```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; In an unpublished paper, Textbook Examples of Recursion, Donald E.
; Knuth of Stanford University gives the following generalization of
; McCarthy's 91 function:

; Let a be a real, let b and d be positive reals, and let c be a
; positive integer.

; Define K( x ) for integer inputs x by

;   K( x ) <== if  x > a  then  x - b
;                         else  K( ... K( x+d ) ... ).

; Here the else-clause in this definition has c applications of the
; function K.

; When a = 100, b = 10, c = 2, and d = 11, the definition specializes
; to McCarthy's original 91 function:

;   K( x ) <== if  x > 100  then  x - 10
;                           else  K( K( x+11 ) ).

; Knuth calls the first definition of K given above, the generalized
; 91 recursion scheme with parameters ( a,b,c,d ).

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; The purpose of this file of Boyer-Moore-Kaufmann events is to
; provide mechanical verification of the following theorem given by
; Knuth in his paper.

; Theorem.  The generalized 91 recursion with parameters ( a,b,c,d )
;           defines a total function on the integers if and only if
;           (c-1)b < d.  In such a case the values of K( x ) also
;           satisfy the much simpler recurrence

;           K( x )  =  if  x > a  then  x - b
;                                 else  K( x+d-(c-1)b ).

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Use the library of integer facts.
```

EVENT: Start with the library "integers".

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Introduce a new function FN of one argument with
;  no constraints.
```

CONSERVATIVE AXIOM: fn-intro
t

Simultaneously, we introduce the new function symbol *fn*.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Recursively define what it means to
; iterate applications of FN.
```

DEFINITION:
iter-fn $(i,\ x)$
$=$ **if** $i \simeq 0$ **then** x
else fn (iter-fn $(i - 1,\ x)$) **endif**

THEOREM: iter-fn=fn
iter-fn $(1,\ x) = $ fn (x)

THEOREM: iter-fn-sum
iter-fn $(i,$ iter-fn $(j,\ x)) = $ iter-fn $(i + j,\ x)$

THEOREM: iter-fn-sum-integer
(ilessp $(0,\ i)$ $\wedge$ ilessp $(0,\ j)$)
$\rightarrow$ (iter-fn $(i,$ iter-fn $(j,\ x)) = $ iter-fn (iplus $(i,\ j),\ x$))

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Introduce new constants (ie., functions with no arguments)
;  P, Q, R, and S with the indicated constaints
```

CONSERVATIVE AXIOM: p-q-r-s-intro
integerp (P)
$\wedge$ integerp (Q)
$\wedge$ integerp (R)
$\wedge$ integerp (S)
$\wedge$ ($\neg$ ilessp (Q, 0))
$\wedge$ ($\neg$ ilessp (P, S))

Simultaneously, we introduce the new function symbols p, q, r, and s.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define a "general Knuth" function, GK, of one argument
;  which will be used later to help witness the existence
;  of functions which satisfy the defining recursion of
;  the generalized 91 function.
```

DEFINITION:
k-measure(a, x)
$=$ **if** ilessp(a, x) **then** 0
else iplus$(1, \text{iplus}(a, \text{ineg}(x)))$ **endif**

THEOREM: k-measure-returns-numberp
k-measure$(a, x) \in \mathbf{N}$

THEOREM: k-measure-x+y<k-measure-y
$((\neg\ \text{ilessp}(a, y)) \wedge \text{ilessp}(0, x))$
$\rightarrow\ (\text{k-measure}(a, \text{iplus}(x, y)) < \text{k-measure}(a, y))$

EVENT: Disable k-measure.

DEFINITION:
gk(x)
$=$ **if** ilessp(P, x) **then** iplus$(x, \text{ineg}(\text{Q}))$
elseif ilessp$(0, \text{R})$ **then** gk$(\text{iplus}(\text{R}, x))$
else S **endif**

DEFINITION:
iter-gk(i, x)
$=$ **if** $i \simeq 0$ **then** x
else gk$(\text{iter-gk}(i - 1, x))$ **endif**

THEOREM: iter-gk=gk
iter-gk$(1, x) = \text{gk}(x)$

THEOREM: itergk-x=itergk-x+r
$((0 < i) \wedge \text{ilessp}(0, \text{R}) \wedge (\neg\ \text{ilessp}(\text{P}, x)))$
$\rightarrow\ (\text{iter-gk}(i, x) = \text{iter-gk}(i, \text{iplus}(\text{R}, x)))$

THEOREM: count_x-1<count_x
$(\text{ilessp}(0, x) \wedge (x \neq 1)) \rightarrow (\text{count}(\text{iplus}(-1, x)) < \text{count}(x))$

DEFINITION:
induct-hint-pos-int (x)
$=$ **if** $\neg$ ilessp $(\texttt{0}, x)$ **then t**
elseif $x = \texttt{1}$ **then t**
else induct-hint-pos-int (iplus $(\texttt{-1}, x)$) **endif**

THEOREM: n-1>0-when-n>0&n<>1
(ilessp $(\texttt{0}, n) \wedge (n \neq \texttt{1})) \rightarrow$ ilessp ($\texttt{0}$, iplus $(\texttt{-1}, n)$)

THEOREM: x<=z-when-x<y&y<=z
(ilessp $(x, y) \wedge (\neg$ ilessp $(z, y))) \rightarrow (\neg$ ilessp $(z, x))$

THEOREM: x+_n-1_r<x+nr-when-0<r
ilessp $(\texttt{0}, r)$
$\rightarrow$ ilessp (iplus (x, itimes (iplus $(\texttt{-1}, n)$, r)), iplus (x, itimes (n, r)))

THEOREM: p+r>=x+_n-1_r-when-p+r>=x+nr
(ilessp $(\texttt{0}, \text{R}) \wedge (\neg$ ilessp (iplus (P, R), iplus (x, itimes (n, R)))))
$\rightarrow$ ($\neg$ ilessp (iplus (P, R), iplus (x, itimes (iplus $(\texttt{-1}, n)$, R))))

