EVENT: Start with the library `"c-predefined1"`.

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

THEOREM: mg-integer-le-args-have-simple-mg-type-refps

((car (*stmt*) = **'predefined-proc-call-mg**)
 ∧ (call-name (*stmt*) = **'mg-integer-le**)
 ∧ 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*))
  ∧ int-identifierp (cadr (call-actuals (*stmt*)), mg-alist (*mg-state*))
  ∧ int-identifierp (caddr (call-actuals (*stmt*)), mg-alist (*mg-state*)))

THEOREM: mg-integer-le-args-definedp

((car (*stmt*) = **'predefined-proc-call-mg**)
 ∧ (call-name (*stmt*) = **'mg-integer-le**)
 ∧ 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-le-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-le**)
 ∧ 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*)

$\wedge$ listp (*ctrl-stk*)
$\wedge$ all-cars-unique (mg-alist (*mg-state*))
$\wedge$ signatures-match (mg-alist (*mg-state*), *name-alist*)
$\wedge$ mg-vars-list-ok-in-p-state (mg-alist (*mg-state*),
bindings (top (*ctrl-stk*)),
temp-stk)
$\wedge$ no-p-aliasing (bindings (top (*ctrl-stk*)), mg-alist (*mg-state*))
$\wedge$ normal (*mg-state*))
$\rightarrow$ (p-step (p-step (p-step (map-down (*mg-state*,
proc-list,
ctrl-stk,
temp-stk,
tag (`'pc`,
cons (*subr*, length (code (*cinfo*)))),
t-cond-list))))
$=$ p-state (tag (`'pc`, cons (*subr*, 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-le-step-4
(($n \not\simeq$ `0`)
$\wedge$ ($\neg$ resources-inadequatep (*stmt*,
proc-list,
list (length (*temp-stk*),
p-ctrl-stk-size (*ctrl-stk*))))
$\wedge$ (car (*stmt*) $=$ `'predefined-proc-call-mg`)
$\wedge$ (call-name (*stmt*) $=$ `'mg-integer-le`)
$\wedge$ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
$\wedge$ 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-le . 0)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons ('x,
                                            value (cadr (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons ('y,
                                            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-integer-le-step-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-integer-le)
 ∧   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-integer-le . 0)),
                     push (p-frame (list (cons ('ans,
                                                value (car (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          cons ('x,
                                                value (cadr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          cons ('y,
                                                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-integer-le . 1)),
                push (p-frame (list (cons ('ans,
                                           value (car (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk)))),
                                     cons ('x,
                                           value (cadr (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk)))),
                                     cons ('y,
                                           value (caddr (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk))))),
                              tag ('pc,
                                   cons (subr, length (code (cinfo))
                                               +    4))),
                      ctrl-stk),
                push (value (caddr (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-le-step-6
$((n \not\simeq 0)$

```
∧  (¬ 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-le)
∧  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-integer-le . 1)),
               push (p-frame (list (cons ('ans,
                                          value (car (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk)))),
                                    cons ('x,
                                          value (cadr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk)))),
                                    cons ('y,
                                          value (caddr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk))))),
                              tag ('pc,
                                   cons (subr, length (code (cinfo))
                                               +   4))),
                     ctrl-stk),
               push (value (caddr (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-le . 2)),
             push (p-frame (list (cons ('ans,
                                        value (car (call-actuals (stmt)),
                                               bindings (top (ctrl-stk)))),
                                  cons ('x,
                                        value (cadr (call-actuals (stmt)),
                                               bindings (top (ctrl-stk)))),
                                  cons ('y,
                                        value (caddr (call-actuals (stmt)),
                                               bindings (top (ctrl-stk))))),
                             tag ('pc,
                                  cons (subr, length (code (cinfo))
                                              +  4))),
                   ctrl-stk),
             push (rget (untag (value (caddr (call-actuals (stmt)),
                                       bindings (top (ctrl-stk)))),
                         map-down-values (mg-alist (mg-state),
                                          bindings (top (ctrl-stk)),
                                          temp-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-le-step-7

```
((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-le)
 ∧  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-integer-le . 2)),
                     push (p-frame (list (cons ('ans,
                                                value (car (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          cons ('x,
                                                value (cadr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          cons ('y,
                                                value (caddr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk))))),
                                    tag ('pc,
                                         cons (subr, length (code (cinfo))
                                                     +  4))),
                           ctrl-stk),
                     push (rget (untag (value (caddr (call-actuals (stmt)),
                                               bindings (top (ctrl-stk)))),
                                 map-down-values (mg-alist (mg-state),
                                                  bindings (top (ctrl-stk)),
                                                  temp-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-le . 3)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
```

