

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; In an unpublished paper, Textbook Examples of Recursion, Donald E.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 ).

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

; The first problem to solve is: How can Knuth's problem be stated so
; the theorem prover can work on it?

```

```

; Define two mutually recursive ( partial ) functions:

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

;   IterateK( a,b,c,d,e,x ) <== if  e <= 1
;                               then  K( a,b,c,d,x )
;                               else  K( a,b,c,d,
;                                       IterateK( a,b,c,d,e-1,x))

; Knuth's parameters a, b, c, and d are included in the formal
; parameters of both K and IterateK because the theorem prover does
; not allow functions with definitions which contain "global"
; variables.

; Intuitively, IterateK iterates K e times, that is, K is applied e
; times.

; When the specified number, e, of times K is be iterated is is not
; positive, K is iterated one time. That is, when e<1 the result in
; IterateK is the same as if e were 1.

; Thus K( a,b,c,d,x ) = IterateK( a,b,c,d,1,x ).

; Since the theorem prover does not deal with reals, the parameters
; a,b,c, and d are assumed to be integers.

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

; To avoid the complications of dealing with mutually recursive
; partial functions, a suggestion of M. Kaufmann is followed:

; In the definition of IterateK, replace occurrences of K with the
; body of K.

; Define a recursive partial function by

;   IterK( a,b,c,d,e,x ) <== if  1 < e
;                               then  IterK( a,b,c,d,1,
;                                       IterK( a,b,c,d,e-1,x ))
;                               else if  a < x
;                                       then  x - b
;                                       else  IterK( a,b,c,d,c,x+d).

```

```

; Then the partial function K can be defined by

;   K( a,b,c,d,x ) <== IterK( a,b,c,d,1,x ).

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

; K( a,b,c,d,x ) is said to exist just in case the tuple of integers
; ( a,b,c,d,x ) is in the domain of those input values where the
; partial function K halts and produces output values.

; Knuth's theorem follows from parts 1, 4, 6, 8, and 9 of the
; following Main Theorem.

; 1. If  $x > a$ , then  $K( a,b,c,d,x )$  exists.

; 2. If  $x \leq a$ ,  $d \leq 0$ , and  $c \leq 1$ , then  $K( a,b,c,d,x )$  does not
; exist.

; 3. If  $x \leq a$ ,  $d \leq 0$ , and  $c > 1$ , then  $K( a,b,c,d,x )$  does not
; exist.

; 4. If  $x \leq a$ ,  $d > 0$ , and  $c \leq 1$ , then  $K( a,b,c,d,x )$  exists.

; 5. If  $x \leq a$ ,  $d > 0$ ,  $c > 1$ , and  $b \leq 0$ , then  $K( a,b,c,d,x )$ 
; exists.

; 6. If  $x \leq a$ ,  $d > 0$ ,  $c > 1$ , and  $b > 0$ , then  $K( a,b,c,d,x )$ 
; exists if and only if  $K( a,b,c,d,x+d-(c-1)b )$  exists and
;  $K( a,b,c,d,x ) = K( a,b,c,d,x+d-(c-1)b )$ .

; 7. If  $d > 0$ ,  $c > 1$ ,  $b > 0$ , and  $(c-1)b < d$ , then
;  $K( a,b,c,d,x )$  exists if and only if  $K( a,b,1,d-(c-1)b,x )$ 
; exists and  $K( a,b,c,d,x ) = K( a,b,1,d-(c-1)b,x )$ .

; 8. If  $d > 0$ ,  $c > 1$ ,  $b > 0$ , and  $(c-1)b < d$ , then
;  $K( a,b,c,d,x )$  exists.

; 9. If  $x \leq a$ ,  $d > 0$ ,  $c > 1$ ,  $b > 0$ , and  $(c-1)b \geq d$ , then
;  $K( a,b,c,d,x )$  does not exist

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

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

```

; Use the library of integer facts.

EVENT: Start with the library "integers".

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define the partial function IterK using EVAL$,
; and define the partial function K using IterK.
```

DEFINITION:

```
iterk(a, b, c, d, e, x)
= eval$(t,
  'if
    (ilessp '1 e)
      (iterk a b c d
        '1
        (iterk a b c d (idifference e '1) x))
    (if
      (ilessp a x)
        (idifference x b)
        (iterk a b c d c (iplus x d))),
  list (cons ('a, a),
        cons ('b, b),
        cons ('c, c),
        cons ('d, d),
        cons ('e, e),
        cons ('x, x)))
```

DEFINITION: $k(a, b, c, d, x) = \text{iterk}(a, b, c, d, 1, x)$

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Recursively distribute V&C-APPLY$ throughout the body of the
; defintion of ITERK. Break the definition into several cases.
```

THEOREM: iterk-exists-iff-body-exists

