))),
                                                                        tag ('pc,
                                                                             cons (subr,
                                                                                    length (code (cinfo))
                                                                                    +   4))),
                                                             ctrl-stk),
                                                      map-down-values (mg-alist (mg-state),
                                                                            bindings (top (ctrl-stk)),
                                                                            temp-stk),
                                                      translate-proc-list (proc-list),
                                                      list (list ('c-c,
                                                                996mg-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, '(mg-integer-add . 5)),
               push (p-frame (cons (cons ('ans,
                                              value (car (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                        cons (cons ('y,
                                                     value (cadr (call-actuals (stmt)),
                                                             bindings (top (ctrl-stk)))),
                                              cons (cons ('z,
                                                            value (caddr (call-actuals (stmt)),
                                                                   bindings (top (ctrl-stk)))),
                                                     '((t1 int 0))))),
                                 tag ('pc,
                                      cons (subr, length (code (cinfo))
                                                    +   4))),
                      ctrl-stk),
               push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                 mg-alist (mg-state)))),
                      push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                         mg-alist (mg-state)))),
                             push ('(bool f),
                                    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))
```

THEOREM: mg-integer-add-step-10-nonerror

```
 ((n ≄ 0)
  ∧  (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
  ∧  (car (stmt) = 'predefined-proc-call-mg)
  ∧  (call-name (stmt) = 'mg-integer-add)
  ∧  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))
  ∧  user-defined-procp (subr, proc-list)
  ∧  listp (ctrl-stk)
  ∧  all-cars-unique (mg-alist (mg-state))
  ∧  signatures-match (mg-alist (mg-state), name-alist)
  ∧  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))
  ∧  normal (mg-state)
  ∧  small-integerp (iplus (0,
                      iplus (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                          mg-alist (mg-state))))),
                             untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                          mg-alist (mg-state))))))),
                 MG-WORD-SIZE))
 →   (p-step (p-state (tag ('pc, '(mg-integer-add . 5)),
                  push (p-frame (cons (cons ('ans,
                                             value (car (call-actuals (stmt)),
                                                    bindings (top (ctrl-stk)))),
                                       cons (cons ('y,
                                                         value (cadr (call-actuals (stmt)),
                                                                 bindings (top (ctrl-stk)))),
                                                  cons (cons ('z,
                                                               value (caddr (call-actuals (stmt)),
                                                                       bindings (top (ctrl-stk)))),
                                                         '((t1 int 0))))),
                                    tag ('pc,
                                          cons (subr, length (code (cinfo))
                                                        +   4))),
                          ctrl-stk),
                   push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                     mg-alist (mg-state)))),
                          push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                             mg-alist (mg-state)))),
                                 push ('(bool f),
                                        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))
=    p-state (tag ('pc, '(mg-integer-add . 6)),
               push (p-frame (cons (cons ('ans,
                                              value (car (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk)))),
                                        cons (cons ('y,
                                                     value (cadr (call-actuals (stmt)),
                                                             bindings (top (ctrl-stk)))),
                                              cons (cons ('z,
                                                            value (caddr (call-actuals (stmt)),
                                                                    bindings (top (ctrl-stk)))),
                                                     '((t1 int 0))))),
                                 tag ('pc,
                                      cons (subr, length (code (cinfo))
                                                    +   4))),
                      ctrl-stk),
               push (tag ('int,
                           fix-small-integer (iplus (0,
                                                       iplus (untag (mg-to-p-simple-literal (caddr (assoc (cadr (ca
                                                                                                                  mg-alist
```

```
                                                 untag (mg-to-p-simple-literal (caddr (assoc (caddr (c
                                                                                         mg-alist
                                          MG-WORD-SIZE)),
                       push (tag ('bool, 'f),
                             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))
```

THEOREM: mg-integer-add-steps-11-17-nonerror

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-add)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p-step (p-step (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                                         '(mg-integer-add
                                                                           . 6)),
                                                                    push (p-frame (cons (cons ('ans,
                                                                                               value (car (call-ac
                                                                                                      bindings (t
                                                                                        cons (cons ('y,
                                                                                                    value (cadr
                                                                                                              bindi
                                                                                              cons (cons ('z,
                                                                                                          value

                                                                                                    '((t1
                                                                                                        int
                                                                                                        0))))),
                                                                                  tag ('pc,
                                                                                      cons (subr,
                                                                                            length (code (cinfo
                                                                                            +   4))),
                                                                        ctrl-stk),
                                                                 push (sum,
                                                                        push (tag ('bool,
                                                                                   'f),
                                                                              map-down-values (mg-alist (mg-
                                                                                                  bindings (top
                                                                                                  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 (subr, length (code (cinfo)) + 4)),
              ctrl-stk,
              rput (sum,
                    untag (value (car (call-actuals (stmt)),
                                  bindings (top (ctrl-stk)))),
                    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))
```

THEOREM: mg-integer-add-push-cc

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-add)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  normal (mg-state))
→  (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 4)),
                     ctrl-stk,
                     temp-stk,
                     translate-proc-list (proc-list),
                     list (list ('c-c, cc-value)),
                     MG-MAX-CTRL-STK-SIZE,
                     MG-MAX-TEMP-STK-SIZE,
                     MG-WORD-SIZE,
                     'run))
     =  p-state (tag ('pc, cons (subr, length (code (cinfo)) + 5)),
                 ctrl-stk,
                 push (cc-value, temp-stk),
                 translate-proc-list (proc-list),
                 list (list ('c-c, cc-value)),
                 MG-MAX-CTRL-STK-SIZE,
                 MG-MAX-TEMP-STK-SIZE,
                 MG-WORD-SIZE,
                 'run))
```

THEOREM: mg-integer-add-sub1-cc

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-add)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  normal (mg-state)
 ∧  (cc-value ∈ list ('(nat 1), '(nat 2))))
→   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 5)),
                      ctrl-stk,
                      push (cc-value, temp-stk),
                      translate-proc-list (proc-list),
                      list (list ('c-c, cc-value)),
                      MG-MAX-CTRL-STK-SIZE,
                      MG-MAX-TEMP-STK-SIZE,
                      MG-WORD-SIZE,
                      'run))
     =  p-state (tag ('pc, cons (subr, length (code (cinfo)) + 6)),
                 ctrl-stk,
                 push (tag ('nat, untag (cc-value) − 1), temp-stk),
                 translate-proc-list (proc-list),
                 list (list ('c-c, cc-value)),
                 MG-MAX-CTRL-STK-SIZE,
                 MG-MAX-TEMP-STK-SIZE,
                 MG-WORD-SIZE,
                 'run))
```

THEOREM: mg-integer-add-step-20-nonerror

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-add)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state)
 ∧  small-integerp (iplus (0,
                           iplus (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                               mg-alist (mg-state))))),
                                  untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                               mg-alist (mg-state))))))),
                    MG-WORD-SIZE))
 →  (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 6)),
                      ctrl-stk,
                      push (tag ('nat,
                                 untag (mg-cond-to-p-nat (cc (mg-state),
                                                          t-cond-list)) − 1),
                            rput (tag ('int,
                                       fix-small-integer (iplus (0,
                                                                 iplus (untag (mg-to-p-simple-literal (caddr (assoc

                                                                        untag (mg-to-p-simple-literal (caddr (assoc

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

THEOREM: mg-integer-add-step-10-error

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-add)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state)
 ∧  (¬ small-integerp (iplus (0,
                            iplus (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                  mg-alist (mg-state))))),
                                   untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                                  mg-alist (mg-state))))))),
                     MG-WORD-SIZE)))
→   (p-step (p-state (tag ('pc, '(mg-integer-add . 5)),
                      push (p-frame (cons (cons ('ans,
                                                 value (car (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk)))),
                                           cons (cons ('y,
                                                       value (cadr (call-actuals (stmt)),
                                                              bindings (top (ctrl-stk)))),
                                                 cons (cons ('z,
                                                             value (caddr (call-actuals (stmt)),
                                                                    bindings (top (ctrl-stk)))),
                                                       '((t1 int 0))))),
                    tag ('pc,
                         cons (subr, length (code (cinfo))
                                     +   4))),
          ctrl-stk),
     push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                  mg-alist (mg-state)))),
          push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                        mg-alist (mg-state)))),
               push ('(bool f),
                    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))
=   p-state (tag ('pc, '(mg-integer-add . 6)),
             push (p-frame (cons (cons ('ans,
                                        value (car (call-actuals (stmt)),
                                               bindings (top (ctrl-stk)))),
                                  cons (cons ('y,
                                              value (cadr (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                        cons (cons ('z,
                                                    value (caddr (call-actuals (stmt)),
                                                           bindings (top (ctrl-stk)))),
                                              '((t1 int 0))))),
                           tag ('pc,
                                cons (subr, length (code (cinfo))
                                            +   4))),
                   ctrl-stk),
             push (tag ('int,
                        fix-small-integer (iplus (0,
                                                  iplus (untag (mg-to-p-simple-literal (caddr (assoc (cadr (ca
                                                                                                      mg-alist
                                                         untag (mg-to-p-simple-literal (caddr (assoc (caddr (c
                                                                                                      mg-alist
                                           MG-WORD-SIZE)),
                   push (tag ('bool, 't),
                        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))
```

THEOREM: mg-integer-add-steps-11-15-error

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-integer-add)
  ∧   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))
  ∧   user-defined-procp (subr, proc-list)
  ∧   listp (ctrl-stk)
  ∧   all-cars-unique (mg-alist (mg-state))
  ∧   signatures-match (mg-alist (mg-state), name-alist)
  ∧   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))
  ∧   normal (mg-state))
 →    (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                             '(mg-integer-add
                                                               . 6)),
                                                        push (p-frame (cons (cons ('ans,
                                                                                   value (car (call-actuals (stmt)),
                                                                                          bindings (top (ctrl-stk)))),
                                                                             cons (cons ('y,
                                                                                         value (cadr (call-actuals (st
                                                                                                bindings (top (ctrl-st
                                                                                   cons (cons ('z,
                                                                                               value (caddr (call-act
                                                                                                      bindings (top (
                                                                    '((t1
                                                                       int
                                                                       0))))),
                                                        tag ('pc,
                                                             cons (subr,
                                                                   length (code (cinfo))
                                                                   +   4))),
                                                   ctrl-stk),
                                              push (sum,
                                                    push (tag ('bool, 't),
                                                          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))))))
    =   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 4)),
                 ctrl-stk,
                 map-down-values (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                  temp-stk),
                 translate-proc-list (proc-list),
                 list (list ('c-c, '(nat 1))),
                 MG-MAX-CTRL-STK-SIZE,
                 MG-MAX-TEMP-STK-SIZE,
                 MG-WORD-SIZE,
                 'run))
```