THEOREM: x>=y-z-when-x+z>=y
($\neg$ ilessp (iplus (x, z), y)) $\rightarrow$ ($\neg$ ilessp (x, iplus (y, ineg (z))))

THEOREM: itergk-x=itergk-x+nr-induction-step
(ilessp $(\texttt{0}, n)$
$\wedge$ $(n \neq \texttt{1})$
$\wedge$ ((($\texttt{0} < i$)
$\wedge$ ilessp $(\texttt{0}, \text{R})$
$\wedge$ ilessp ($\texttt{0}$, iplus $(\texttt{-1}, n)$)
$\wedge$ ($\neg$ ilessp (iplus (P, R), iplus (x, itimes (iplus $(\texttt{-1}, n)$, R)))))
$\rightarrow$ (iter-gk (i, x) = iter-gk (i, iplus (x, itimes (iplus $(\texttt{-1}, n)$, R)))))
$\wedge$ $(\texttt{0} < i)$
$\wedge$ ilessp $(\texttt{0}, \text{R})$
$\wedge$ ($\neg$ ilessp (iplus (P, R), iplus (x, itimes (n, R)))))
$\rightarrow$ ((iter-gk (i, x) = iter-gk (i, iplus (x, itimes (R, n)))) = **t**)

THEOREM: itergk-x=itergk-x+nr
(($\texttt{0} < i$)
$\wedge$ ilessp $(\texttt{0}, \text{R})$
$\wedge$ ilessp $(\texttt{0}, n)$
$\wedge$ ($\neg$ ilessp (iplus (P, R), iplus (x, itimes (n, R)))))
$\rightarrow$ (iter-gk (i, x) = iter-gk (i, iplus (x, itimes (n, R))))