```
v&c-apply$ ('iterk,
  list (cons (a, 0),
            cons (b, 0),
            cons (c, 0),
            cons (d, 0),
            cons (e, 0),
            cons (x, 0)))
```

```

↔ v&c-apply$('if,
      list(v&c-apply$('ilessp, list('(1 . 0), cons(e, 0))),
          v&c-apply$('iterk,
                    list(cons(a, 0),
                          cons(b, 0),
                          cons(c, 0),
                          cons(d, 0),
                          '(1 . 0),
                          v&c-apply$('iterk,
                                      list(cons(a, 0),
                                            cons(b, 0),
                                            cons(c, 0),
                                            cons(d, 0),
                                            v&c-apply$('idifference,
                                                        cons(cons(e,
                                                                0),
                                                                '(1
                                                                . 0))),
                                            cons(x, 0))))),
          v&c-apply$('if,
                    list(v&c-apply$('ilessp,
                                      list(cons(a, 0), cons(x, 0))),
                          v&c-apply$('idifference,
                                      list(cons(x, 0), cons(b, 0))),
                          v&c-apply$('iterk,
                                      list(cons(a, 0),
                                            cons(b, 0),
                                            cons(c, 0),
                                            cons(d, 0),
                                            cons(c, 0),
                                            v&c-apply$('iplus,
                                                        list(cons(x,
                                                                0),
                                                                cons(d,
                                                                0))))))))))

```

THEOREM: iterk-value=body-value

```

car(v&c-apply$('iterk,
  list(cons(a, 0),
        cons(b, 0),
        cons(c, 0),
        cons(d, 0),
        cons(e, 0),
        cons(x, 0))))

```

```

= car (v&c-apply$ ('if,
               list (v&c-apply$ ('ilessp, list ('(1 . 0), cons (e, 0))),
                   v&c-apply$ ('iterk,
                               list (cons (a, 0),
                                       cons (b, 0),
                                       cons (c, 0),
                                       cons (d, 0),
                                       '(1 . 0),
                                       v&c-apply$ ('iterk,
                                                   list (cons (a, 0),
                                                           cons (b, 0),
                                                           cons (c, 0),
                                                           cons (d, 0),
                                                           v&c-apply$ ('idifference,
                                                                     cons (cons (e,
                                                                    0),
                                                                    '( (1
                                                                    . 0))))),
                                                           cons (x, 0))))),
                   v&c-apply$ ('if,
                               list (v&c-apply$ ('ilessp,
                                                 list (cons (a, 0),
                                                         cons (x, 0))),
                                       v&c-apply$ ('idifference,
                                                 list (cons (x, 0),
                                                         cons (b, 0))),
                                       v&c-apply$ ('iterk,
                                                 list (cons (a, 0),
                                                         cons (b, 0),
                                                         cons (c, 0),
                                                         cons (d, 0),
                                                         cons (c, 0),
                                                         v&c-apply$ ('iplus,
                                                                     list (cons (x,
                                                                    0),
                                                                    cons (d,
                                                                    0))))))))))

```

THEOREM: cost-is-a-numberp
 $\text{cdr}(\text{v\&c-apply}\$ (fn, args)) \in \mathbf{N}$

THEOREM: cost>0-if-fn-exists
 $\text{v\&c-apply}\$ (fn, args) \rightarrow (0 < \text{cdr}(\text{v\&c-apply}\$ (fn, args)))$

THEOREM: iterk-cost>body-cost

```

v&c-apply$ ('iterk,
  list (cons (a, 0),
        cons (b, 0),
        cons (c, 0),
        cons (d, 0),
        cons (e, 0),
        cons (x, 0)))
→ (cdr (v&c-apply$ ('if,
  list (v&c-apply$ ('ilessp, list ('(1 . 0), cons (e, 0))),
        v&c-apply$ ('iterk,
          list (cons (a, 0),
                cons (b, 0),
                cons (c, 0),
                cons (d, 0),
                '(1 . 0),
                v&c-apply$ ('iterk,
                  list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        v&c-apply$ ('idifference,
                          cons (cons (e,
                                      0),
                                      '((1
                                          . 0))))),
                          cons (x, 0))))),
        v&c-apply$ ('if,
          list (v&c-apply$ ('ilessp,
                        list (cons (a, 0),
                              cons (x, 0))),
                v&c-apply$ ('idifference,
                  list (cons (x, 0),
                        cons (b, 0))),
                v&c-apply$ ('iterk,
                  list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (c, 0),
                        v&c-apply$ ('iplus,
                          list (cons (x,
                                      0),
                                      cons (d,
                                          0)))))))))))))

```

```

< cdr (v&c-apply$ ('iterk,
                list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (e, 0),
                        cons (x, 0))))))

```

THEOREM: v&c-apply\$-if