```
                                  bindings (top (ctrl-stk)))),
                        cons ('x,
                              value (cadr (call-actuals (stmt)),
                                     bindings (top (ctrl-stk)))),
                        cons ('y,
                              value (caddr (call-actuals (stmt)),
                                     bindings (top (ctrl-stk))))),
                   tag ('pc,
                        cons (subr, length (code (cinfo))
                                    +  4))),
          ctrl-stk),
     push (value (cadr (call-actuals (stmt)),
                  bindings (top (ctrl-stk))),
           push (rget (untag (value (caddr (call-actuals (stmt)),
                                     bindings (top (ctrl-stk)))),
                       map-down-values (mg-alist (mg-state),
                                        bindings (top (ctrl-stk)),
                                        temp-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-le-step-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-integer-le)
  ∧  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-integer-le . 3)),
                       push (p-frame (list (cons ('ans,
                                                  value (car (call-actuals (stmt)),
                                                         bindings (top (ctrl-stk)))),
                                            cons ('x,
                                                  value (cadr (call-actuals (stmt)),
                                                         bindings (top (ctrl-stk)))),
                                            cons ('y,
                                                  value (caddr (call-actuals (stmt)),
                                                         bindings (top (ctrl-stk))))),
                                      tag ('pc,
                                           cons (subr, length (code (cinfo))
                                                       +    4))),
                             ctrl-stk),
                       push (value (cadr (call-actuals (stmt)),
                                    bindings (top (ctrl-stk))),
                             push (rget (untag (value (caddr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                         map-down-values (mg-alist (mg-state),
                                                          bindings (top (ctrl-stk)),
                                                          temp-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-le . 4)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons ('x,
```

```
                                   value (cadr (call-actuals (stmt)),
                                             bindings (top (ctrl-stk)))),
                               cons ('y,
                                   value (caddr (call-actuals (stmt)),
                                             bindings (top (ctrl-stk))))),
                            tag ('pc,
                                 cons (subr, length (code (cinfo))
                                                + 4))),
                  ctrl-stk),
              push (rget (untag (value (cadr (call-actuals (stmt)),
                                      bindings (top (ctrl-stk)))),
                      map-down-values (mg-alist (mg-state),
                                            bindings (top (ctrl-stk)),
                                            temp-stk)),
                  push (rget (untag (value (caddr (call-actuals (stmt)),
                                            bindings (top (ctrl-stk)))),
                           map-down-values (mg-alist (mg-state),
                                                 bindings (top (ctrl-stk)),
                                                 temp-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-le-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-integer-le)
 ∧  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-integer-le . 4)),
                     push (p-frame (list (cons ('ans,
                                                value (car (call-actuals (stmt)),
                                                            bindings (top (ctrl-stk)))),
                                          cons ('x,
                                                value (cadr (call-actuals (stmt)),
                                                            bindings (top (ctrl-stk)))),
                                          cons ('y,
                                                value (caddr (call-actuals (stmt)),
                                                            bindings (top (ctrl-stk))))),
                                    tag ('pc,
                                         cons (subr, length (code (cinfo))
                                                      +  4))),
                          ctrl-stk),
                     push (rget (untag (value (cadr (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                 map-down-values (mg-alist (mg-state),
                                                  bindings (top (ctrl-stk)),
                                                  temp-stk)),
                           push (rget (untag (value (caddr (call-actuals (stmt)),
                                                            bindings (top (ctrl-stk)))),
                                       map-down-values (mg-alist (mg-state),
                                                        bindings (top (ctrl-stk)),
                                                        temp-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-le . 5)),
```