THEOREM: mg-integer-add-step-18-error

```
((n ≄ 0)
 ∧   (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
 ∧   (car (stmt) = 'predefined-proc-call-mg)
 ∧   (call-name (stmt) = 'mg-integer-add)
 ∧   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))
 ∧   user-defined-procp (subr, proc-list)
 ∧   listp (ctrl-stk)
 ∧   all-cars-unique (mg-alist (mg-state))
 ∧   signatures-match (mg-alist (mg-state), name-alist)
 ∧   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))
 ∧   normal (mg-state)
 ∧   (¬ small-integerp (iplus (0,
                               iplus (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                   mg-alist (mg-state))))),
                                     untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                                   mg-alist (mg-state))))))),
                        MG-WORD-SIZE)))
 →   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 6)),
                      ctrl-stk,
                      push (tag ('nat, untag ('(nat 1)) − 1),
                            map-down-values (mg-alist (mg-state),
                                             bindings (top (ctrl-stk)),
                                             temp-stk)),
                      translate-proc-list (proc-list),
                      '((c-c (nat 1))),
                      MG-MAX-CTRL-STK-SIZE,
                      MG-MAX-TEMP-STK-SIZE,
                      MG-WORD-SIZE,
                      'run))
       =   p-state (tag ('pc,
                         cons (subr,
                               if normal (mg-meaning-r (stmt,
                                                        proc-list,
                                                        mg-state,
                                                        n,
                                                        list (length (temp-stk),
                                                              p-ctrl-stk-size (ctrl-stk))))
                               then length (code (translate (cinfo,
                                                             t-cond-list,
                                                             stmt,
                                                             proc-list)))
                               else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                               proc-list,
                                                                   mg-state,
                                                                   n,
                                                                   list (length (temp-stk),
                                                                         p-ctrl-stk-size (ctrl-stk)))),
                                                      label-alist (translate (cinfo,
                                                                              t-cond-list,
                                                                              stmt,
                                                                              proc-list))),
                                               append (code (translate (cinfo,
                                                                        t-cond-list,
                                                                        stmt,
                                                                        proc-list)),
                                                       code2)) endif)),
                               ctrl-stk,
                               map-down-values (mg-alist (mg-meaning-r (stmt,
                                                                        proc-list,
                                                                        mg-state,
                                                                        n,
                                                                        list (length (temp-stk),
                                                                              p-ctrl-stk-size (ctrl-stk)))),
                                                bindings (top (ctrl-stk)),
                                                temp-stk),
                               translate-proc-list (proc-list),
                               list (list ('c-c,
                                           mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                                               proc-list,
                                                                               mg-state,
                                                                               n,
                                                                               list (length (temp-stk),
                                                                                     p-ctrl-stk-size (ctrl-stk)))),
                                                             t-cond-list))),
                               MG-MAX-CTRL-STK-SIZE,
                               MG-MAX-TEMP-STK-SIZE,
                               MG-WORD-SIZE,
                               'run))
```

Theorem: mg-integer-add-exact-time-lemma

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-integer-add)
```

```
∧  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))
∧  user-defined-procp (subr, proc-list)
∧  listp (ctrl-stk)
∧  all-cars-unique (mg-alist (mg-state))
∧  signatures-match (mg-alist (mg-state), name-alist)
∧  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))
∧  normal (mg-state))
→  (p (map-down (mg-state,
                 proc-list,
                 ctrl-stk,
                 temp-stk,
                 tag ('pc, cons (subr, length (code (cinfo)))),
                 t-cond-list),
      clock (stmt, proc-list, mg-state, n))
     =  p-state (tag ('pc,
                      cons (subr,
                            if normal (mg-meaning-r (stmt,
                                                     proc-list,
                                                     mg-state,
                                                     n,
                                                     list (length (temp-stk),
                                                           p-ctrl-stk-size (ctrl-stk))))
                            then length (code (translate (cinfo,
                                                          t-cond-list,
                                                          stmt,
                                                          proc-list)))
                            else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                             proc-list,
                                                                             mg-state,
                                                                             n,
                                                                             list (length (temp-stk),
                                                                                   p-ctrl-stk-size (ctrl-stk)))),
                                             label-alist (translate (cinfo,
                                                                     t-cond-list,
                                                                     stmt,
                                                                     proc-list))),
                                 append (code (translate (cinfo,
                                                          t-cond-list,
                                                          stmt,
                                                          proc-list)),
                                               code2)) endif)),
              ctrl-stk,
              map-down-values (mg-alist (mg-meaning-r (stmt,
                                                       proc-list,
                                                       mg-state,
                                                       n,
                                                       list (length (temp-stk),
                                                             p-ctrl-stk-size (ctrl-stk)))),
                               bindings (top (ctrl-stk)),
                               temp-stk),
              translate-proc-list (proc-list),
              list (list ('c-c,
                          mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                              proc-list,
                                                              mg-state,
                                                              n,
                                                              list (length (temp-stk),
                                                                    p-ctrl-stk-size (ctrl-stk)))),
                                            t-cond-list))),
              MG-MAX-CTRL-STK-SIZE,
              MG-MAX-TEMP-STK-SIZE,
              MG-WORD-SIZE,
              'run))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                          ;;
;;                 EXACT-TIME LEMMA MG-INTEGER-SUBTRACT                     ;;
;;                                                                          ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
THEOREM: mg-integer-subtract-args-have-simple-mg-type-refps
 ((car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-integer-subtract)
  ∧   ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
  ∧   ok-mg-statep (mg-state, r-cond-list)
  ∧   signatures-match (mg-alist (mg-state), name-alist))
 →    (int-identifierp (car (call-actuals (stmt)), mg-alist (mg-state))
       ∧   int-identifierp (cadr (call-actuals (stmt)), mg-alist (mg-state))
       ∧   int-identifierp (caddr (call-actuals (stmt)), mg-alist (mg-state)))
```

THEOREM: mg-integer-subtract-args-definedp

```
((car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 ∧  ok-mg-statep (mg-state, r-cond-list)
 ∧  signatures-match (mg-alist (mg-state), name-alist))
→  (definedp (car (call-actuals (stmt)), mg-alist (mg-state))
       ∧  definedp (cadr (call-actuals (stmt)), mg-alist (mg-state))
       ∧  definedp (caddr (call-actuals (stmt)), mg-alist (mg-state)))
```

THEOREM: mg-integer-subtract-steps-1-3

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→  (p-step (p-step (p-step (map-down (mg-state,
                                      proc-list,
                                      ctrl-stk,
                                      temp-stk,
                                      tag ('pc,
                                           cons (subr, length (code (cinfo)))),
                                      t-cond-list))))
      =  p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                  ctrl-stk,
                  push (value (caddr (call-actuals (stmt)),
                               bindings (top (ctrl-stk))),
                push (value (cadr (call-actuals (stmt)),
                                   bindings (top (ctrl-stk))),
                       push (value (car (call-actuals (stmt)),
                                         bindings (top (ctrl-stk))),
                             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))
```