```

v&c-apply$ ('if, args)
=  if car (args)
    then if caar (args) then fix-cost (cadr (args), 1 + cdar (args))
        else fix-cost (caddr (args), 1 + cdar (args)) endif
    else f endif

```

THEOREM: iterk-exists-iff-body-exists-when-e>1

```

ilessp (1, e)
→ (v&c-apply$ ('iterk,
                list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (e, 0),
                        cons (x, 0))))
↔ v&c-apply$ ('iterk,
                list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        '(1 . 0),
                        v&c-apply$ ('iterk,
                                    list (cons (a, 0),
                                            cons (b, 0),
                                            cons (c, 0),
                                            cons (d, 0),
                                            v&c-apply$ ('idifference,
                                                        cons (cons (e, 0),
                                                                '((1 . 0))))),
                                    cons (x, 0))))))

```

THEOREM: iterk-value=body-value-when-e>1

```

ilessp (1, e)
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),

```



```

cons(c, 0),
cons(d, 0),
cons(e, 0),
cons(x, 0)))
= car(v&c-apply$('iterk,
list(cons(a, 0),
cons(b, 0),
cons(c, 0),
cons(d, 0),
'(1 . 0),
v&c-apply$('iterk,
list(cons(a, 0),
cons(b, 0),
cons(c, 0),
cons(d, 0),
v&c-apply$('idifference,
cons(cons(e, 0),
'(1
. 0))),
cons(x, 0))))))

```

THEOREM: iterk-cost > body-cost-when-e > 1

```

(ilessp(1, e)
^ v&c-apply$('iterk,
list(cons(a, 0),
cons(b, 0),
cons(c, 0),
cons(d, 0),
cons(e, 0),
cons(x, 0)))
→ (cdr(v&c-apply$('iterk,
list(cons(a, 0),
cons(b, 0),
cons(c, 0),
cons(d, 0),
'(1 . 0),
v&c-apply$('iterk,
list(cons(a, 0),
cons(b, 0),
cons(c, 0),
cons(d, 0),
v&c-apply$('idifference,
cons(cons(e, 0),
'((1 . 0))),

```

```

                                cons(x, 0))))))
< cdr (v&c-apply$('iterk,
                                list(cons(a, 0),
                                       cons(b, 0),
                                       cons(c, 0),
                                       cons(d, 0),
                                       cons(e, 0),
                                       cons(x, 0))))))

```

THEOREM: eq-args-give-eq-existence

```

((fn ≠ 'quote)
 ∧ (fn ≠ 'if)
 ∧ (strip-cars(args1) = strip-cars(args2))
 ∧ ((f ∈ args1) = (f ∈ args2)))
→ (v&c-apply$(fn, args1) ↔ v&c-apply$(fn, args2))

```

THEOREM: eq-args-give-eq-values

```

((fn ≠ 'quote)
 ∧ (fn ≠ 'if)
 ∧ (strip-cars(args1) = strip-cars(args2))
 ∧ ((f ∈ args1) = (f ∈ args2)))
→ (car(v&c-apply$(fn, args1)) = car(v&c-apply$(fn, args2)))

```

THEOREM: eq-args-cost-depends-on-cost-of-args

```

((fn ≠ 'if)
 ∧ (sum-cdrs(args1) < sum-cdrs(args2))
 ∧ (strip-cars(args1) = strip-cars(args2))
 ∧ v&c-apply$(fn, args1)
 ∧ v&c-apply$(fn, args2))
→ (cdr(v&c-apply$(fn, args1)) < cdr(v&c-apply$(fn, args2)))

```

THEOREM: iterk-v&c-apply\$-idifference-exists

```

v&c-apply$('iterk,
           list(cons(a, 0),
                 cons(b, 0),
                 cons(c, 0),
                 cons(d, 0),
                 v&c-apply$('idifference, cons(cons(e, 0), '((1 . 0))),
                 cons(x, 0)))
↔ v&c-apply$('iterk,
           list(cons(a, 0),
                 cons(b, 0),
                 cons(c, 0),
                 cons(d, 0),
                 cons(idifference(e, 1), 0),
                 cons(x, 0)))

```

THEOREM: iterk-v&c-apply\$idifference-value

```

car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        v&c-apply$ ('idifference,
                                    cons (cons (e, 0), '(1 . 0))),
                        cons (x, 0))))
=   car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (idifference (e, 1), 0),
                            cons (x, 0))))

```

THEOREM: iterk-v&c-apply\$idifference-cost