```
push (p-frame (list (cons ('ans,
                           value (car (call-actuals (stmt)),
                                  bindings (top (ctrl-stk)))),
                     cons ('x,
                           value (cadr (call-actuals (stmt)),
                                  bindings (top (ctrl-stk)))),
                     cons ('y,
                           value (caddr (call-actuals (stmt)),
                                  bindings (top (ctrl-stk))))),
                tag ('pc,
                     cons (subr, length (code (cinfo))
                                 +   4))),
      ctrl-stk),
push (tag ('bool,
           if ilessp (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                                 mg-alist (mg-state))))),
                      untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                 mg-alist (mg-state))))))
           then 't
           else 'f endif),
      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-le-step-10

```
 ((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-le)
  ∧   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)
∧   (lt-value ∈ '(t f)))
→   (p-step (p-state (tag ('pc, '(mg-integer-le . 5)),
                      push (p-frame (list (cons ('ans,
                                                 value (car (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk)))),
                                           cons ('x,
                                                 value (cadr (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk)))),
                                           cons ('y,
                                                 value (caddr (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk))))),
                                     tag ('pc,
                                          cons (subr, length (code (cinfo))
                                                      +   4))),
                            ctrl-stk),
                      push (tag ('bool, lt-value),
                            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-le . 6)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons ('x,
                                            value (cadr (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons ('y,
```

```
                                    value (caddr (call-actuals (stmt)),
                                           bindings (top (ctrl-stk))))),
                        tag ('pc,
                             cons (subr, length (code (cinfo))
                                          +    4))),
          ctrl-stk),
   push (tag ('bool, not-bool (lt-value)),
          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-le-step-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-integer-le)
  ∧  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-integer-le . 6)),
                      push (p-frame (list (cons ('ans,
                                                 value (car (call-actuals (stmt)),
```

```
                                                  bindings (top (ctrl-stk)))),
                                    cons ('x,
                                          value (cadr (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk)))),
                                    cons ('y,
                                          value (caddr (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk))))),
                                tag ('pc,
                                    cons (subr, length (code (cinfo))
                                                 +   4))),
                    ctrl-stk),
              push (tag ('bool, not-bool (lt-value)),
                    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-le . 7)),
           push (p-frame (list (cons ('ans,
                                          value (car (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk)))),
                                    cons ('x,
                                          value (cadr (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk)))),
                                    cons ('y,
                                          value (caddr (call-actuals (stmt)),
                                                  bindings (top (ctrl-stk))))),
                                tag ('pc,
                                    cons (subr, length (code (cinfo))
                                                 +   4))),
                    ctrl-stk),
           push (value (car (call-actuals (stmt)),
                         bindings (top (ctrl-stk))),
                 push (tag ('bool, not-bool (lt-value)),
                       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-le-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-integer-le)
 ∧  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-integer-le . 7)),
                     push (p-frame (list (cons ('ans,
                                                value (car (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          cons ('x,
                                                value (cadr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          cons ('y,
                                                value (caddr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk))))),
                                    tag ('pc,
                                         cons (subr, length (code (cinfo))
                                                     +   4))),
                           ctrl-stk),
                     push (value (car (call-actuals (stmt)),
```

```
                bindings (top (ctrl-stk))),
           push (tag ('bool, not-bool (lt-value)),
                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-le . 8)),
     push (p-frame (list (cons ('ans,
                              value (car (call-actuals (stmt)),
                                     bindings (top (ctrl-stk)))),
                        cons ('x,
                              value (cadr (call-actuals (stmt)),
                                     bindings (top (ctrl-stk)))),
                        cons ('y,
                              value (caddr (call-actuals (stmt)),
                                     bindings (top (ctrl-stk))))),
                   tag ('pc,
                        cons (subr, length (code (cinfo))
                                    +  4))),
         ctrl-stk),
     rput (tag ('bool, not-bool (lt-value)),
          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-le-step-13-true-case
((*n* $\not\simeq$ 0)
 $\wedge$ ($\neg$ 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-le)
∧  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)
∧  ilessp (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                       mg-alist (mg-state))))),
          untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                       mg-alist (mg-state)))))))
→  (p-step (p-state (tag ('pc, '(mg-integer-le . 8)),
                     push (p-frame (list (cons ('ans,
                                               value (car (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk)))),
                                         cons ('x,
                                               value (cadr (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk)))),
                                         cons ('y,
                                               value (caddr (call-actuals (stmt)),
                                                      bindings (top (ctrl-stk))))),
                                   tag ('pc,
                                        cons (subr, length (code (cinfo))
                                                    +   4))),
                           ctrl-stk),
                     rput (tag ('bool, not-bool ('t)),
                           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-le-step-13-false-case

```
 ((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-le)
  ∧   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)
  ∧   (¬ ilessp (untag (mg-to-p-simple-literal (caddr (assoc (caddr (call-actuals (stmt)),
                                                              mg-alist (mg-state))))),
                 untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                              mg-alist (mg-state))))))))
 →   (p-step (p-state (tag ('pc, '(mg-integer-le . 8)),
                      push (p-frame (list (cons ('ans,
                                                 value (car (call-actuals (stmt)),
                                                        bindings (top (ctrl-stk)))),
                                           cons ('x,
```