THEOREM: mg-integer-subtract-step-4

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                      ctrl-stk,
                      push (value (caddr (call-actuals (stmt)),
                                    bindings (top (ctrl-stk))),
                            push (value (cadr (call-actuals (stmt)),
                                          bindings (top (ctrl-stk))),
                    push (value (car (call-actuals (stmt)),
                                bindings (top (ctrl-stk))),
                          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))
=   p-state (tag ('pc, '(mg-integer-subtract . 0)),
         push (p-frame (cons (cons ('ans,
                                    value (car (call-actuals (stmt)),
                                           bindings (top (ctrl-stk)))),
                              cons (cons ('y,
                                          value (cadr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk)))),
                                    cons (cons ('z,
                                                value (caddr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          '((t1 int 0))))),
                        tag ('pc,
                             cons (subr, length (code (cinfo))
                                         +   4))),
               ctrl-stk),
         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))
```

THEOREM: mg-integer-subtract-steps-5-9

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
```

```
∧   (car (stmt) = 'predefined-proc-call-mg)
∧   (call-name (stmt) = 'mg-integer-subtract)
∧   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))
∧   user-defined-procp (subr, proc-list)
∧   listp (ctrl-stk)
∧   all-cars-unique (mg-alist (mg-state))
∧   signatures-match (mg-alist (mg-state), name-alist)
∧   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))
∧   normal (mg-state))
→   (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                           '(mg-integer-subtract
                                                             . 0)),
                                                      push (p-frame (cons (cons ('ans,
                                                                                 value (car (call-actuals (stmt)),
                                                                                        bindings (top (ctrl-stk)))),
                                                                          cons (cons ('y,
                                                                                      value (cadr (call-actuals (st
                                                                                                   bindings (top (ctrl-st
                                                                                cons (cons ('z,
                                                                                            value (caddr (call-act
                                                                                                   bindings (top (
                                                                                      '((t1
                                                                                         int
                                                                                         0))))),
                                                                       tag ('pc,
                                                                            cons (subr,
                                                                                  length (code (cinfo))
                                                                                  +   4))),
                                                           ctrl-stk),
                                                      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))))))
    =  p-state (tag ('pc, '(mg-integer-subtract . 5)),
            push (p-frame (cons (cons ('ans,
                                       value (car (call-actuals (stmt)),
                                              bindings (top (ctrl-stk)))),
                                 cons (cons ('y,
                                             value (cadr (call-actuals (stmt)),
                                                    bindings (top (ctrl-stk)))),
                                       cons (cons ('z,
                                                   value (caddr (call-actuals (stmt)),
                                                          bindings (top (ctrl-stk)))),
                                             '((t1 int 0))))),
                           tag ('pc,
                                cons (subr, length (code (cinfo))
                                            +   4))),
                  ctrl-stk),
            push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                        mg-alist (mg-state)))),
                  push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                              mg-alist (mg-state)))),
                        push ('(bool f),
                              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))
```

THEOREM: mg-integer-subtract-step-10-nonerror

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
∧  user-defined-procp (subr, proc-list)
∧  listp (ctrl-stk)
∧  all-cars-unique (mg-alist (mg-state))
∧  signatures-match (mg-alist (mg-state), name-alist)
∧  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))
∧  normal (mg-state)
∧  small-integerp (idifference (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                            mg-alist (mg-state))))),
                               iplus (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                                   mg-alist (mg-state))))),
                                      0)),
                  MG-WORD-SIZE))
→  (p-step (p-state (tag ('pc, '(mg-integer-subtract . 5)),
                    push (p-frame (cons (cons ('ans,
                                              value (car (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                        cons (cons ('y,
                                                    value (cadr (call-actuals (stmt)),
                                                           bindings (top (ctrl-stk)))),
                                              cons (cons ('z,
                                                          value (caddr (call-actuals (stmt)),
                                                                 bindings (top (ctrl-stk)))),
                                                    '((t1 int 0))))),
                                  tag ('pc,
                                       cons (subr, length (code (cinfo))
                                                   +   4))),
                         ctrl-stk),
                    push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                 mg-alist (mg-state)))),
                         push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                      mg-alist (mg-state)))),
                              push ('(bool f),
                                   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))
 =   p-state (tag ('pc, '(mg-integer-subtract . 6)),
                 push (p-frame (cons (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons (cons ('y,
                                                  value (cadr (call-actuals (stmt)),
                                                         bindings (top (ctrl-stk)))),
                                            cons (cons ('z,
                                                        value (caddr (call-actuals (stmt)),
                                                               bindings (top (ctrl-stk)))),
                                                  '((t1 int 0))))),
                                tag ('pc,
                                     cons (subr, length (code (cinfo))
                                                 +   4))),
                       ctrl-stk),
                 push (tag ('int,
                            fix-small-integer (idifference (untag (mg-to-p-simple-literal (caddr (assoc (cadr (cal
                                                                                                          mg-alist
                                                            iplus (untag (mg-to-p-simple-literal (caddr (assoc (ca
                                                                                                                mg
                                                                   0)),
                                               MG-WORD-SIZE)),
                       push (tag ('bool, 'f),
                             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))
```

THEOREM: mg-integer-subtract-steps-11-17-nonerror

```
((n ≄ 0)
 ∧   (¬ resources-inadequatep (stmt,
                               proc-list,
                        list (length (temp-stk),
                              p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p-step (p-step (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                                      '(mg-integer-subtract
                                                                        . 6)),
                                                                  push (p-frame (cons (cons ('ans,
                                                                                             value (car (call-ac
                                                                                                        bindings (t
                                                                                       cons (cons ('y,
                                                                                                   value (cadr (
                                                                                                              bindi
                                                                                             cons (cons ('z,
                                                                                                         value (

                                                                                                   '((t1
                                                                                                      int
                                                                                                      0))))),
                                                                              tag ('pc,
                                                                                   cons (subr,
                                                                                         length (code (cinfo
                                                                                         +   4))),
                                                                        ctrl-stk),
                                                                  push (diff,
                                                                        push (tag ('bool,
                                                                                   'f),
                                                                              map-down-values (mg-alist (mg-
                                                                                                         bindings (top
                                                                         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 (subr, length (code (cinfo)) + 4)),
                 ctrl-stk,
                 rput (diff,
                       untag (value (car (call-actuals (stmt)),
                                     bindings (top (ctrl-stk))))),
                       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))
```

THEOREM: mg-integer-subtract-push-cc

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-integer-subtract)
  ∧   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))
  ∧   user-defined-procp (subr, proc-list)
  ∧   listp (ctrl-stk)
  ∧   all-cars-unique (mg-alist (mg-state))
  ∧   signatures-match (mg-alist (mg-state), name-alist)
  ∧   normal (mg-state))
```

```
→  (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 4)),
                     ctrl-stk,
                     temp-stk,
                     translate-proc-list (proc-list),
                     list (list ('c-c, cc-value)),
                     MG-MAX-CTRL-STK-SIZE,
                     MG-MAX-TEMP-STK-SIZE,
                     MG-WORD-SIZE,
                     'run))
      =  p-state (tag ('pc, cons (subr, length (code (cinfo)) + 5)),
                  ctrl-stk,
                  push (cc-value, temp-stk),
                  translate-proc-list (proc-list),
                  list (list ('c-c, cc-value)),
                  MG-MAX-CTRL-STK-SIZE,
                  MG-MAX-TEMP-STK-SIZE,
                  MG-WORD-SIZE,
                  'run))
```

THEOREM: mg-integer-subtract-sub1-cc

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  normal (mg-state)
 ∧  (cc-value ∈ list ('(nat 1), '(nat 2))))
→  (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 5)),
                     ctrl-stk,
                     push (cc-value, temp-stk),
                     translate-proc-list (proc-list),
                     list (list ('c-c, cc-value)),
                          MG-MAX-CTRL-STK-SIZE,
                          MG-MAX-TEMP-STK-SIZE,
                          MG-WORD-SIZE,
                          'run))
         =   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 6)),
                     ctrl-stk,
                     push (tag ('nat, untag (cc-value) − 1), temp-stk),
                     translate-proc-list (proc-list),
                     list (list ('c-c, cc-value)),
                     MG-MAX-CTRL-STK-SIZE,
                     MG-MAX-TEMP-STK-SIZE,
                     MG-WORD-SIZE,
                     'run))
```

THEOREM: mg-integer-subtract-step-20-nonerror

```
 ((n ≄ 0)
  ∧  (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
  ∧  (car (stmt) = 'predefined-proc-call-mg)
  ∧  (call-name (stmt) = 'mg-integer-subtract)
  ∧  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))
  ∧  user-defined-procp (subr, proc-list)
  ∧  listp (ctrl-stk)
  ∧  all-cars-unique (mg-alist (mg-state))
  ∧  signatures-match (mg-alist (mg-state), name-alist)
  ∧  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))
  ∧  normal (mg-state)
  ∧  small-integerp (idifference (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                             mg-alist (mg-state))))),
                                  iplus (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                                    mg-alist (mg-state))))),
                                         0)),
                     MG-WORD-SIZE))
 →   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 6)),
                    ctrl-stk,
                    push (tag ('nat,
                               untag (mg-cond-to-p-nat (cc (mg-state),
                                                            t-cond-list)) − 1),
                          rput (tag ('int,
                                     fix-small-integer (idifference (untag (mg-to-p-simple-literal (caddr (assoc (