```

v&c-apply$ ('iterk,
            list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (idifference (e, 1), 0),
                    cons (x, 0)))
→  (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (idifference (e, 1), 0),
                            cons (x, 0))))
    <  cdr (v&c-apply$ ('iterk,
                        list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                v&c-apply$ ('idifference,
                                            cons (cons (e, 0), '(1 . 0))),
                                cons (x, 0))))

```

THEOREM: iterk-exists-iff-body-exists-when-e>1-version-2

```

ilessp (1, e)
→  (v&c-apply$ ('iterk,

```

```

list (cons (a, 0),
      cons (b, 0),
      cons (c, 0),
      cons (d, 0),
      cons (e, 0),
      cons (x, 0)))
↔ v&c-apply$ ('iterk,
              list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    '(1 . 0),
                    v&c-apply$ ('iterk,
                                list (cons (a, 0),
                                      cons (b, 0),
                                      cons (c, 0),
                                      cons (d, 0),
                                      cons (idifference (e, 1), 0),
                                      cons (x, 0))))))

```

THEOREM: iterk-value=body-value-when-e>1-version-2

```

ilessp(1, e)
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (e, 0),
                          cons (x, 0))))))
= car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          '(1 . 0),
                          v&c-apply$ ('iterk,
                                      list (cons (a, 0),
                                                cons (b, 0),
                                                cons (c, 0),
                                                cons (d, 0),
                                                cons (idifference (e, 1), 0),
                                                cons (x, 0))))))

```

THEOREM: args-exist-when-fn-exists

$((fn \neq 'if) \wedge v\&c\text{-apply}\$(fn, args)) \rightarrow (f \notin args)$

THEOREM: iterk-cost > body-cost-when-e > 1-version-2

```

(ilessp (1, e)
  ^ v&c-apply$ ('iterk,
    list (cons (a, 0),
          cons (b, 0),
          cons (c, 0),
          cons (d, 0),
          cons (e, 0),
          cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
  list (cons (a, 0),
        cons (b, 0),
        cons (c, 0),
        cons (d, 0),
        '(1 . 0),
        v&c-apply$ ('iterk,
          list (cons (a, 0),
                cons (b, 0),
                cons (c, 0),
                cons (d, 0),
                cons (idifference (e, 1), 0),
                cons (x, 0))))))
  < cdr (v&c-apply$ ('iterk,
    list (cons (a, 0),
          cons (b, 0),
          cons (c, 0),
          cons (d, 0),
          cons (e, 0),
          cons (x, 0))))))

```

THEOREM: iterk-exists-iff-body-exists-when-e > 1-version-3-case-1

```

(vc-x ^ illessp (1, e))
→ (v&c-apply$ ('iterk,
  list (cons (a, 0),
        cons (b, 0),
        cons (c, 0),
        cons (d, 0),
        cons (e, 0),
        vc-x))
  ↔ v&c-apply$ ('iterk,
    list (cons (a, 0),
          cons (b, 0),
          cons (c, 0),
          cons (d, 0),

```

```

'(1 . 0),
v&c-apply$ ('iterk,
             list (cons (a, 0),
                   cons (b, 0),
                   cons (c, 0),
                   cons (d, 0),
                   cons (idifference (e, 1), 0),
                   vc-x))))

```

THEOREM: iterk-exists-iff-body-exists-when-e>1-version-3-case-2

$((vc-x = \mathbf{f}) \wedge \text{ilessp}(1, e))$

```

→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     cons (e, 0),
                     vc-x))
   ↔ v&c-apply$ ('iterk,
                 list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       '(1 . 0),
                       v&c-apply$ ('iterk,
                                   list (cons (a, 0),
                                           cons (b, 0),
                                           cons (c, 0),
                                           cons (d, 0),
                                           cons (idifference (e, 1), 0),
                                           vc-x))))))

```

THEOREM: iterk-exists-iff-body-exists-when-e>1-version-3

$\text{ilessp}(1, e)$

```

→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     cons (e, 0),
                     vc-x))
   ↔ v&c-apply$ ('iterk,
                 list (cons (a, 0),
                       cons (b, 0),

```

```

cons (c, 0),
cons (d, 0),
'(1 . 0),
v&c-apply$ ('iterk,
             list (cons (a, 0),
                   cons (b, 0),
                   cons (c, 0),
                   cons (d, 0),
                   cons (idifference (e, 1), 0),
                   vc-x))))

```

THEOREM: iterk-value=body-value-when-e>1-version-3-case-1

$(vc-x \wedge \text{ilessp}(1, e))$

```

→ (car (v&c-apply$ ('iterk,
                   list (cons (a, 0),
                           cons (b, 0),
                           cons (c, 0),
                           cons (d, 0),
                           cons (e, 0),
                           vc-x)))
    = car (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                               cons (b, 0),
                               cons (c, 0),
                               cons (d, 0),
                               '(1 . 0),
                               v&c-apply$ ('iterk,
                                             list (cons (a, 0),
                                                     cons (b, 0),
                                                     cons (c, 0),
                                                     cons (d, 0),
                                                     cons (idifference (e, 1), 0),
                                                     vc-x))))))

```

THEOREM: iterk-value=body-value-when-e>1-version-3-case-2

$((vc-x = \mathbf{f}) \wedge \text{ilessp}(1, e))$