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define the function N( a,d,x ) recursively,
;  so that whenever d > 0, N( a,d,x ) is the smallest
;  nonnegative integer i such that x + id > a.
```

DEFINITION:
n $(a,\ d,\ x)$
$=$ **if** $\neg$ ilessp $(\mathtt{0},\ d)$ **then** $\mathtt{0}$
elseif ilessp $(a,\ x)$ **then** $\mathtt{0}$
else iplus $(\mathtt{1},\ \mathrm{n}\,(a,\ d,\ \mathrm{iplus}\,(x,\ d)))$ **endif**

THEOREM: n>=0
$\neg\ \mathrm{ilessp}\,(\mathrm{n}\,(a,\ d,\ x),\ \mathtt{0})$

THEOREM: n>0-when-d>o&x<=a
$(\mathrm{ilessp}\,(\mathtt{0},\ d) \wedge (\neg\ \mathrm{ilessp}\,(a,\ x))) \rightarrow \mathrm{ilessp}\,(\mathtt{0},\ \mathrm{n}\,(a,\ d,\ x))$

THEOREM: a<x+nd
$\mathrm{ilessp}\,(\mathtt{0},\ d) \rightarrow \mathrm{ilessp}\,(a,\ \mathrm{iplus}\,(x,\ \mathrm{itimes}\,(\mathrm{n}\,(a,\ d,\ x),\ d)))$

THEOREM: a+d>=x+nd-when-d>0&a>=x
$(\mathrm{ilessp}\,(\mathtt{0},\ d) \wedge (\neg\ \mathrm{ilessp}\,(a,\ x)))$
$\rightarrow\ (\neg\ \mathrm{ilessp}\,(\mathrm{iplus}\,(a,\ d),\ \mathrm{iplus}\,(x,\ \mathrm{itimes}\,(\mathrm{n}\,(a,\ d,\ x),\ d))))$

THEOREM: itergk_i_x=itergk_i-1_x-q+nr-step-1
$((\mathtt{0} < i) \wedge \mathrm{ilessp}\,(\mathtt{0},\ \mathrm{R}) \wedge (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$
$\rightarrow\ (\mathrm{iter\text{-}gk}\,(i,\ x) = \mathrm{iter\text{-}gk}\,(i,\ \mathrm{iplus}\,(x,\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x)))))$

THEOREM: itergk_i_x=itergk_i-1_x-q+nr-step-2
$(\mathtt{0} < i)$
$\rightarrow\ (\mathrm{iter\text{-}gk}\,(i,\ \mathrm{iplus}\,(x,\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x))))$
$=\ \mathrm{iter\text{-}gk}\,(i-1,\ \mathrm{gk}\,(\mathrm{iplus}\,(x,\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x))))))$

THEOREM: itergk_i_x=itergk_i-1_x-q+nr-step-3
$((\mathtt{0} < i) \wedge \mathrm{ilessp}\,(\mathtt{0},\ \mathrm{R}) \wedge (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$
$\rightarrow\ (\mathrm{iter\text{-}gk}\,(i-1,\ \mathrm{gk}\,(\mathrm{iplus}\,(x,\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x)))))$
$=\ \mathrm{iter\text{-}gk}\,(i-1,\ \mathrm{iplus}\,(x,\ \mathrm{iplus}\,(\mathrm{ineg}\,(\mathrm{Q}),\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x))))))$

THEOREM: itergk_i_x=itergk_i-1_x-q+nr
$((\mathtt{0} < i) \wedge \mathrm{ilessp}\,(\mathtt{0},\ \mathrm{R}) \wedge (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$
$\rightarrow\ (\mathrm{iter\text{-}gk}\,(i,\ x)$
$=\ \mathrm{iter\text{-}gk}\,(i-1,\ \mathrm{iplus}\,(x,\ \mathrm{iplus}\,(\mathrm{ineg}\,(\mathrm{Q}),\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x))))))$

THEOREM: x>=z-when-x>=y&y>=z
$((\neg\ \mathrm{ilessp}\,(x,\ y)) \wedge (\neg\ \mathrm{ilessp}\,(y,\ z))) \rightarrow (\neg\ \mathrm{ilessp}\,(x,\ z))$

THEOREM: x+y>=x+-z+y-when-z>=0
$(\neg\ \mathrm{ilessp}\,(z,\ \mathtt{0})) \rightarrow (\neg\ \mathrm{ilessp}\,(\mathrm{iplus}\,(x,\ y),\ \mathrm{iplus}\,(\mathrm{iplus}\,(x,\ \mathrm{ineg}\,(z)),\ y)))$

THEOREM: itergk_i-1_x-q=itergk_i-1_x-q+nr
$((\mathtt{1} < i) \wedge \mathrm{ilessp}\,(\mathtt{0},\ \mathrm{R}) \wedge (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$
$\rightarrow\ (\mathrm{iter\text{-}gk}\,(i-1,\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{Q})))$
$=\ \mathrm{iter\text{-}gk}\,(i-1,\ \mathrm{iplus}\,(x,\ \mathrm{iplus}\,(\mathrm{ineg}\,(\mathrm{Q}),\ \mathrm{itimes}\,(\mathrm{R},\ \mathrm{n}\,(\mathrm{P},\ \mathrm{R},\ x))))))$

THEOREM: itergk_i_x=itergk_i-1_x-q-when-p>=x&r>0

$$((1 < i) \land \mathrm{ilessp}\,(0,\ \mathrm{R}) \land (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x) = \text{iter-gk}\,(i - 1,\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{Q}))))$$

THEOREM: itergk=s-when-p>=x&0>=r&0<i

$$((0 < i) \land (\neg\ \mathrm{ilessp}\,(0,\ \mathrm{R})) \land (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x) = \mathrm{S})$$

THEOREM: x>=x+-y-when-y>=0

$$(\neg\ \mathrm{ilessp}\,(y,\ 0)) \rightarrow (\neg\ \mathrm{ilessp}\,(x,\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(y))))$$

THEOREM: itergk_i_x=itergk_i-1_x-q-when-p>=x&0>=r

$$((1 < i) \land (\neg\ \mathrm{ilessp}\,(0,\ \mathrm{R})) \land (\neg\ \mathrm{ilessp}\,(\mathrm{P},\ x)))$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x) = \text{iter-gk}\,(i - 1,\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{Q}))))$$

THEOREM: itergk_i_x=itergk_i-1_x-q-when-p<x

$$((0 < i) \land \mathrm{ilessp}\,(\mathrm{P},\ x))$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x) = \text{iter-gk}\,(i - 1,\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{Q}))))$$

THEOREM: itergk_i_x=itergk_i-1_x-q

$$(1 < i) \rightarrow (\text{iter-gk}\,(i,\ x) = \text{iter-gk}\,(i - 1,\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{Q}))))$$

THEOREM: itergk_i_x=itergk_i-1_x-q-integer

$$\mathrm{ilessp}\,(1,\ i) \rightarrow (\text{iter-gk}\,(i,\ x) = \text{iter-gk}\,(\mathrm{iplus}\,(\texttt{-1},\ i),\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{Q}))))$$

THEOREM: one-one+y=y

$$\mathrm{integerp}\,(y) \rightarrow (\mathrm{iplus}\,(1,\ \mathrm{iplus}\,(\texttt{-1},\ y)) = y)$$

THEOREM: itergk_i_x=itergk_i-j_x-jq

$$(\mathrm{ilessp}\,(0,\ j) \land \mathrm{ilessp}\,(j,\ i))$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x)$$
$$=\quad \text{iter-gk}\,(\mathrm{iplus}\,(i,\ \mathrm{ineg}\,(j)),\ \mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{itimes}\,(j,\ \mathrm{Q})))))$$

THEOREM: itergk_i_x=gk_x-_i-1_q-when-i>1

$$\mathrm{ilessp}\,(1,\ i)$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x) = \mathrm{gk}\,(\mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{itimes}\,(\mathrm{iplus}\,(\texttt{-1},\ i),\ \mathrm{Q})))))$$

THEOREM: x=1-or-x>1-when-x>0

$$\mathrm{ilessp}\,(0,\ x) \rightarrow ((x = 1) \lor \mathrm{ilessp}\,(1,\ x))$$

THEOREM: itergk_i_x=gk_x-_i-1_q

$$\mathrm{ilessp}\,(0,\ i)$$
$$\rightarrow\quad (\text{iter-gk}\,(i,\ x) = \mathrm{gk}\,(\mathrm{iplus}\,(x,\ \mathrm{ineg}\,(\mathrm{itimes}\,(\mathrm{iplus}\,(\texttt{-1},\ i),\ \mathrm{Q})))))$$

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Introduce new constants (ie., functions with no arguments)
;  A, B, C, and E with the indicated constaints.
```

Conservative Axiom: a-b-c-d-intro

integerp (A)
∧ integerp (B)
∧ integerp (C)
∧ integerp (D)
∧ ilessp (0, B)
∧ ilessp (0, C)
∧ ilessp (0, D)

Simultaneously, we introduce the new function symbols a, b, c, and d.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define two functions K and K1 which will be shown to satisfy
; the generalized 91 recursion scheme with parameters
; ( (A),(B),(C),(D) ).

#| The following is the original version, which failed to be the
   correct lemma for the acceptance of the definition of K in Nqthm-1992.
   Following that is a new version.