                                                                      iplus (untag (mg-to-p-simple-literal (caddr (

                                                                                   0)),
                                                             MG-WORD-SIZE)),
                                untag (value (car (call-actuals (stmt)),
                                              bindings (top (ctrl-stk)))),
                                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))
=   p-state (tag ('pc,
                  cons (subr,
                        if normal (mg-meaning-r (stmt,
                                                 proc-list,
                                                 mg-state,
                                                 n,
                                                 list (length (temp-stk),
                                                       p-ctrl-stk-size (ctrl-stk))))
                        then length (code (translate (cinfo,
                                                      t-cond-list,
                                                      stmt,
                                                      proc-list)))
                        else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                        proc-list,
                                                                        mg-state,
                                                                        n,
                                                                        list (length (temp-stk),
                                                                              p-ctrl-stk-size (ctrl-stk)))),
                                                     label-alist (translate (cinfo,
                                                                             t-cond-list,
                                                                             stmt,
                                                                    proc-list))),
                                             append (code (translate (cinfo,
                                                                      t-cond-list,
                                                                      stmt,
                                                                      proc-list)),
                                                  code2)) endif)),
                        ctrl-stk,
                        map-down-values (mg-alist (mg-meaning-r (stmt,
                                                                  proc-list,
                                                                  mg-state,
                                                                  n,
                                                                  list (length (temp-stk),
                                                                     p-ctrl-stk-size (ctrl-stk)))),
                                         bindings (top (ctrl-stk)),
                                         temp-stk),
                        translate-proc-list (proc-list),
                        list (list ('c-c,
                                   mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                                       proc-list,
                                                                       mg-state,
                                                                       n,
                                                                       list (length (temp-stk),
                                                                          p-ctrl-stk-size (ctrl-stk)))),
                                                     t-cond-list))),
                        MG-MAX-CTRL-STK-SIZE,
                        MG-MAX-TEMP-STK-SIZE,
                        MG-WORD-SIZE,
                        'run))
```

THEOREM: mg-integer-subtract-step-10-error

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                 p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
∧   listp (ctrl-stk)
∧   all-cars-unique (mg-alist (mg-state))
∧   signatures-match (mg-alist (mg-state), name-alist)
∧   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))
∧   normal (mg-state)
∧   (¬ small-integerp (idifference (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                              mg-alist (mg-state))))),
                                   iplus (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)
                                                                                      mg-alist (mg-state))))),
                                          0)),
                      MG-WORD-SIZE)))
→   (p-step (p-state (tag ('pc, '(mg-integer-subtract . 5)),
                      push (p-frame (cons (cons ('ans,
                                                 value (car (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk)))),
                                           cons (cons ('y,
                                                       value (cadr (call-actuals (stmt)),
                                                              bindings (top (ctrl-stk)))),
                                                 cons (cons ('z,
                                                             value (caddr (call-actuals (stmt)),
                                                                    bindings (top (ctrl-stk)))),
                                                       '((t1 int 0))))),
                                     tag ('pc,
                                          cons (subr, length (code (cinfo))
                                                      +   4))),
                            ctrl-stk),
                      push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                  mg-alist (mg-state)))),
                            push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                        mg-alist (mg-state)))),
                                  push ('(bool f),
                                        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))
```

```
=   p-state (tag ('pc, '(mg-integer-subtract . 6)),
            push (p-frame (cons (cons ('ans,
                                       value (car (call-actuals (stmt)),
                                              bindings (top (ctrl-stk)))),
                                 cons (cons ('y,
                                             value (cadr (call-actuals (stmt)),
                                                    bindings (top (ctrl-stk)))),
                                       cons (cons ('z,
                                                   value (caddr (call-actuals (stmt)),
                                                          bindings (top (ctrl-stk)))),
                                             '((t1 int 0))))),
                           tag ('pc,
                                cons (subr, length (code (cinfo))
                                            +   4))),
                  ctrl-stk),
            push (tag ('int,
                       fix-small-integer (idifference (untag (mg-to-p-simple-literal (caddr (assoc (cadr (cal
                                                                                                       mg-alist
                                                        iplus (untag (mg-to-p-simple-literal (caddr (assoc (ca
                                                                                                             mg
                                                               0)),
                                          MG-WORD-SIZE)),
                  push (tag ('bool, 't),
                        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))
```

THEOREM: mg-integer-subtract-steps-11-15-error

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                             proc-list,
                             list (length (temp-stk),
                                   p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
∧   user-defined-procp (subr, proc-list)
∧   listp (ctrl-stk)
∧   all-cars-unique (mg-alist (mg-state))
∧   signatures-match (mg-alist (mg-state), name-alist)
∧   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))
∧   normal (mg-state))
→   (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                           '(mg-integer-subtract
                                                             . 6)),
                                                      push (p-frame (cons (cons ('ans,
                                                                                 value (car (call-actuals (stmt)),
                                                                                        bindings (top (ctrl-stk)))),
                                                                           cons (cons ('y,
                                                                                       value (cadr (call-actuals (st
                                                                                                    bindings (top (ctrl-s
                                                                                 cons (cons ('z,
                                                                                             value (caddr (call-act
                                                                                                    bindings (top (
                                                                                       '((t1
                                                                                          int
                                                                                          0))))),
                                                                     tag ('pc,
                                                                          cons (subr,
                                                                                length (code (cinfo))
                                                                                +   4))),
                                                           ctrl-stk),
                                                      push (diff,
                                                            push (tag ('bool, 't),
                                                                  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'))))))
    =   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 4)),
                 ctrl-stk,
                 map-down-values (mg-alist (mg-state),
                                  bindings (top (ctrl-stk)),
                                  temp-stk),
                 translate-proc-list (proc-list),
                 list (list ('c-c, '(nat 1))),
                 MG-MAX-CTRL-STK-SIZE,
                 MG-MAX-TEMP-STK-SIZE,
                 MG-WORD-SIZE,
                 'run))
```

THEOREM: mg-integer-subtract-step-18-error

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-integer-subtract)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state)
 ∧  (¬ small-integerp (idifference (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                      mg-alist (mg-state))))),
                                     iplus (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)
                                                                                                mg-alist (mg-state))))),
                                            0)),
                       MG-WORD-SIZE)))
 →  (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 6)),
                  ctrl-stk,
                  push (tag ('nat, untag ('(nat 1)) − 1),
                        map-down-values (mg-alist (mg-state),
                                         bindings (top (ctrl-stk)),
                                         temp-stk)),
                  translate-proc-list (proc-list),
                  '((c-c (nat 1))),
                  MG-MAX-CTRL-STK-SIZE,
                  MG-MAX-TEMP-STK-SIZE,
                  MG-WORD-SIZE,
                  'run))
=  p-state (tag ('pc,
                 cons (subr,
                       if normal (mg-meaning-r (stmt,
                                                proc-list,
                                                mg-state,
                                                n,
                                                list (length (temp-stk),
                                                      p-ctrl-stk-size (ctrl-stk))))
                       then length (code (translate (cinfo,
                                                     t-cond-list,
                                                     stmt,
                                                     proc-list)))
                       else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                       proc-list,
                                                                       mg-state,
                                                                       n,
                                                                       list (length (temp-stk),
                                                                             p-ctrl-stk-size (ctrl-stk)))),
                                                    label-alist (translate (cinfo,
                                                                            t-cond-list,
                                                                            stmt,
                                                                            proc-list))),
                                        append (code (translate (cinfo,
                                                                 t-cond-list,
                                                                 stmt,
                                                                 proc-list)),
                                                code2)) endif)),
               ctrl-stk,
               map-down-values (mg-alist (mg-meaning-r (stmt,
                                                        proc-list,
                                                        mg-state,
                                                        n,
                                                        list (length (temp-stk),
                                                            p-ctrl-stk-size (ctrl-stk)))),
                                         bindings (top (ctrl-stk)),
                                         temp-stk),
                             translate-proc-list (proc-list),
                             list (list ('c-c,
                                        mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                                            proc-list,
                                                                            mg-state,
                                                                            n,
                                                                            list (length (temp-stk),
                                                                                  p-ctrl-stk-size (ctrl-stk)))),
                                                         t-cond-list))),
                             MG-MAX-CTRL-STK-SIZE,
                             MG-MAX-TEMP-STK-SIZE,
                             MG-WORD-SIZE,
                             'run))
```

THEOREM: mg-integer-subtract-exact-time-lemma

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-integer-subtract)
  ∧   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))
  ∧   user-defined-procp (subr, proc-list)
  ∧   listp (ctrl-stk)
  ∧   all-cars-unique (mg-alist (mg-state))
  ∧   signatures-match (mg-alist (mg-state), name-alist)
  ∧   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))
  ∧   normal (mg-state))
  →   (p (map-down (mg-state,
                    proc-list,
                    ctrl-stk,
                    temp-stk,
                tag ('pc, cons (subr, length (code (cinfo)))),
                t-cond-list),
         clock (stmt, proc-list, mg-state, n))
=   p-state (tag ('pc,
                      cons (subr,
                            if normal (mg-meaning-r (stmt,
                                                     proc-list,
                                                     mg-state,
                                                     n,
                                                     list (length (temp-stk),
                                                           p-ctrl-stk-size (ctrl-stk))))
                            then length (code (translate (cinfo,
                                                          t-cond-list,
                                                          stmt,
                                                          proc-list)))
                            else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                            proc-list,
                                                                            mg-state,
                                                                            n,
                                                                            list (length (temp-stk),
                                                                                  p-ctrl-stk-size (ctrl-stk)))),
                                                          label-alist (translate (cinfo,
                                                                                  t-cond-list,
                                                                                  stmt,
                                                                                  proc-list))),
                                             append (code (translate (cinfo,
                                                                      t-cond-list,
                                                                      stmt,
                                                                      proc-list)),
                                                     code2)) endif)),
             ctrl-stk,
             map-down-values (mg-alist (mg-meaning-r (stmt,
                                                      proc-list,
                                                      mg-state,
                                                      n,
                                                      list (length (temp-stk),
                                                            p-ctrl-stk-size (ctrl-stk)))),
                              bindings (top (ctrl-stk)),
                              temp-stk),
             translate-proc-list (proc-list),
             list (list ('c-c,
                         mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                             proc-list,
                                                             mg-state,
                                                     n,
                                                     list (length (temp-stk),
                                                           p-ctrl-stk-size (ctrl-stk)))),
                                         t-cond-list))),
                       MG-MAX-CTRL-STK-SIZE,
                       MG-MAX-TEMP-STK-SIZE,
                       MG-WORD-SIZE,
                       'run))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                      ;;
;;                    EXACT-TIME LEMMA MG-BOOLEAN-OR                    ;;
;;                                                                      ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

EVENT: Disable or-bool.