```

→ (car (v&c-apply$ ('iterk,
                   list (cons (a, 0),
                           cons (b, 0),
                           cons (c, 0),
                           cons (d, 0),
                           cons (e, 0),
                           vc-x)))
    = car (v&c-apply$ ('iterk,

```

```

list (cons (a, 0),
      cons (b, 0),
      cons (c, 0),
      cons (d, 0),
      '(1 . 0),
      v&c-apply$ ('iterk,
                  list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (idifference (e, 1), 0),
                        vc-x))))))

```

THEOREM: iterk-value=body-value-when-e>1-version-3

```

ilessp (1, e)
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (e, 0),
                          vc-x)))
     = car (v&c-apply$ ('iterk,
                        list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              v&c-apply$ ('iterk,
                                              list (cons (a, 0),
                                                    cons (b, 0),
                                                    cons (c, 0),
                                                    cons (d, 0),
                                                    cons (idifference (e, 1), 0),
                                                    vc-x))))))

```

THEOREM: iterk-exists-when-e<=1&a<x

```

((¬ illessp (1, e)) ∧ illessp (a, x))
→ v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (e, 0),
                       cons (x, 0)))

```


THEOREM: iterk-value-when-e<=1&a<x

$$\begin{aligned}
 & ((\neg \text{ilessp}(1, e)) \wedge \text{ilessp}(a, x)) \\
 & \rightarrow (\text{car}(\text{v\&c-apply}\$('iterk, \\
 & \quad \text{list}(\text{cons}(a, 0), \\
 & \quad \quad \text{cons}(b, 0), \\
 & \quad \quad \text{cons}(c, 0), \\
 & \quad \quad \text{cons}(d, 0), \\
 & \quad \quad \text{cons}(e, 0), \\
 & \quad \quad \text{cons}(x, 0)))) \\
 & = \text{idifference}(x, b)
 \end{aligned}$$

THEOREM: iterk-exists-iff-body-exists-when-e<=1&a>=x

$$\begin{aligned}
 & ((\neg \text{ilessp}(1, e)) \wedge (\neg \text{ilessp}(a, x))) \\
 & \rightarrow (\text{v\&c-apply}\$('iterk, \\
 & \quad \text{list}(\text{cons}(a, 0), \\
 & \quad \quad \text{cons}(b, 0), \\
 & \quad \quad \text{cons}(c, 0), \\
 & \quad \quad \text{cons}(d, 0), \\
 & \quad \quad \text{cons}(e, 0), \\
 & \quad \quad \text{cons}(x, 0))) \\
 & \leftrightarrow \text{v\&c-apply}\$('iterk, \\
 & \quad \text{list}(\text{cons}(a, 0), \\
 & \quad \quad \text{cons}(b, 0), \\
 & \quad \quad \text{cons}(c, 0), \\
 & \quad \quad \text{cons}(d, 0), \\
 & \quad \quad \text{cons}(c, 0), \\
 & \quad \quad \text{v\&c-apply}\$('iplus, \\
 & \quad \quad \quad \text{list}(\text{cons}(x, 0), \text{cons}(d, 0))))))
 \end{aligned}$$

THEOREM: iterk-value=body-value-when-e<=1&a>=x

$$\begin{aligned}
 & ((\neg \text{ilessp}(1, e)) \wedge (\neg \text{ilessp}(a, x))) \\
 & \rightarrow (\text{car}(\text{v\&c-apply}\$('iterk, \\
 & \quad \text{list}(\text{cons}(a, 0), \\
 & \quad \quad \text{cons}(b, 0), \\
 & \quad \quad \text{cons}(c, 0), \\
 & \quad \quad \text{cons}(d, 0), \\
 & \quad \quad \text{cons}(e, 0), \\
 & \quad \quad \text{cons}(x, 0)))) \\
 & = \text{car}(\text{v\&c-apply}\$('iterk, \\
 & \quad \text{list}(\text{cons}(a, 0), \\
 & \quad \quad \text{cons}(b, 0), \\
 & \quad \quad \text{cons}(c, 0), \\
 & \quad \quad \text{cons}(d, 0), \\