(PROVE-LEMMA K-MEASURE-DECREASES-DEFN-K&K1
  (REWRITE)
  (IMPLIES
       (AND (NOT (ILESSP (A) X))
            (ILESSP 0
                    (IPLUS (D)
                           (IPLUS (INEG (ITIMES (B) (C)))
                                  (INEG (ITIMES (B) -1))))))
       (LESSP
        (K-MEASURE
                (A)
                (IPLUS (D)
                       (IPLUS (INEG (ITIMES (B) (C)))
                              (IPLUS (INEG (ITIMES (B) -1)) X))))
        (K-MEASURE (A) X)))
  ; hint
  ( (USE (K-MEASURE-X+Y<K-MEASURE-Y
  (A (A))
  (Y X)
  (X (IPLUS (D)
    (IPLUS (INEG (ITIMES (B) (C)))
   (INEG (ITIMES (B) -1))))))) ))
|#
```

THEOREM: k-measure-decreases-defn-k&k1
((¬ ilessp (A, x)) ∧ ilessp (itimes (B, C), iplus (B, D)))
→ (k-measure (A, iplus (B, iplus (D, iplus (ineg (itimes (B, C)), x))))
< k-measure (A, x))

DEFINITION:
k (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
elseif ilessp (0, iplus (D, ineg (itimes (B, iplus (-1, C)))))
then k (iplus (x, iplus (D, ineg (itimes (B, iplus (-1, C))))))
else A **endif**

DEFINITION:
k1 (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
elseif ilessp (0, iplus (D, ineg (itimes (B, iplus (-1, C)))))
then k1 (iplus (x, iplus (D, ineg (itimes (B, iplus (-1, C))))))
else iplus (A, ineg (B)) **endif**

DEFINITION:
iter-k (i, x)
= **if** $i \simeq 0$ **then** x
else k (iter-k ($i - 1$, x)) **endif**

DEFINITION:
iter-k1 (i, x)
= **if** $i \simeq 0$ **then** x
else k1 (iter-k1 ($i - 1$, x)) **endif**

THEOREM: x>=x
¬ ilessp (x, x)

THEOREM: iterk_i_x=k_x-_i-1_b
ilessp (0, i) → (iter-k (i, x) = k (iplus (x, ineg (itimes (iplus (-1, i), B)))))

THEOREM: x>=x-y-when-y>0
ilessp (0, y) → (¬ ilessp (x, iplus (x, ineg (y))))

THEOREM: iterk1_i_x=k1_x-_i-1_b
ilessp (0, i)
→ (iter-k1 (i, x) = k1 (iplus (x, ineg (itimes (iplus (-1, i), B)))))

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The two functions K and K1 satisfy the generalized 91
; recursion scheme with parameters ( (A),(B),(C),(D) ):
```

THEOREM: x>=y+z-when-x>=y&0>=z

$((\neg \text{ ilessp}(x, y)) \land (\neg \text{ ilessp}(\texttt{0}, z))) \rightarrow (\neg \text{ ilessp}(x, \text{iplus}(y, z)))$

THEOREM: gk_x+r=s-when-p>=x&0>=r

$((\neg \text{ ilessp}(\text{P}, x)) \land (\neg \text{ ilessp}(\texttt{0}, \text{R}))) \rightarrow (\text{gk}(\text{iplus}(x, \text{R})) = \text{S})$

THEOREM: k_x+d-_c-1_b=a-when-a>=x&0>=d-_c-1_b

$((\neg \text{ ilessp}(\text{A}, x)) \land (\neg \text{ ilessp}(\texttt{0}, \text{iplus}(\text{D}, \text{ineg}(\text{itimes}(\text{B}, \text{iplus}(\texttt{-1}, \text{C})))))))$
$\rightarrow \quad (\text{k}(\text{iplus}(x, \text{iplus}(\text{D}, \text{ineg}(\text{itimes}(\text{B}, \text{iplus}(\texttt{-1}, \text{C})))))) = \text{A})$

THEOREM: k1_x+d-_c-1_b=a-b-when-a>=x&0>=d-_c-1_b

$((\neg \text{ ilessp}(\text{A}, x)) \land (\neg \text{ ilessp}(\texttt{0}, \text{iplus}(\text{D}, \text{ineg}(\text{itimes}(\text{B}, \text{iplus}(\texttt{-1}, \text{C})))))))$
$\rightarrow \quad (\text{k1}(\text{iplus}(x, \text{iplus}(\text{D}, \text{ineg}(\text{itimes}(\text{B}, \text{iplus}(\texttt{-1}, \text{C}))))))$
$\quad = \quad \text{iplus}(\text{A}, \text{ineg}(\text{B})))$

THEOREM: itimes2–1

$\text{itimes}(x, \texttt{-1}) = \text{ineg}(x)$

THEOREM: k-satisfies-gen-91-recursion

$\text{k}(x)$
$= \quad$ **if** ilessp (A, x) **then** iplus (x, ineg (B))
else iter-k (C, iplus (x, D)) **endif**

THEOREM: k1-satisfies-gen-91-recursion

$\text{k1}(x)$
$= \quad$ **if** ilessp (A, x) **then** iplus (x, ineg (B))
else iter-k1 (C, iplus (x, D)) **endif**

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; So now that it has been verified that the two functions
; K and K1 satisfy the generalized 91 recursion scheme
; with parameters ( (A),(B),(C),(D) ); we now verify that
; the two functions are not the same when
; (D) <= (B)[(C)-1].
```

THEOREM: x-y<=0-when-x<=y

$(\neg \text{ ilessp}(y, x)) \rightarrow (\neg \text{ ilessp}(\texttt{0}, \text{iplus}(x, \text{ineg}(y))))$

THEOREM: x-y<x-when-0<y

$\text{ilessp}(\texttt{0}, y) \rightarrow \text{ilessp}(\text{iplus}(x, \text{ineg}(y)), x)$

THEOREM: k&k1-are-not-equal-functions

$((\neg \text{ ilessp}(\text{itimes}(\text{B}, \text{iplus}(\text{C}, \texttt{-1})), \text{D})) \land (\neg \text{ ilessp}(\text{A}, x)))$
$\rightarrow \quad (\text{k}(x) \neq \text{k1}(x))$

THEOREM: k&k1-are-not-equal-functions-version-2
(¬ ilessp (itimes (B, iplus (C, -1)), D)) → (k (A) ≠ k1 (A))

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Introduce a new function G of one argument which
; satisfies the simpler recurrence given in Knuth's
; Theorem.
```

THEOREM: gk-version-of-simple-recursion
gk (x)
= **if** ilessp (P, x) **then** iplus (x, ineg (Q))
else gk (iplus (x, R)) **endif**

EVENT: Disable k-satisfies-gen-91-recursion.