```
                                          value (cadr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk)))),
                                    cons ('y,
                                          value (caddr (call-actuals (stmt)),
                                                 bindings (top (ctrl-stk))))),
                               tag ('pc,
                                    cons (subr, length (code (cinfo))
                                                + 4))),
                     ctrl-stk),
               rput (tag ('bool, not-bool ('f)),
                     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-le-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-le)
 ∧ 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-UNARY-MINUS                   ;;
;;                                                                            ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

THEOREM: min-int-only-non-negatable-small-int
(small-integerp (x, n) $\land$ ($x \neq$ ($-$ exp (2, $n - 1$))))
$\rightarrow$ small-integerp (inegate (x), n)

THEOREM: mg-integer-unary-minus-args-have-simple-mg-type-refps
((car (*stmt*) = '`predefined-proc-call-mg`)
$\land$ (call-name (*stmt*) = '`mg-integer-unary-minus`)
$\land$ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
$\land$ ok-mg-statep (*mg-state*, *r-cond-list*)
$\land$ signatures-match (mg-alist (*mg-state*), *name-alist*))
$\rightarrow$ (int-identifierp (car (call-actuals (*stmt*)), mg-alist (*mg-state*))
$\land$ int-identifierp (cadr (call-actuals (*stmt*)), mg-alist (*mg-state*)))

THEOREM: mg-integer-unary-minus-args-definedp
((car (*stmt*) = '`predefined-proc-call-mg`)
$\land$ (call-name (*stmt*) = '`mg-integer-unary-minus`)
$\land$ ok-mg-statement (*stmt*, *r-cond-list*, *name-alist*, *proc-list*)
$\land$ ok-mg-statep (*mg-state*, *r-cond-list*)

```
 ∧  signatures-match (mg-alist (mg-state), name-alist))
 →  (definedp (car (call-actuals (stmt)), mg-alist (mg-state))
      ∧  definedp (cadr (call-actuals (stmt)), mg-alist (mg-state)))
```

THEOREM: mg-integer-unary-minus-steps-1-2

```
((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-unary-minus)
 ∧  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-integer-unary-minus-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-integer-unary-minus)
 ∧  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-integer-unary-minus . 0)),
                   push (p-frame (cons (cons ('ans,
                                              value (car (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                        cons (cons ('x,
                                                    value (cadr (call-actuals (stmt)),
                                                           bindings (top (ctrl-stk)))),
                                              '((min-int int -2147483648)
                                                (temp-x int 0)))),
                                  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-integer-unary-minus-steps-4-8

```
 ((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-integer-unary-minus)
  ∧  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-unary-minus
                                                    . 0)),
                                               push (p-frame (cons (cons ('ans,
                                                                          value (car (call-actuals (stmt)),
                                                                                 bindings (top (ctrl-stk)))),
                                                                    cons (cons ('x,
                                                                                value (cadr (call-actuals (stmt)),
                                                                                       bindings (top (ctrl-stk)))),
                                                                          '((min-int
                                                                             int
                                                                             -2147483648)
                                                                            (temp-x
                                                                             int
                                                                             0)))),
                                                             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-integer-unary-minus . 5)),
                 push (p-frame (list (cons ('ans,
                                            value (car (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      cons ('x,
                                            value (cadr (call-actuals (stmt)),
                                                   bindings (top (ctrl-stk)))),
                                      '(min-int int -2147483648),
                                      cons ('temp-x,
```

```
                              mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                   mg-alist (mg-state)))))),
                         tag ('pc,
                              cons (subr, length (code (cinfo))
                                          +   3))),
                 ctrl-stk),
             push (tag ('bool,
                        if untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                       mg-alist (mg-state)))))
                           =   untag ('(int -2147483648))
                        then 't
                        else 'f endif),
                 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))
```

```
;; (test-bool-and-jump f 0)
;; (push-constant (nat 1))
;; (pop-global c-c)
;; (jump 1)
;; (dl 1 nil (ret))
```

```
THEOREM: mg-integer-unary-minus-steps-9-13-error-case
 ((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-unary-minus)
  ∧   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)
∧  (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                 mg-alist (mg-state)))))
     =  untag ('(int -2147483648)))
→  (p-step (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                          '(mg-integer-unary-minus
                                                            . 5)),
                                                     push (p-frame (list (cons ('ans,
                                                                                value (car (call-actuals (stmt)),
                                                                                       bindings (top (ctrl-stk)))),
                                                                          cons ('x,
                                                                                value (cadr (call-actuals (stmt)),
                                                                                       bindings (top (ctrl-stk)))),
                                                                          '(min-int
                                                                            int
                                                                            -2147483648),
                                                                          cons ('temp-x,
                                                                                mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                                                      mg-alist (mg-state)))))),
                                                                     tag ('pc,
                                                                          cons (subr,
                                                                                length (code (cinfo))
                                                                                + 3))),
                                                          ctrl-stk),
                                                     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)) + 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, '(nat 1))),
                 MG-MAX-CTRL-STK-SIZE,
                 MG-MAX-TEMP-STK-SIZE,
                 MG-WORD-SIZE,
                 'run))
```

THEOREM: mg-integer-unary-minus-steps-9-12-nonerror-case