 & \quad \quad \text{cons}(c, 0),
 \end{aligned}$$

$$\text{v\&c-apply}\$ ('iplus, \\ \text{list} (\text{cons} (x, 0), \text{cons} (d, 0))))))$$

THEOREM: iterk-cost>body-cost-when-e<=1&a>=x

$$\begin{aligned} & ((\neg \text{ilessp} (1, e)) \\ & \wedge (\neg \text{ilessp} (a, x)) \\ & \wedge \text{v\&c-apply}\$ ('iterk, \\ & \quad \text{list} (\text{cons} (a, 0), \\ & \quad \quad \text{cons} (b, 0), \\ & \quad \quad \text{cons} (c, 0), \\ & \quad \quad \text{cons} (d, 0), \\ & \quad \quad \text{cons} (e, 0), \\ & \quad \quad \text{cons} (x, 0)))) \\ \rightarrow & (\text{cdr} (\text{v\&c-apply}\$ ('iterk, \\ & \quad \text{list} (\text{cons} (a, 0), \\ & \quad \quad \text{cons} (b, 0), \\ & \quad \quad \text{cons} (c, 0), \\ & \quad \quad \text{cons} (d, 0), \\ & \quad \quad \text{cons} (c, 0), \\ & \quad \quad \text{v\&c-apply}\$ ('iplus, \text{list} (\text{cons} (x, 0), \text{cons} (d, 0)))))) \\ & < \text{cdr} (\text{v\&c-apply}\$ ('iterk, \\ & \quad \text{list} (\text{cons} (a, 0), \\ & \quad \quad \text{cons} (b, 0), \\ & \quad \quad \text{cons} (c, 0), \\ & \quad \quad \text{cons} (d, 0), \\ & \quad \quad \text{cons} (e, 0), \\ & \quad \quad \text{cons} (x, 0)))))) \end{aligned}$$

EVENT: Disable iterk-exists-iff-body-exists.

EVENT: Disable iterk-value=body-value.

EVENT: Disable iterk-cost>body-cost.

THEOREM: iterk-exists-iff-body-exists-when-e<=1&a>=x-version-2

$$\begin{aligned} & ((\neg \text{ilessp} (1, e)) \wedge (\neg \text{ilessp} (a, x))) \\ \rightarrow & (\text{v\&c-apply}\$ ('iterk, \\ & \quad \text{list} (\text{cons} (a, 0), \\ & \quad \quad \text{cons} (b, 0), \\ & \quad \quad \text{cons} (c, 0), \\ & \quad \quad \text{cons} (d, 0), \\ & \quad \quad \text{cons} (e, 0), \\ & \quad \quad \text{cons} (x, 0))) \end{aligned}$$

$$\leftrightarrow \text{v\&c-apply}\$ ('iterk, \\ \text{list (cons (a, 0),} \\ \text{cons (b, 0),} \\ \text{cons (c, 0),} \\ \text{cons (d, 0),} \\ \text{cons (c, 0),} \\ \text{cons (iplus (x, d), 0))})$$

THEOREM: iterk-value=body-value-when-e<=1&a>=x-version-2

$$\begin{aligned} & ((\neg \text{ilessp (1, e)}) \wedge (\neg \text{ilessp (a, x)})) \\ \rightarrow & \text{ (car (v\&c-apply}\$ ('iterk, \\ & \text{list (cons (a, 0),} \\ & \text{cons (b, 0),} \\ & \text{cons (c, 0),} \\ & \text{cons (d, 0),} \\ & \text{cons (e, 0),} \\ & \text{cons (x, 0))})} \\ = & \text{ car (v\&c-apply}\$ ('iterk, \\ & \text{list (cons (a, 0),} \\ & \text{cons (b, 0),} \\ & \text{cons (c, 0),} \\ & \text{cons (d, 0),} \\ & \text{cons (c, 0),} \\ & \text{cons (iplus (x, d), 0))})} \end{aligned}$$

THEOREM: iterk-cost>body-cost-when-e<=1&a>=x-version-2

$$\begin{aligned} & ((\neg \text{ilessp (1, e)}) \\ & \wedge (\neg \text{ilessp (a, x)}) \\ & \wedge \text{v\&c-apply}\$ ('iterk, \\ & \text{list (cons (a, 0),} \\ & \text{cons (b, 0),} \\ & \text{cons (c, 0),} \\ & \text{cons (d, 0),} \\ & \text{cons (e, 0),} \\ & \text{cons (x, 0))}) \\ \rightarrow & \text{ (cdr (v\&c-apply}\$ ('iterk, \\ & \text{list (cons (a, 0),} \\ & \text{cons (b, 0),} \\ & \text{cons (c, 0),} \\ & \text{cons (d, 0),} \\ & \text{cons (c, 0),} \\ & \text{cons (iplus (x, d), 0))})} \\ < & \text{ cdr (v\&c-apply}\$ ('iterk, \\ & \text{list (cons (a, 0),} \end{aligned}$$

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0))))))

```

EVENT: Disable iterk-exists-iff-body-exists-when-e>1.