EVENT: Disable k1-satisfies-gen-91-recursion.

THEOREM: k-satisfies-simple-recursion
k (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
else k (iplus (x, iplus (D, ineg (itimes (B, iplus (-1, C)))))) **endif**

CONSERVATIVE AXIOM: g-satisfies-simple-recursion
g (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
else g (iplus (x, iplus (D, ineg (itimes (B, iplus (-1, C)))))) **endif**

Simultaneously, we introduce the new function symbol g.

THEOREM: g=x-b-when-a<x
ilessp (A, x) → (g (x) = iplus (x, ineg (B)))

EVENT: Disable g-satisfies-simple-recursion.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Verify that any function, such as G, which satisfies the
; simpler recurrence given in Knuth's Theorem, is in fact
; equal to K, provided (B)[(C) - 1] < (D):
```

THEOREM: x-y>0-when-y<x
ilessp (y, x) → ilessp (0, iplus (x, ineg (y)))

THEOREM: lessp-k-measure
$(\text{ilessp}(\text{iplus}(\text{ineg}(\text{B}), \text{itimes}(\text{B}, \text{C})), \text{D}) \land (\neg\, \text{ilessp}(\text{A}, x)))$
$\to\ (\text{k-measure}(\text{A}, \text{iplus}(\text{B}, \text{iplus}(\text{D}, \text{iplus}(\text{ineg}(\text{itimes}(\text{B}, \text{C})), x))))$
$\quad < \ \text{k-measure}(\text{A}, x))$

DEFINITION:
k-induct-hint (x)
= **if** ¬ ilessp (itimes (B, iplus (C, `-1`)), D) **then t**
elseif ilessp (A, x) **then t**
else k-induct-hint (iplus (x,
iplus (D, ineg (itimes (B, iplus (C, `-1`)))))) **endif**

THEOREM: g=k-when-b_c-1_<d-induction-step
(ilessp (itimes (B, iplus (C, `-1`)), D)
∧ (¬ ilessp (A, x))
∧ (g (iplus (x, iplus (D, ineg (itimes (B, iplus (C, `-1`))))))
= k (iplus (x, iplus (D, ineg (itimes (B, iplus (C, `-1`))))))))
→ (g (x) = k (x))

THEOREM: g=k-when-b_c-1_<d
ilessp (itimes (B, iplus (C, `-1`)), D) → (g (x) = k (x))

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Introduce a new function H of one argument which
; satisfies the general 91 recursion with parameters
; ( (A),(B),(C),(D) ).
```

CONSERVATIVE AXIOM: h-satisfies-gen-91-recursion
(h (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
else iter-h (C, iplus (x, D)) **endif**)
∧ (iter-h (i, x)
= **if** $i \simeq$ `0` **then** x
else h (iter-h ($i - 1$, x)) **endif**)

Simultaneously, we introduce the new function symbols h and *iter-h*.

THEOREM: h=x-b-when-a<x
ilessp (A, x) → (h (x) = iplus (x, ineg (B)))

THEOREM: h=iterh-when-a>=x
(¬ ilessp (A, x)) → (h (x) = iter-h (C, iplus (D, x)))

THEOREM: iterh=x-when-i-is-zerop
($i \simeq$ `0`) → (iter-h (i, x) = x)

EVENT: Disable h-satisfies-gen-91-recursion.

THEOREM: iterh=h-iter-h-when-not-zerop-i
$(i \not\simeq 0) \rightarrow (\text{iter-h}\,(i,\ x) = \text{h}\,(\text{iter-h}\,(i - 1,\ x)))$