EVENT: Disable p-objectp-type.

THEOREM: mg-boolean-or-args-have-simple-mg-type-refps

```
((car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 ∧  ok-mg-statep (mg-state, r-cond-list)
 ∧  signatures-match (mg-alist (mg-state), name-alist))
→   (boolean-identifierp (car (call-actuals (stmt)), mg-alist (mg-state))
      ∧  boolean-identifierp (cadr (call-actuals (stmt)),
                              mg-alist (mg-state))
      ∧  boolean-identifierp (caddr (call-actuals (stmt)),
                              mg-alist (mg-state)))
```

THEOREM: mg-boolean-or-args-definedp

```
((car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
 ∧  ok-mg-statep (mg-state, r-cond-list)
 ∧  signatures-match (mg-alist (mg-state), name-alist))
→   (definedp (car (call-actuals (stmt)), mg-alist (mg-state))
      ∧  definedp (cadr (call-actuals (stmt)), mg-alist (mg-state))
      ∧  definedp (caddr (call-actuals (stmt)), mg-alist (mg-state)))
```

THEOREM: mg-boolean-or-steps-1-3

```
((n ≄ 0)
∧   (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
∧   (car (stmt) = 'predefined-proc-call-mg)
∧   (call-name (stmt) = 'mg-boolean-or)
∧   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))
∧   user-defined-procp (subr, proc-list)
∧   listp (ctrl-stk)
∧   all-cars-unique (mg-alist (mg-state))
∧   signatures-match (mg-alist (mg-state), name-alist)
∧   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))
∧   normal (mg-state))
→   (p-step (p-step (p-step (map-down (mg-state,
                                       proc-list,
                                       ctrl-stk,
                                       temp-stk,
                                       tag ('pc,
                                            cons (subr, length (code (cinfo)))),
                                       t-cond-list))))
      =   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                   ctrl-stk,
                   push (value (caddr (call-actuals (stmt)),
                                bindings (top (ctrl-stk))),
                         push (value (cadr (call-actuals (stmt)),
                                      bindings (top (ctrl-stk))),
                               push (value (car (call-actuals (stmt)),
                                            bindings (top (ctrl-stk))),
                                     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))
```

THEOREM: mg-boolean-or-step-4

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                     ctrl-stk,
                     push (value (caddr (call-actuals (stmt)),
                                  bindings (top (ctrl-stk))),
                           push (value (cadr (call-actuals (stmt)),
                                        bindings (top (ctrl-stk))),
                                 push (value (car (call-actuals (stmt)),
                                              bindings (top (ctrl-stk))),
                                       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))
=   p-state (tag ('pc, '(mg-boolean-or . 0)),
            push (p-frame (list (cons ('ans,
                                       value (car (call-actuals (stmt)),
                                              bindings (top (ctrl-stk)))),
                                 cons ('b1,
                                       value (cadr (call-actuals (stmt)),
                                              bindings (top (ctrl-stk)))),
                                 cons ('b2,
                                       value (caddr (call-actuals (stmt)),
                                              bindings (top (ctrl-stk))))),
                              tag ('pc,
                                   cons (subr, length (code (cinfo))
                                               +   4))),
                  ctrl-stk),
            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))
```

THEOREM: mg-boolean-or-steps-5-8

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
∧   normal (mg-state))
→   (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                              '(mg-boolean-or . 0)),
                                         push (p-frame (list (cons ('ans,
                                                                    value (car (call-actuals (stmt)),
                                                                               bindings (top (ctrl-stk)))),
                                                              cons ('b1,
                                                                    value (cadr (call-actuals (stmt)),
                                                                               bindings (top (ctrl-stk)))),
                                                              cons ('b2,
                                                                    value (caddr (call-actuals (stmt)),
                                                                               bindings (top (ctrl-stk))))),
                                                        tag ('pc,
                                                             cons (subr,
                                                                   length (code (cinfo))
                                                                   +   4))),
                                               ctrl-stk),
                                         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)))))
    =   p-state (tag ('pc, '(mg-boolean-or . 4)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                      cons ('b1,
                                            value (cadr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                      cons ('b2,
                                            value (caddr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk))))),
                                tag ('pc,
                                     cons (subr, length (code (cinfo))
                                                 +   4))),
                    ctrl-stk),
          push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                      mg-alist (mg-state)))),
                push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                            mg-alist (mg-state)))),
                      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))
```

THEOREM: mg-boolean-or-steps-9-11

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p-step (p-step (p-step (p-state (tag ('pc, '(mg-boolean-or . 4)),
                                      push (p-frame (list (cons ('ans,
                                                                 value (car (call-actuals (stmt)),
                                                                        bindings (top (ctrl-stk)))),
                                                           cons ('b1,
                                                    value (cadr (call-actuals (stmt)),
                                                           bindings (top (ctrl-stk)))),
                                                cons ('b2,
                                                    value (caddr (call-actuals (stmt)),
                                                           bindings (top (ctrl-stk))))),
                                            tag ('pc,
                                                cons (subr,
                                                    length (code (cinfo))
                                                    +    4))),
                                   ctrl-stk),
                              push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                         mg-alist (mg-state)))),
                                  push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt))
                                                                              mg-alist (mg-state)))),
                                      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))))
=   p-state (tag ('pc, '(mg-boolean-or . 7)),
             push (p-frame (list (cons ('ans,
                                        value (car (call-actuals (stmt)),
                                               bindings (top (ctrl-stk)))),
                                   cons ('b1,
                                        value (cadr (call-actuals (stmt)),
                                               bindings (top (ctrl-stk)))),
                                   cons ('b2,
                                        value (caddr (call-actuals (stmt)),
                                               bindings (top (ctrl-stk))))),
                              tag ('pc,
                                   cons (subr, length (code (cinfo))
                                               +    4))),
                   ctrl-stk),
             rput (tag ('bool,
                        or-bool (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                              mg-alist (mg-state))))),
                                 untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                              mg-alist (mg-state))))))),
                    untag (value (car (call-actuals (stmt)),
                                  bindings (top (ctrl-stk)))),
                    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))
```

THEOREM: boolean-literal-simple-typed-literalp

$\text{boolean-literalp}\,(x) \rightarrow \text{simple-typed-literalp}\,(x,\ \texttt{'boolean-mg})$

THEOREM: mg-boolean-or-step-12

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p-step (p-state (tag ('pc, '(mg-boolean-or . 7)),
                      push (p-frame (list (cons ('ans,
                                                 value (car (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk)))),
                                           cons ('b1,
                                    value (cadr (call-actuals (stmt)),
                                           bindings (top (ctrl-stk)))),
                              cons ('b2,
                                    value (caddr (call-actuals (stmt)),
                                           bindings (top (ctrl-stk))))),
                         tag ('pc,
                              cons (subr, length (code (cinfo))
                                          +    4))),
                  ctrl-stk),
           rput (tag ('bool,
                      or-bool (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                           mg-alist (mg-state))))),
                               untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                            mg-alist (mg-state))))))),
                 untag (value (car (call-actuals (stmt)),
                               bindings (top (ctrl-stk)))),
                 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))
=    p-state (tag ('pc,
                   cons (subr,
                         if normal (mg-meaning-r (stmt,
                                                  proc-list,
                                                  mg-state,
                                                  n,
                                                  list (length (temp-stk),
                                                        p-ctrl-stk-size (ctrl-stk))))
                         then length (code (translate (cinfo,
                                                       t-cond-list,
                                                       stmt,
                                                       proc-list)))
                         else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                         proc-list,
                                                                         mg-state,
                                                                         n,
                                                                         list (length (temp-stk),
                                                                               p-ctrl-stk-size (ctrl-stk)))),
                                                        label-alist (translate (cinfo,
                                                                                 t-cond-list,
                                                                                 stmt,
                                                                                 proc-list))),
                                                 append (code (translate (cinfo,
                                                                            t-cond-list,
                                                                            stmt,
                                                                            proc-list)),
                                                        code2)) endif)),
                    ctrl-stk,
                    map-down-values (mg-alist (mg-meaning-r (stmt,
                                                             proc-list,
                                                             mg-state,
                                                             n,
                                                             list (length (temp-stk),
                                                                   p-ctrl-stk-size (ctrl-stk)))),
                                     bindings (top (ctrl-stk)),
                                     temp-stk),
                    translate-proc-list (proc-list),
                    list (list ('c-c,
                          mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                              proc-list,
                                                              mg-state,
                                                              n,
                                                              list (length (temp-stk),
                                                                    p-ctrl-stk-size (ctrl-stk)))),
                                            t-cond-list))),
                    MG-MAX-CTRL-STK-SIZE,
                    MG-MAX-TEMP-STK-SIZE,
                    MG-WORD-SIZE,
                    'run))
```