EVENT: Disable iterk-value=body-value-when-e>1.

EVENT: Disable iterk-cost>body-cost-when-e>1.

EVENT: Disable iterk-exists-iff-body-exists-when-e<=1&a>=x.

EVENT: Disable iterk-value=body-value-when-e<=1&a>=x.

EVENT: Disable iterk-cost>body-cost-when-e<=1&a>=x.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 1 of the
; Main Theorem given above in the introduction.

```

```

; 1. If  $x > a$ , then  $K(a, b, c, d, x)$  exists.

```

```

; Proof. By the definitions of IterK and K.

```

THEOREM: k-exists-when-x>a

```

ilessp(a, x)
→ v&c-apply$('k,
             list(cons(a, 0), cons(b, 0), cons(c, 0), cons(d, 0), cons(x, 0)))

```

THEOREM: k-halts-when-x>a

```

(vc-a ∧ vc-b ∧ vc-c ∧ vc-d ∧ vc-x ∧ illessp(car(vc-a), car(vc-x)))
→ v&c-apply$('k, list(vc-a, vc-b, vc-c, vc-d, vc-x))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 1. The cost of computing IterK is unbounded
; when  $x \leq a$ ,  $d \leq 0$ , and  $c \leq 1$ :

```

```

; Assume  $x \leq a$ ,  $d \leq 0$ ,  $c \leq 1$ ,  $i > 0$ ,
; and

```

```

; IterK( a,b,c,d,1,x ) exists.

; Then IterK( a,b,c,d,c,x+id ) exists

; and

; cost[ IterK( a,b,c,d,c,x+id ) ] + i-1
; <
; cost[ IterK( a,b,c,d,1,x ) ].

; Proof. By induction on i.

```

THEOREM: $\text{count-sub1-x} < \text{count-x-when-x} > 0 \& x < > 1$
 $(\text{ilessp}(0, x) \wedge (x \neq 1)) \rightarrow (\text{count}(\text{iplus}(-1, x)) < \text{count}(x))$

DEFINITION:

```

induct-hint-positive-int (x)
= if  $\neg \text{ilessp}(0, x)$  then t
  elseif  $x = 1$  then t
  else induct-hint-positive-int (iplus(-1, x)) endif

```

THEOREM: $\text{sub1-x} > 0\text{-when-x} > 0 \& x < > 1$
 $(\text{ilessp}(0, x) \wedge (x \neq 1)) \rightarrow \text{ilessp}(0, \text{iplus}(-1, x))$

THEOREM: $x + i - 1.d \leq a\text{-when-} x \leq a \& d \leq 0 \& i > 0$
 $((\neg \text{ilessp}(a, x)) \wedge (\neg \text{ilessp}(0, d)) \wedge \text{ilessp}(0, i))$
 $\rightarrow (\neg \text{ilessp}(a, \text{iplus}(x, \text{itimes}(\text{iplus}(-1, i), d))))$

THEOREM: $\text{iterk}_x + i - 1.d$ implies $\text{iterk}_x + id$ -when- $x \leq a \& d \leq 0 \& c \leq 1 \& i > 0$
 $((\neg \text{ilessp}(a, x))$
 $\wedge (\neg \text{ilessp}(0, d))$
 $\wedge (\neg \text{ilessp}(1, c))$
 $\wedge \text{ilessp}(0, i)$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(\text{iplus}(x, \text{itimes}(\text{iplus}(-1, i), d)), 0))))$
 $\rightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$

```

cons (d, 0),
cons (c, 0),
cons (iplus (x, itimes (d, i)), 0)))

```

THEOREM: iterk-e=1&x-implies-iterk-e=c&x+id-when-x<=a&d<=0&c<=1

```

((¬ ilessp (a, x))
 ∧ (¬ ilessp (0, d))
 ∧ (¬ ilessp (1, c))
 ∧ ilessp (0, i)
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     cons (1, 0),
                     cons (x, 0))))
 → v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     cons (c, 0),
                     cons (iplus (x, itimes (i, d)), 0)))

```

THEOREM: w-1+y<z&x<y-implies-w+x<z

```

(ilessp (0, w) ∧ (x < y) ∧ ((iplus (-1, w) + y) < z))
 → (((w + x) < z) = t)

```

THEOREM: cost-induction-step-when-x<=a&d<=0&c<=1

```

((¬ ilessp (a, x))
 ∧ (¬ ilessp (0, d))
 ∧ (¬ ilessp (1, c))
 ∧ ilessp (0, i)
 ∧ (i ≠ 1)
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     ' (1 . 0),
                     cons (x, 0))))
 ∧ ((cdr (v&c-apply$ ('iterk,
                      list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),

```