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; To prevent looping at least one of the following
; should be disabled:
```

EVENT: Disable h=iterh-when-a>=x.

EVENT: Disable iterh=h-iter-h-when-not-zerop-i.

THEOREM: iter-h=h
$\text{iter-h}\,(\texttt{1},\ x) = \text{h}\,(x)$

THEOREM: iter-h-sum
$\text{iter-h}\,(i,\ \text{iter-h}\,(j,\ x)) = \text{iter-h}\,(i + j,\ x)$

THEOREM: iter-h-sum-integer
$(\text{ilessp}\,(\texttt{0},\ i) \wedge \text{ilessp}\,(\texttt{0},\ j))$
$\rightarrow \quad (\text{iter-h}\,(i,\ \text{iter-h}\,(j,\ x)) = \text{iter-h}\,(\text{iplus}\,(i,\ j),\ x))$

THEOREM: h-satisfies-simple-rec-when-c=1
$(\text{C} = \texttt{1})$
$\rightarrow \quad (\text{h}\,(x)$
$= \ \textbf{if}\ \text{ilessp}\,(\text{A},\ x)\ \ \textbf{then}\ \text{iplus}\,(x,\ \text{ineg}\,(\text{B}))$
$\textbf{else}\ \text{h}\,(\text{iplus}\,(x,\ \text{iplus}\,(\text{D},\ \text{ineg}\,(\text{itimes}\,(\text{B},\ \text{iplus}\,(\text{C},\ \texttt{-1}))))))\ \textbf{endif})$

THEOREM: i-is-integer-when-1<i
$\text{ilessp}\,(\texttt{1},\ i) \rightarrow \text{integerp}\,(i)$

THEOREM: iterh_i_x=iterh_i-1_x-b-when-x>a
$(\text{ilessp}\,(\texttt{1},\ i) \wedge \text{ilessp}\,(\text{A},\ x))$
$\rightarrow \quad (\text{iter-h}\,(i,\ x) = \text{iter-h}\,(\text{iplus}\,(\texttt{-1},\ i),\ \text{iplus}\,(x,\ \text{ineg}\,(\text{B}))))$

THEOREM: a>=x-when-a+d>=d+x
$(\neg\ \text{ilessp}\,(\text{iplus}\,(a,\ d),\ \text{iplus}\,(d,\ x))) \rightarrow (\neg\ \text{ilessp}\,(a,\ x))$

THEOREM: iterh_1+x_y=iterh_x_h_y
$\text{ilessp}\,(\texttt{0},\ x) \rightarrow (\text{iter-h}\,(\text{iplus}\,(\texttt{1},\ x),\ y) = \text{iter-h}\,(x,\ \text{h}\,(y)))$

THEOREM: iterh_x_h_y=iterh_x+c-1_x+d-when-x>0&a>=y
$(\text{ilessp}\,(\texttt{0},\ x) \wedge (\neg\ \text{ilessp}\,(\text{A},\ y)))$
$\rightarrow \quad (\text{iter-h}\,(x,\ \text{h}\,(y)) = \text{iter-h}\,(\text{iplus}\,(\text{C},\ x),\ \text{iplus}\,(\text{D},\ y)))$

Theorem: iterh_1_x=iterh_1+n_c-1_x+nd-induction-step-part-1

$$
\begin{array}{l}
(\text{ilessp}\,(\mathtt{0},\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \text{iplus}\,(\textsc{c},\ \mathtt{-1})))\\
\ \wedge\ \ (\neg\ \text{ilessp}\,(\textsc{a},\ \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \textsc{d})))))\\
\to\ \ (\text{iter-h}\,(\text{iplus}\,(\mathtt{1},\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \text{iplus}\,(\textsc{c},\ \mathtt{-1}))),\\
\qquad\qquad \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \textsc{d})))\\
\qquad = \ \text{iter-h}\,(\text{iplus}\,(\mathtt{1},\ \text{itimes}\,(n,\ \text{iplus}\,(\textsc{c},\ \mathtt{-1}))),\ \text{iplus}\,(x,\ \text{itimes}\,(n,\ \textsc{d}))))
\end{array}
$$

Theorem: n-1*c-1>0

$$
\begin{array}{l}
(\text{ilessp}\,(\mathtt{0},\ n) \wedge (n \neq \mathtt{1}) \wedge \text{ilessp}\,(\mathtt{1},\ \textsc{c}))\\
\to\ \ \text{ilessp}\,(\mathtt{0},\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \text{iplus}\,(\textsc{c},\ \mathtt{-1})))
\end{array}
$$

Theorem: a>=x+_n-1_d-when-a+d>=x+dn

$$
\begin{array}{l}
(\neg\ \text{ilessp}\,(\text{iplus}\,(a,\ d),\ \text{iplus}\,(x,\ \text{itimes}\,(d,\ n))))\\
\to\ \ (\neg\ \text{ilessp}\,(a,\ \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ d))))
\end{array}
$$

Theorem: iterh_1_x=iterh_1+n_c-1_x+nd-induction-step-part-2

$$
\begin{array}{l}
(\text{ilessp}\,(\mathtt{0},\ n)\\
\ \wedge\ \ (n \neq \mathtt{1})\\
\ \wedge\ \ \text{ilessp}\,(\mathtt{1},\ \textsc{c})\\
\ \wedge\ \ (\neg\ \text{ilessp}\,(\text{iplus}\,(\textsc{a},\ \textsc{d}),\ \text{iplus}\,(x,\ \text{itimes}\,(n,\ \textsc{d}))))\\
\ \wedge\ \ (\text{iter-h}\,(\mathtt{1},\ x)\\
\qquad = \ \text{iter-h}\,(\text{iplus}\,(\mathtt{1},\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \text{iplus}\,(\textsc{c},\ \mathtt{-1}))),\\
\qquad\qquad\qquad \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \textsc{d})))))\\
\to\ \ (\text{iter-h}\,(\mathtt{1},\ x)\\
\qquad = \ \text{iter-h}\,(\text{iplus}\,(\mathtt{1},\ \text{itimes}\,(n,\ \text{iplus}\,(\textsc{c},\ \mathtt{-1}))),\ \text{iplus}\,(x,\ \text{itimes}\,(n,\ \textsc{d}))))
\end{array}
$$

Theorem: a+d>=x+_n-1_d-when-a+d>=x+dn

$$
\begin{array}{l}