```
((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-unary-minus)
 ∧  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)
 ∧  (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                  mg-alist (mg-state)))))
      ≠  untag ('(int -2147483648))))
→   (p-step (p-step (p-step (p-step (p-state (tag ('pc,
                                                   '(mg-integer-unary-minus
                                                     . 5)),
                                              push (p-frame (list (cons ('ans,
                                                                         value (car (call-actuals (stmt)),
                                                                                bindings (top (ctrl-stk)))),
```

```
                                          cons ('x,
                                                value (cadr (call-actuals (stmt)),
                                                       bindings (top (ctrl-stk)))),
                                          '(min-int
                                            int
                                            -2147483648),
                                          cons ('temp-x,
                                                mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                      mg-alist (mg-state)))))),
                                     tag ('pc,
                                          cons (subr,
                                                length (code (cinfo))
                                                +   3))),
                            ctrl-stk),
                     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)))))
=   p-state (tag ('pc, '(mg-integer-unary-minus . 12)),
         push (p-frame (list (cons ('ans,
                                    value (car (call-actuals (stmt)),
                                           bindings (top (ctrl-stk)))),
                              cons ('x,
                                    value (cadr (call-actuals (stmt)),
                                           bindings (top (ctrl-stk)))),
                              '(min-int int -2147483648),
                              cons ('temp-x,
                                    mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                          mg-alist (mg-state)))))),
                        tag ('pc,
                             cons (subr, length (code (cinfo))
                                         +   3))),
                ctrl-stk),
         push (value (car (call-actuals (stmt)),
                      bindings (top (ctrl-stk))),
               push (tag ('int,
```

```
                              inegate (untag (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))
```

```
;; (deposit-temp-stk)
;; (dl 1 nil (ret))
```

Theorem: mg-integer-unary-minus-steps-13-14-nonerror-case

```
 ((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-unary-minus)
  ∧  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)
  ∧  (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                   mg-alist (mg-state)))))
        ≠   untag ('(int -2147483648))))
 →   (p-step (p-step (p-state (tag ('pc,
```

```
                        '(mg-integer-unary-minus . 12)),
                    push (p-frame (list (cons ('ans,
                                              value (car (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                         cons ('x,
                                              value (cadr (call-actuals (stmt)),
                                                     bindings (top (ctrl-stk)))),
                                         '(min-int int
                                           -2147483648),
                                         cons ('temp-x,
                                              mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                                                    mg-alist (mg-state)))))),
                                    tag ('pc,
                                         cons (subr,
                                              length (code (cinfo))
                                              +   3))),
                         ctrl-stk),
                    push (value (car (call-actuals (stmt)),
                                bindings (top (ctrl-stk))),
                         push (tag ('int,
                                    inegate (untag (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, cons (subr, length (code (cinfo)) + 3)),
             ctrl-stk,
             rput (tag ('int,
                        inegate (untag (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-integer-unary-minus-push-c-c-effect

```
((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-unary-minus)
 ∧  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)
 ∧  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,
                      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)) + 4)),
                  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-unary-minus-sub1-nat-effect

```
((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-unary-minus)
 ∧  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)) + 4)),
                      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)) + 5)),
                  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-unary-minus-step-16-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-unary-minus)
∧   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)
∧   (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                  mg-alist (mg-state)))))
     =  untag ('(int -2147483648))))
→   (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 5)),
                      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-unary-minus-step-17-nonerror
 (($n \not\simeq 0$)
 $\land$ ($\neg$ 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-unary-minus)
 ∧  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)
 ∧  (untag (mg-to-p-simple-literal (caddr (assoc (cadr (call-actuals (stmt)),
                                                  mg-alist (mg-state)))))
       ≠  untag ('(int -2147483648))))
 →  (p-step (p-state (tag ('pc, cons (subr, length (code (cinfo)) + 5)),
                      ctrl-stk,
                      push (tag ('nat,
                                 untag (mg-cond-to-p-nat (cc (mg-state),
                                                          t-cond-list)) - 1),
                            rput (tag ('int,
                                       inegate (untag (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-integer-unary-minus-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-unary-minus)
 ∧  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-predefined2".

Index