THEOREM: mg-boolean-or-exact-time-lemma

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-or)
 ∧  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))
    ∧   user-defined-procp (subr, proc-list)
    ∧   listp (ctrl-stk)
    ∧   all-cars-unique (mg-alist (mg-state))
    ∧   signatures-match (mg-alist (mg-state), name-alist)
    ∧   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))
    ∧   normal (mg-state))
    →   (p (map-down (mg-state,
                      proc-list,
                      ctrl-stk,
                      temp-stk,
                      tag ('pc, cons (subr, length (code (cinfo)))),
                      t-cond-list),
           clock (stmt, proc-list, mg-state, n))
         =   p-state (tag ('pc,
                           cons (subr,
                                 if normal (mg-meaning-r (stmt,
                                                          proc-list,
                                                          mg-state,
                                                          n,
                                                          list (length (temp-stk),
                                                                p-ctrl-stk-size (ctrl-stk))))
                                 then length (code (translate (cinfo,
                                                               t-cond-list,
                                                               stmt,
                                                               proc-list)))
                                 else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                                 proc-list,
                                                                                 mg-state,
                                                                                 n,
                                                                                 list (length (temp-stk),
                                                                                       p-ctrl-stk-size (ctrl-stk)))),
                                                               label-alist (translate (cinfo,
                                                                                       t-cond-list,
                                                                                       stmt,
                                                                                       proc-list))),
                                                  append (code (translate (cinfo,
                                                                           t-cond-list,
                                                                           stmt,
                                                                           proc-list)),
                                    code2)) endif)),
              ctrl-stk,
              map-down-values (mg-alist (mg-meaning-r (stmt,
                                                       proc-list,
                                                       mg-state,
                                                       n,
                                                       list (length (temp-stk),
                                                             p-ctrl-stk-size (ctrl-stk)))),
                               bindings (top (ctrl-stk)),
                               temp-stk),
              translate-proc-list (proc-list),
              list (list ('c-c,
                     mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                         proc-list,
                                                         mg-state,
                                                         n,
                                                         list (length (temp-stk),
                                                               p-ctrl-stk-size (ctrl-stk)))),
                                       t-cond-list))),
              MG-MAX-CTRL-STK-SIZE,
              MG-MAX-TEMP-STK-SIZE,
              MG-WORD-SIZE,
              'run))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                        ;;
;;                   EXACT-TIME LEMMA MG-BOOLEAN-AND                      ;;
;;                                                                        ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

EVENT: Disable and-bool.

EVENT: Disable p-objectp-type.

THEOREM: mg-boolean-and-args-have-simple-mg-type-refps

```
 ((car (stmt) = 'predefined-proc-call-mg)
  ∧  (call-name (stmt) = 'mg-boolean-and)
  ∧  ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
  ∧  ok-mg-statep (mg-state, r-cond-list)
  ∧  signatures-match (mg-alist (mg-state), name-alist))
 →  (boolean-identifierp (car (call-actuals (stmt)), mg-alist (mg-state))
      ∧  boolean-identifierp (cadr (call-actuals (stmt)),
                               mg-alist (mg-state))
        ∧    boolean-identifierp (caddr (call-actuals (stmt)),
                               mg-alist (mg-state)))
```

THEOREM: mg-boolean-and-args-definedp

```
 ((car (stmt) = 'predefined-proc-call-mg)
  ∧    (call-name (stmt) = 'mg-boolean-and)
  ∧    ok-mg-statement (stmt, r-cond-list, name-alist, proc-list)
  ∧    ok-mg-statep (mg-state, r-cond-list)
  ∧    signatures-match (mg-alist (mg-state), name-alist))
 →     (definedp (car (call-actuals (stmt)), mg-alist (mg-state))
         ∧    definedp (cadr (call-actuals (stmt)), mg-alist (mg-state))
         ∧    definedp (caddr (call-actuals (stmt)), mg-alist (mg-state)))
```

THEOREM: mg-boolean-and-steps-1-3

```
 ((n ≄ 0)
  ∧    (¬ resources-inadequatep (stmt,
                                 proc-list,
                                 list (length (temp-stk),
                                       p-ctrl-stk-size (ctrl-stk))))
  ∧    (car (stmt) = 'predefined-proc-call-mg)
  ∧    (call-name (stmt) = 'mg-boolean-and)
  ∧    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))
  ∧    user-defined-procp (subr, proc-list)
  ∧    listp (ctrl-stk)
  ∧    all-cars-unique (mg-alist (mg-state))
  ∧    signatures-match (mg-alist (mg-state), name-alist)
  ∧    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))
  ∧    normal (mg-state))
 →     (p-step (p-step (p-step (map-down (mg-state,
                                          proc-list,
                                          ctrl-stk,
                                          temp-stk,
                                          tag ('pc,
                                               cons (subr, length (code (cinfo)))),
                                          t-cond-list))))
```

```
=   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
            ctrl-stk,
            push (value (caddr (call-actuals (stmt)),
                         bindings (top (ctrl-stk))),
                  push (value (cadr (call-actuals (stmt)),
                               bindings (top (ctrl-stk))),
                        push (value (car (call-actuals (stmt)),
                                     bindings (top (ctrl-stk))),
                              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))
```

THEOREM: mg-boolean-and-step-4

```
 ((n ≄ 0)
  ∧  (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
  ∧  (car (stmt) = 'predefined-proc-call-mg)
  ∧  (call-name (stmt) = 'mg-boolean-and)
  ∧  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))
  ∧  user-defined-procp (subr, proc-list)
  ∧  listp (ctrl-stk)
  ∧  all-cars-unique (mg-alist (mg-state))
  ∧  signatures-match (mg-alist (mg-state), name-alist)
  ∧  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))
  ∧  normal (mg-state))
 →   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 3)),
                       ctrl-stk,
                   push (value (caddr (call-actuals (stmt)),
                                bindings (top (ctrl-stk))),
                         push (value (cadr (call-actuals (stmt)),
                                      bindings (top (ctrl-stk))),
                               push (value (car (call-actuals (stmt)),
                                            bindings (top (ctrl-stk))),
                                     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))
 =   p-state (tag ('pc, '(mg-boolean-and . 0)),
               push (p-frame (list (cons ('ans,
                                          value (car (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk)))),
                                    cons ('b1,
                                          value (cadr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk)))),
                                    cons ('b2,
                                          value (caddr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk))))),
                               tag ('pc,
                                    cons (subr, length (code (cinfo))
                                                + 4))),
                     ctrl-stk),
               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))
```

THEOREM: mg-boolean-and-steps-5-8

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧   (car (stmt) = 'predefined-proc-call-mg)
 ∧   (call-name (stmt) = 'mg-boolean-and)
 ∧   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))
 ∧   user-defined-procp (subr, proc-list)
 ∧   listp (ctrl-stk)
 ∧   all-cars-unique (mg-alist (mg-state))
 ∧   signatures-match (mg-alist (mg-state), name-alist)
 ∧   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))
 ∧   normal (mg-state))
→    (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                  '(mg-boolean-and . 0)),
                                              push (p-frame (list (cons ('ans,
                                                                         value (car (call-actuals (stmt)),
                                                                                bindings (top (ctrl-stk)))),
                                                                   cons ('b1,
                                                                         value (cadr (call-actuals (stmt)),
                                                                                bindings (top (ctrl-stk)))),
                                                                   cons ('b2,
                                                                         value (caddr (call-actuals (stmt)),
                                                                                bindings (top (ctrl-stk))))),
                                                             tag ('pc,
                                                                  cons (subr,
                                                                        length (code (cinfo))
                                                                        +  4))),
                                                    ctrl-stk),
                                              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)))))
     =   p-state (tag ('pc, '(mg-boolean-and . 4)),
              push (p-frame (list (cons ('ans,
                                         value (car (call-actuals (stmt)),
                                                bindings (top (ctrl-stk)))),
                                   cons ('b1,
                                         value (cadr (call-actuals (stmt)),
                                                bindings (top (ctrl-stk)))),
                                   cons ('b2,
                                         value (caddr (call-actuals (stmt)),
                                                bindings (top (ctrl-stk))))),
                             tag ('pc,
                                  cons (subr, length (code (cinfo))
                                              +   4))),
                    ctrl-stk),
              push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                          mg-alist (mg-state)))),
                    push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                mg-alist (mg-state)))),
                          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))
```

THEOREM: mg-boolean-and-steps-9-11

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-boolean-and)
  ∧   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))
    ∧   user-defined-procp (subr, proc-list)
    ∧   listp (ctrl-stk)
    ∧   all-cars-unique (mg-alist (mg-state))
    ∧   signatures-match (mg-alist (mg-state), name-alist)
    ∧   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))
    ∧   normal (mg-state))
    →   (p-step (p-step (p-step (p-state (tag ('pc, '(mg-boolean-and . 4)),
                                          push (p-frame (list (cons ('ans,
                                                                     value (car (call-actuals (stmt)),
                                                                            bindings (top (ctrl-stk)))),
                                                               cons ('b1,
                                                                     value (cadr (call-actuals (stmt)),
                                                                            bindings (top (ctrl-stk)))),
                                                               cons ('b2,
                                                                     value (caddr (call-actuals (stmt)),
                                                                            bindings (top (ctrl-stk))))),
                                                          tag ('pc,
                                                               cons (subr,
                                                                     length (code (cinfo))
                                                                     +   4))),
                                                ctrl-stk),
                                          push (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                                      mg-alist (mg-state)))),
                                                push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt))
                                                                                            mg-alist (mg-state)))),
                                                      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))))
        =   p-state (tag ('pc, '(mg-boolean-and . 7)),
                     push (p-frame (list (cons ('ans,
                                                value (car (call-actuals (stmt)),
                                                bindings (top (ctrl-stk)))),
                                         cons ('b1,
                                                value (cadr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                         cons ('b2,
                                                value (caddr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk))))),
                                      tag ('pc,
                                         cons (subr, length (code (cinfo))
                                                      +    4))),
                           ctrl-stk),
                     rput (tag ('bool,
                                 and-bool (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                                   mg-alist (mg-state))))),
                                                  untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                                                   mg-alist (mg-state))))))),
                           untag (value (car (call-actuals (stmt)),
                                          bindings (top (ctrl-stk)))),
                           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))
```