(\text{ilessp}\,(\mathtt{0},\ d) \wedge (\neg\ \text{ilessp}\,(\text{iplus}\,(a,\ d),\ \text{iplus}\,(x,\ \text{itimes}\,(d,\ n)))))\\
\to\ \ (\neg\ \text{ilessp}\,(\text{iplus}\,(a,\ d),\ \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ d))))
\end{array}
$$

Theorem: iterh_1_x=iterh_1+n_c-1_x+nd-induction-step

$$
\begin{array}{l}
(\text{ilessp}\,(\mathtt{0},\ n)\\
\ \wedge\ \ (n \neq \mathtt{1})\\
\ \wedge\ \ ((\text{ilessp}\,(\mathtt{1},\ \textsc{c})\\
\qquad \wedge\ \ \text{ilessp}\,(\mathtt{0},\ \text{iplus}\,(\mathtt{-1},\ n))\\
\qquad \wedge\ \ (\neg\ \text{ilessp}\,(\text{iplus}\,(\textsc{a},\ \textsc{d}),\ \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \textsc{d})))))\\
\qquad \to\ \ (\text{iter-h}\,(\mathtt{1},\ x)\\
\qquad\qquad = \ \text{iter-h}\,(\text{iplus}\,(\mathtt{1},\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \text{iplus}\,(\textsc{c},\ \mathtt{-1}))),\\
\qquad\qquad\qquad\qquad \text{iplus}\,(x,\ \text{itimes}\,(\text{iplus}\,(\mathtt{-1},\ n),\ \textsc{d})))))\\
\ \wedge\ \ \text{ilessp}\,(\mathtt{1},\ \textsc{c})\\
\ \wedge\ \ (\neg\ \text{ilessp}\,(\text{iplus}\,(\textsc{a},\ \textsc{d}),\ \text{iplus}\,(x,\ \text{itimes}\,(n,\ \textsc{d})))))\\
\to\ \ (\text{iter-h}\,(\mathtt{1},\ x)\\
\qquad = \ \text{iter-h}\,(\text{iplus}\,(\mathtt{1},\ \text{itimes}\,(n,\ \text{iplus}\,(\textsc{c},\ \mathtt{-1}))),\ \text{iplus}\,(x,\ \text{itimes}\,(n,\ \textsc{d}))))
\end{array}
$$

Theorem: iterh_1_x=iterh_1+n_c-1_x+nd-base-step

$$
\begin{array}{l}
(\text{ilessp}\,(\mathtt{1},\ \textsc{c}) \wedge (\neg\ \text{ilessp}\,(\text{iplus}\,(\textsc{a},\ \textsc{d}),\ \text{iplus}\,(\textsc{d},\ x))))\\
\to\ \ (\text{iter-h}\,(\mathtt{1},\ x) = \text{iter-h}\,(\textsc{c},\ \text{iplus}\,(\textsc{d},\ x)))
\end{array}
$$

THEOREM: iterh_1_x=iterh_1+n_c-1_x+nd
(ilessp (1, C)
 ∧ ilessp (0, n)
 ∧ (¬ ilessp (iplus (A, D), iplus (x, itimes (n, D)))))
→ (iter-h (1, x)
   = iter-h (iplus (1, itimes (n, iplus (C, -1))), iplus (x, itimes (n, D))))

THEOREM: iterh_i_x=iterh_i+n_c-1_x+nd-step-1
ilessp (1, i) → (iter-h (i, x) = iter-h (iplus (-1, i), iter-h (1, x)))

THEOREM: iterh_i_x=iterh_i+n_c-1_x+nd-step-2
(ilessp (1, C)
 ∧ ilessp (0, n)
 ∧ (¬ ilessp (iplus (A, D), iplus (x, itimes (n, D)))))
→ (iter-h (iplus (-1, i), iter-h (1, x))
   = iter-h (iplus (-1, i),
      iter-h (iplus (1, itimes (n, iplus (C, -1))),
       iplus (x, itimes (n, D)))))

THEOREM: cn-n>=0-when-1<c&0<n
(ilessp (1, c) ∧ ilessp (0, n))
→ (¬ ilessp (iplus (ineg (n), itimes (c, n)), 0))

THEOREM: iterh_i_x=iterh_i+n_c-1_x+nd-step-3
(ilessp (1, i) ∧ ilessp (1, C) ∧ ilessp (0, n))
→ (iter-h (iplus (-1, i),
     iter-h (iplus (1, itimes (n, iplus (C, -1))), iplus (x, itimes (n, D))))
   = iter-h (iplus (i, itimes (n, iplus (C, -1))), iplus (x, itimes (n, D))))

THEOREM: iterh_i_x=iterh_i+n_c-1_x+nd
(ilessp (1, i)
 ∧ ilessp (1, C)
 ∧ ilessp (0, n)
 ∧ (¬ ilessp (iplus (A, D), iplus (x, itimes (n, D)))))
→ (iter-h (i, x)
   = iter-h (iplus (i, itimes (n, iplus (C, -1))), iplus (x, itimes (n, D))))

THEOREM: iterh_i_x=iterh_i-1+n_c-1_x+nd-b-step-1
(ilessp (1, i) ∧ ilessp (1, C) ∧ (¬ ilessp (A, x)))
→ (iter-h (i, x)
   = iter-h (iplus (i, itimes (n (A, D, x), iplus (C, -1))),
      iplus (x, itimes (n (A, D, x), D))))

THEOREM: i+n_c-1_-1>0
(ilessp (1, i) ∧ ilessp (1, c) ∧ ilessp (0, n))
→ ilessp (0, iplus (-1, iplus (i, itimes (n, iplus (c, -1)))))

THEOREM: iterh_i_x=iterh_i-1+n_c-1_x+nd-b-step-2
$(\text{ilessp}(\mathtt{1},\ i) \wedge \text{ilessp}(\mathtt{1},\ \text{C}) \wedge (\neg\ \text{ilessp}(\text{A},\ x)))$
$\rightarrow\ (\text{iter-h}(\text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1}))),$
$\text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))$
$=\ \text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{h}(\text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))))$

THEOREM: iterh_i_x=iterh_i-1+n_c-1_x+nd-b-step-3
$(\text{ilessp}(\mathtt{1},\ i) \wedge \text{ilessp}(\mathtt{1},\ \text{C}) \wedge (\neg\ \text{ilessp}(\text{A},\ x)))$
$\rightarrow\ (\text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{h}(\text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D}))))$
$=\ \text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{iplus}(\text{ineg}(\text{B}),\ \text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))))$

THEOREM: iterh_i_x=iterh_i-1+n_c-1_x+nd-b
$(\text{ilessp}(\mathtt{1},\ i) \wedge \text{ilessp}(\mathtt{1},\ \text{C}) \wedge (\neg\ \text{ilessp}(\text{A},\ x)))$
$\rightarrow\ (\text{iter-h}(i,\ x)$
$=\ \text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{iplus}(\text{ineg}(\text{B}),\ \text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))))$

THEOREM: x>=y-z-when-x>=y&0<z
$((\neg\ \text{ilessp}(x,\ y)) \wedge \text{ilessp}(\mathtt{0},\ z)) \rightarrow (\neg\ \text{ilessp}(x,\ \text{iplus}(y,\ \text{ineg}(z))))$

THEOREM: x+2-1=1+x
$\text{iplus}(\mathtt{-1},\ \text{iplus}(\mathtt{2},\ x)) = \text{iplus}(\mathtt{1},\ x)$

THEOREM: iterh_i-1_x-b=iterh_i-1+n_c-1_x+nd-b-when-i=2
$((i = \mathtt{2}) \wedge \text{ilessp}(\mathtt{1},\ i) \wedge \text{ilessp}(\mathtt{1},\ \text{C}) \wedge (\neg\ \text{ilessp}(\text{A},\ x)))$
$\rightarrow\ (\text{iter-h}(\text{iplus}(\mathtt{-1},\ i),\ \text{iplus}(x,\ \text{ineg}(\text{B})))$
$=\ \text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{iplus}(\text{ineg}(\text{B}),\ \text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))))$

THEOREM: i-1>1-when-i<>2&1<i
$((i \neq \mathtt{2}) \wedge \text{ilessp}(\mathtt{1},\ i)) \rightarrow \text{ilessp}(\mathtt{1},\ \text{iplus}(\mathtt{-1},\ i))$

THEOREM: iterh_i-1_x-b=iterh_i-1+n_c-1_x+nd-b-when-i>2
$((i \neq \mathtt{2}) \wedge \text{ilessp}(\mathtt{1},\ i) \wedge \text{ilessp}(\mathtt{1},\ \text{C}) \wedge (\neg\ \text{ilessp}(\text{A},\ x)))$
$\rightarrow\ (\text{iter-h}(\text{iplus}(\mathtt{-1},\ i),\ \text{iplus}(x,\ \text{ineg}(\text{B})))$
$=\ \text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{iplus}(\text{ineg}(\text{B}),\ \text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))))$

THEOREM: iterh_i-1_x-b=iterh_i-1+n_c-1_x+nd-b
$(\text{ilessp}(\mathtt{1},\ i) \wedge \text{ilessp}(\mathtt{1},\ \text{C}) \wedge (\neg\ \text{ilessp}(\text{A},\ x)))$
$\rightarrow\ (\text{iter-h}(\text{iplus}(\mathtt{-1},\ i),\ \text{iplus}(x,\ \text{ineg}(\text{B})))$
$=\ \text{iter-h}(\text{iplus}(\mathtt{-1},\ \text{iplus}(i,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{iplus}(\text{C},\ \mathtt{-1})))),$
$\text{iplus}(\text{ineg}(\text{B}),\ \text{iplus}(x,\ \text{itimes}(\text{n}(\text{A},\ \text{D},\ x),\ \text{D})))))$

THEOREM: iterh_i_x=iterh_i-1_i-b-when-a>=x
(ilessp (1, i) ∧ ilessp (1, C) ∧ (¬ ilessp (A, x)))
→ (iter-h (i, x) = iter-h (iplus (-1, i), iplus (x, ineg (B))))

THEOREM: iterh_i_x=iterh_i-1_i-b
(ilessp (1, i) ∧ ilessp (1, C))
→ (iter-h (i, x) = iter-h (iplus (-1, i), iplus (x, ineg (B))))

THEOREM: x*1=x-when-0<x
ilessp (0, x) → (itimes (x, 1) = x)

THEOREM: iterh_i_x=iterh_i-j_i-jb
(ilessp (1, i) ∧ ilessp (1, C) ∧ ilessp (0, j) ∧ ilessp (j, i))
→ (iter-h (i, x) = iter-h (iplus (i, ineg (j)), iplus (x, ineg (itimes (B, j)))))

THEOREM: c-1>0-when-c>1
ilessp (1, c) → ilessp (0, iplus (c, -1))

THEOREM: c-1<c
ilessp (iplus (c, -1), c)

THEOREM: iterh_c_x=h_x-_c-1_b
ilessp (1, C) → (iter-h (C, x) = h (iplus (x, ineg (itimes (B, iplus (C, -1))))))

THEOREM: h_x=h_x+d-_c-1_b
(ilessp (1, C) ∧ (¬ ilessp (A, x)))
→ (h (x) = h (iplus (x, iplus (D, ineg (itimes (B, iplus (C, -1)))))))

THEOREM: h-satisfies-simple-rec-when-c>1
ilessp (1, C)
→ (h (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
else h (iplus (x, iplus (D, ineg (itimes (B, iplus (C, -1)))))) **endif**)

THEOREM: h-satisfies-simple-rec
h (x)
= **if** ilessp (A, x) **then** iplus (x, ineg (B))
else h (iplus (x, iplus (D, ineg (itimes (B, iplus (C, -1)))))) **endif**

THEOREM: h=k-when-b_c-1_<d
ilessp (itimes (B, iplus (C, -1)), D) → (h (x) = k (x))

```
;;;;;;;;;;MAKE-LIB IS HERE;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;I am here;;;;;;;;;;;;
; (MAKE-LIB "knuth-91a")
; (MAKE-LIB-CONDITIONAL "knuth-91a" T T)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Index

x*1=x-when-0<x, 16
x+2-1=1+x, 15
x+_n-1_r<x+nr-when-0<r, 4
x+y>=x+-z+y-when-z>=0, 5
x-y<=0-when-x<=y, 9
x-y<x-when-0<y, 9
x-y>0-when-y<x, 10
x=1-or-x>1-when-x>0, 6