THEOREM: mg-boolean-and-step-12

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-boolean-and)
  ∧   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))
  ∧   user-defined-procp (subr, proc-list)
```

```
∧  listp (ctrl-stk)
∧  all-cars-unique (mg-alist (mg-state))
∧  signatures-match (mg-alist (mg-state), name-alist)
∧  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))
∧  normal (mg-state))
→  (p-step (p-state (tag ('pc, '(mg-boolean-and . 7)),
                     push (p-frame (list (cons ('ans,
                                                value (car (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                         cons ('b1,
                                               value (cadr (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk)))),
                                         cons ('b2,
                                               value (caddr (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk))))),
                                    tag ('pc,
                                         cons (subr, length (code (cinfo))
                                                     +   4))),
                           ctrl-stk),
                     rput (tag ('bool,
                                and-bool (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt))
                                                                                      mg-alist (mg-state))))),
                                          untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt
                                                                                      mg-alist (mg-state))))))
                           untag (value (car (call-actuals (stmt)),
                                         bindings (top (ctrl-stk)))),
                           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))
     =  p-state (tag ('pc,
                      cons (subr,
                            if normal (mg-meaning-r (stmt,
                                                     proc-list,
                                                     mg-state,
                                    n,
                                    list (length (temp-stk),
                                        p-ctrl-stk-size (ctrl-stk))))
          then length (code (translate (cinfo,
                                        t-cond-list,
                                        stmt,
                                        proc-list)))
          else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                         proc-list,
                                                         mg-state,
                                                         n,
                                                         list (length (temp-stk),
                                                             p-ctrl-stk-size (ctrl-stk)))),
                                     label-alist (translate (cinfo,
                                                            t-cond-list,
                                                            stmt,
                                                            proc-list))),
                          append (code (translate (cinfo,
                                                   t-cond-list,
                                                   stmt,
                                                   proc-list)),
                                   code2)) endif)),
ctrl-stk,
map-down-values (mg-alist (mg-meaning-r (stmt,
                                         proc-list,
                                         mg-state,
                                         n,
                                         list (length (temp-stk),
                                             p-ctrl-stk-size (ctrl-stk)))),
                   bindings (top (ctrl-stk)),
                   temp-stk),
translate-proc-list (proc-list),
list (list ('c-c,
         mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                             proc-list,
                                             mg-state,
                                             n,
                                             list (length (temp-stk),
                                                 p-ctrl-stk-size (ctrl-stk)))),
                            t-cond-list))),
MG-MAX-CTRL-STK-SIZE,
MG-MAX-TEMP-STK-SIZE,
MG-WORD-SIZE,
'run))
```

THEOREM: mg-boolean-and-exact-time-lemma

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-and)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
→   (p (map-down (mg-state,
                  proc-list,
                  ctrl-stk,
                  temp-stk,
                  tag ('pc, cons (subr, length (code (cinfo)))),
                  t-cond-list),
       clock (stmt, proc-list, mg-state, n))
     =  p-state (tag ('pc,
                      cons (subr,
                            if normal (mg-meaning-r (stmt,
                                                     proc-list,
                                                     mg-state,
                                                     n,
                                                     list (length (temp-stk),
                                                           p-ctrl-stk-size (ctrl-stk))))
                            then length (code (translate (cinfo,
                                                          t-cond-list,
                                                          stmt,
                                                          proc-list)))
                            else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                           proc-list,
                                                              mg-state,
                                                              n,
                                                              list (length (temp-stk),
                                                                    p-ctrl-stk-size (ctrl-stk)))),
                                                 label-alist (translate (cinfo,
                                                                         t-cond-list,
                                                                         stmt,
                                                                         proc-list))),
                                          append (code (translate (cinfo,
                                                                   t-cond-list,
                                                                   stmt,
                                                                   proc-list)),
                                                  code2)) endif)),
                  ctrl-stk,
                  map-down-values (mg-alist (mg-meaning-r (stmt,
                                                           proc-list,
                                                           mg-state,
                                                           n,
                                                           list (length (temp-stk),
                                                                 p-ctrl-stk-size (ctrl-stk)))),
                                   bindings (top (ctrl-stk)),
                                   temp-stk),
                  translate-proc-list (proc-list),
                  list (list ('c-c,
                              mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                                   proc-list,
                                                                   mg-state,
                                                                   n,
                                                                   list (length (temp-stk),
                                                                         p-ctrl-stk-size (ctrl-stk)))),
                                                t-cond-list))),
                  MG-MAX-CTRL-STK-SIZE,
                  MG-MAX-TEMP-STK-SIZE,
                  MG-WORD-SIZE,
                  'run))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;                                                                                ;;
;;                     EXACT-TIME LEMMA MG-BOOLEAN-NOT                             ;;
;;                                                                                ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

EVENT: Disable not-bool.

THEOREM: mg-boolean-not-args-have-simple-mg-type-refps
((car (*stmt*) = `'predefined-proc-call-mg`)
∧ (call-name (*stmt*) = `'mg-boolean-not`)
∧ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
∧ ok-mg-statep (*mg-state*, *r-cond-list*)
∧ signatures-match (mg-alist (*mg-state*), *name-alist*))
→ (boolean-identifierp (car (call-actuals (*stmt*)), mg-alist (*mg-state*))
∧ boolean-identifierp (cadr (call-actuals (*stmt*)),
mg-alist (*mg-state*)))

THEOREM: mg-boolean-not-args-definedp
((car (*stmt*) = `'predefined-proc-call-mg`)
∧ (call-name (*stmt*) = `'mg-boolean-not`)
∧ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
∧ ok-mg-statep (*mg-state*, *r-cond-list*)
∧ signatures-match (mg-alist (*mg-state*), *name-alist*))
→ (definedp (car (call-actuals (*stmt*)), mg-alist (*mg-state*))
∧ definedp (cadr (call-actuals (*stmt*)), mg-alist (*mg-state*)))

THEOREM: mg-boolean-not-steps-1-2
(($n \not\simeq$ `0`)
∧ (¬ resources-inadequatep (*stmt*,
proc-list,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*))))
∧ (car (*stmt*) = `'predefined-proc-call-mg`)
∧ (call-name (*stmt*) = `'mg-boolean-not`)
∧ 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))
∧ user-defined-procp (*subr*, *proc-list*)
∧ listp (*ctrl-stk*)
∧ all-cars-unique (mg-alist (*mg-state*))
∧ signatures-match (mg-alist (*mg-state*), *name-alist*)
∧ 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*))
∧ normal (*mg-state*))
→ (p-step (p-step (map-down (*mg-state*,
proc-list,

```
                                   ctrl-stk,
                                   temp-stk,
                                   tag ('pc, cons (subr, length (code (cinfo)))),
                                   t-cond-list)))
    =   p-state (tag ('pc, cons (subr, length (code (cinfo)) + 2)),
                 ctrl-stk,
                 push (value (cadr (call-actuals (stmt)),
                              bindings (top (ctrl-stk))),
                       push (value (car (call-actuals (stmt)),
                                    bindings (top (ctrl-stk))),
                             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))
```

THEOREM: mg-boolean-not-step-3

```
((n ≄ 0)
 ∧  (¬ resources-inadequatep (stmt,
                              proc-list,
                              list (length (temp-stk),
                                    p-ctrl-stk-size (ctrl-stk))))
 ∧  (car (stmt) = 'predefined-proc-call-mg)
 ∧  (call-name (stmt) = 'mg-boolean-not)
 ∧  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))
 ∧  user-defined-procp (subr, proc-list)
 ∧  listp (ctrl-stk)
 ∧  all-cars-unique (mg-alist (mg-state))
 ∧  signatures-match (mg-alist (mg-state), name-alist)
 ∧  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))
 ∧  normal (mg-state))
 →   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 2)),
                       ctrl-stk,
                       push (value (cadr (call-actuals (stmt)),
                                    bindings (top (ctrl-stk))),
                             push (value (car (call-actuals (stmt)),
                                          bindings (top (ctrl-stk))),
                                   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))
       =   p-state (tag ('pc, '(mg-boolean-not . 0)),
                    push (p-frame (list (cons ('ans,
                                               value (car (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk)))),
                                         cons ('b1,
                                               value (cadr (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk))))),
                                   tag ('pc,
                                        cons (subr, length (code (cinfo))
                                                    +   3))),
                          ctrl-stk),
                    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))
```

THEOREM: mg-boolean-not-steps-4-5

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
```

```
∧   (car (stmt) = 'predefined-proc-call-mg)
∧   (call-name (stmt) = 'mg-boolean-not)
∧   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))
∧   user-defined-procp (subr, proc-list)
∧   listp (ctrl-stk)
∧   all-cars-unique (mg-alist (mg-state))
∧   signatures-match (mg-alist (mg-state), name-alist)
∧   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))
∧   normal (mg-state))
→   (p-step (p-step (p-state (tag ('pc, '(mg-boolean-not . 0)),
                              push (p-frame (list (cons ('ans,
                                                         value (car (call-actuals (stmt)),
                                                                bindings (top (ctrl-stk)))),
                                                   cons ('b1,
                                                         value (cadr (call-actuals (stmt)),
                                                                bindings (top (ctrl-stk))))),
                                             tag ('pc,
                                                  cons (subr,
                                                        length (code (cinfo))
                                                        +   3))),
                                    ctrl-stk),
                              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)))
    =   p-state (tag ('pc, '(mg-boolean-not . 2)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                              cons ('b1,
                                    value (cadr (call-actuals (stmt)),
                                           bindings (top (ctrl-stk))))),
                         tag ('pc,
                              cons (subr, length (code (cinfo))
                                          +   3))),
               ctrl-stk),
       push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                  mg-alist (mg-state)))),
             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))
```

Theorem: mg-boolean-not-steps-6-8

```
 ((n ≄ 0)
  ∧  (¬ resources-inadequatep (stmt,
                               proc-list,
                               list (length (temp-stk),
                                     p-ctrl-stk-size (ctrl-stk))))
  ∧  (car (stmt) = 'predefined-proc-call-mg)
  ∧  (call-name (stmt) = 'mg-boolean-not)
  ∧  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))
  ∧  user-defined-procp (subr, proc-list)
  ∧  listp (ctrl-stk)
  ∧  all-cars-unique (mg-alist (mg-state))
  ∧  signatures-match (mg-alist (mg-state), name-alist)
  ∧  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))
  ∧  normal (mg-state))
 →   (p-step (p-step (p-step (p-state (tag ('pc, '(mg-boolean-not . 2)),
                                        push (p-frame (list (cons ('ans,
                                                               value (car (call-actuals (stmt)),
                                                                      bindings (top (ctrl-stk)))),
                                                           cons ('b1,
                                                               value (cadr (call-actuals (stmt)),
                                                                      bindings (top (ctrl-stk))))),
                                                       tag ('pc,
                                                           cons (subr,
                                                               length (code (cinfo))
                                                               +   3))),
                                            ctrl-stk),
                                        push (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                    mg-alist (mg-state)))),
                                            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))))
=   p-state (tag ('pc, '(mg-boolean-not . 5)),
           push (p-frame (list (cons ('ans,
                                      value (car (call-actuals (stmt)),
                                             bindings (top (ctrl-stk)))),
                                  cons ('b1,
                                      value (cadr (call-actuals (stmt)),
                                             bindings (top (ctrl-stk))))),
                              tag ('pc,
                                  cons (subr, length (code (cinfo))
                                               +   3))),
                 ctrl-stk),
           rput (tag ('bool,
                      not-bool (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                              mg-alist (mg-state))))))),
                 untag (value (car (call-actuals (stmt)),
                                bindings (top (ctrl-stk)))),
                 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))
```

THEOREM: mg-boolean-not-step-9

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-boolean-not)
  ∧   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))
  ∧   user-defined-procp (subr, proc-list)
  ∧   listp (ctrl-stk)
  ∧   all-cars-unique (mg-alist (mg-state))
  ∧   signatures-match (mg-alist (mg-state), name-alist)
  ∧   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))
  ∧   normal (mg-state))
 →   (p-step (p-state (tag ('pc, '(mg-boolean-not . 5)),
                       push (p-frame (list (cons ('ans,
                                                  value (car (call-actuals (stmt)),
                                                         bindings (top (ctrl-stk)))),
                                            cons ('b1,
                                                  value (cadr (call-actuals (stmt)),
                                                         bindings (top (ctrl-stk))))),
                                      tag ('pc,
                                           cons (subr, length (code (cinfo))
                                                       +   3))),
                             ctrl-stk),
                       rput (tag ('bool,
                                  not-bool (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                          mg-alist (mg-state))))))),
                    untag (value (car (call-actuals (stmt)),
                                  bindings (top (ctrl-stk)))),
                    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))
=   p-state (tag ('pc,
                  cons (subr,
                        if normal (mg-meaning-r (stmt,
                                                 proc-list,
                                                 mg-state,
                                                 n,
                                                 list (length (temp-stk),
                                                       p-ctrl-stk-size (ctrl-stk))))
                        then length (code (translate (cinfo,
                                                      t-cond-list,
                                                      stmt,
                                                      proc-list)))
                        else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                        proc-list,
                                                                        mg-state,
                                                                        n,
                                                                        list (length (temp-stk),
                                                                              p-ctrl-stk-size (ctrl-stk)))),
                                                      label-alist (translate (cinfo,
                                                                              t-cond-list,
                                                                              stmt,
                                                                              proc-list))),
                                         append (code (translate (cinfo,
                                                                  t-cond-list,
                                                                  stmt,
                                                                  proc-list)),
                                                 code2)) endif)),
             ctrl-stk,
             map-down-values (mg-alist (mg-meaning-r (stmt,
                                                      proc-list,
                                                      mg-state,
                                                      n,
                                                        list (length (temp-stk),
                                                              p-ctrl-stk-size (ctrl-stk)))),
                                        bindings (top (ctrl-stk)),
                                        temp-stk),
                       translate-proc-list (proc-list),
                       list (list ('c-c,
                                  mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                                       proc-list,
                                                                       mg-state,
                                                                       n,
                                                                       list (length (temp-stk),
                                                                             p-ctrl-stk-size (ctrl-stk)))),
                                                    t-cond-list))),
                       MG-MAX-CTRL-STK-SIZE,
                       MG-MAX-TEMP-STK-SIZE,
                       MG-WORD-SIZE,
                       'run))
```

THEOREM: mg-boolean-not-exact-time-lemma

```
 ((n ≄ 0)
  ∧   (¬ resources-inadequatep (stmt,
                                proc-list,
                                list (length (temp-stk),
                                      p-ctrl-stk-size (ctrl-stk))))
  ∧   (car (stmt) = 'predefined-proc-call-mg)
  ∧   (call-name (stmt) = 'mg-boolean-not)
  ∧   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))
  ∧   user-defined-procp (subr, proc-list)
  ∧   listp (ctrl-stk)
  ∧   all-cars-unique (mg-alist (mg-state))
  ∧   signatures-match (mg-alist (mg-state), name-alist)
  ∧   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))
  ∧   normal (mg-state))
 →    (p (map-down (mg-state,
                    proc-list,
                    ctrl-stk,
                    temp-stk,
                    tag ('pc, cons (subr, length (code (cinfo)))),
                    t-cond-list),
        clock (stmt, proc-list, mg-state, n))
=    p-state (tag ('pc,
                     cons (subr,
                           if normal (mg-meaning-r (stmt,
                                                    proc-list,
                                                    mg-state,
                                                    n,
                                                    list (length (temp-stk),
                                                          p-ctrl-stk-size (ctrl-stk))))
                           then length (code (translate (cinfo,
                                                         t-cond-list,
                                                         stmt,
                                                         proc-list)))
                           else find-label (fetch-label (cc (mg-meaning-r (stmt,
                                                                           proc-list,
                                                                           mg-state,
                                                                           n,
                                                                           list (length (temp-stk),
                                                                                 p-ctrl-stk-size (ctrl-stk)))),
                                                         label-alist (translate (cinfo,
                                                                                 t-cond-list,
                                                                                 stmt,
                                                                                 proc-list))),
                                            append (code (translate (cinfo,
                                                                     t-cond-list,
                                                                     stmt,
                                                                     proc-list)),
                                                    code2)) endif)),
             ctrl-stk,
             map-down-values (mg-alist (mg-meaning-r (stmt,
                                                      proc-list,
                                                      mg-state,
                                                      n,
                                                      list (length (temp-stk),
                                                            p-ctrl-stk-size (ctrl-stk)))),
                              bindings (top (ctrl-stk)),
                              temp-stk),
             translate-proc-list (proc-list),
             list (list ('c-c,
                         mg-cond-to-p-nat (cc (mg-meaning-r (stmt,
                                                             proc-list,
                                                             mg-state,
                                                             n,
                                                             list (length (temp-stk),
                                                                     p-ctrl-stk-size (ctrl-stk)))),
                                                     t-cond-list))),
                            MG-MAX-CTRL-STK-SIZE,
                            MG-MAX-TEMP-STK-SIZE,
                            MG-WORD-SIZE,
                            'run))
```

EVENT: Make the library `"c-predefined3"`.

Index