```

                                cons (d, 0),
                                cons (c, 0),
                                cons (iplus (x, itimes (iplus (-1, i), d)), 0))))
+   iplus (-1, iplus (-1, i))
<   cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              cons (x, 0))))))
→ ((cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              cons (c, 0),
                              cons (iplus (x, itimes (d, i)), 0))))
    +   iplus (-1, i))
   <   cdr (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                                 cons (b, 0),
                                 cons (c, 0),
                                 cons (d, 0),
                                 '(1 . 0),
                                 cons (x, 0))))))

```

THEOREM: cost-base-step-when-x<=a&d<=0&c<=1

```

((¬ ilessp (a, x))
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                         cons (b, 0),
                         cons (c, 0),
                         cons (d, 0),
                         cons (1, 0),
                         cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              cons (c, 0),
                              cons (iplus (d, x), 0))))
   <   cdr (v&c-apply$ ('iterk,

```

```

list (cons (a, 0),
      cons (b, 0),
      cons (c, 0),
      cons (d, 0),
      cons (1, 0),
      cons (x, 0))))

```

THEOREM: iterk-e=1&x-cost>i-1+cost-iterk-e=c&x+id-when-x<=a&d<=0&c<=1

```

((¬ ilessp (a, x))
 ∧ (¬ ilessp (0, d))
 ∧ (¬ ilessp (1, c))
 ∧ ilessp (0, i)
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       '(1 . 0),
                       cons (x, 0))))
→ ((cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (c, 0),
                            cons (iplus (x, itimes (i, d)), 0))))
    + iplus (-1, i))
  < cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            '(1 . 0),
                            cons (x, 0))))))

```

THEOREM: y<=z-when-x+y-1<z&x>=0&y>0

```

((x ∈ ℕ) ∧ ilessp (0, y) ∧ ((x + iplus (-1, y)) < z))
→ ((z < y) = t)

```

THEOREM: iterk-cost-is-unbounded-when-x<=a&d<=0&c<=1

```

((¬ ilessp (a, x))
 ∧ (¬ ilessp (0, d))
 ∧ (¬ ilessp (1, c))
 ∧ ilessp (0, i)
 ∧ v&c-apply$ ('iterk,

```



```

list (cons (a, 0),
      cons (b, 0),
      cons (c, 0),
      cons (d, 0),
      '(1 . 0),
      cons (x, 0)))
→ (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              cons (x, 0))))
    ↯ i)

```

THEOREM: ilessp-0-add1-x
ilessp(0, 1 + x)

THEOREM: iterk-does-not-exist-when-x<=a&d<=0&c<=1
((¬ ilessp(a, x)) ∧ (¬ ilessp(0, d)) ∧ (¬ ilessp(1, c)))
→ (¬ v&c-apply\$ ('iterk,
 list (cons (a, 0),
 cons (b, 0),
 cons (c, 0),
 cons (d, 0),
 '(1 . 0),
 cons (x, 0))))

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 2 of the
; Main Theorem given above in the introduction.

```

```

; 2. If x <= a, d <= 0, and c <= 1, then
; K( a,b,c,d,x ) does not exist.

```

```

; Proof. By Lemma 1 and the definition of K.

```

THEOREM: k-does-not-exist-when-x<=a&d<=0&c<=1
((¬ ilessp(a, x)) ∧ (¬ ilessp(0, d)) ∧ (¬ ilessp(1, c)))
→ (¬ v&c-apply\$ ('k,
 list (cons (a, 0),
 cons (b, 0),
 cons (c, 0),
 cons (d, 0),
 cons (x, 0))))

THEOREM: k-does-not-halt-when- $x \leq a \wedge d \leq 0 \wedge c \leq 1$

$(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge vc-x$
 $\wedge (\neg \text{ilessp}(\text{car}(vc-a), \text{car}(vc-x)))$
 $\wedge (\neg \text{ilessp}(0, \text{car}(vc-d)))$
 $\wedge (\neg \text{ilessp}(1, \text{car}(vc-c)))$
 $\rightarrow (\neg \text{v}\&c\text{-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x)))$

;;;

; Lemma 2. When the number of iterates is positive,
; the cost of computing IterK is an order
; homomorphism of the number of iterates:

; Assume that $0 < e1 < e2$
; and
; IterK(a,b,c,d,e2,x) exists.

; Then IterK(a,b,c,d,e1,x) also exists

; and

; cost[IterK(a,b,c,d,e1,x)]
; <
; cost[IterK(a,b,c,d,e2,x)].

; Proof. Hold e1 constant and induct on e2.

THEOREM: count-sub1-y < count-y-when- $0 < x < y$
 $(\text{ilessp}(0, x) \wedge \text{ilessp}(x, y)) \rightarrow (\text{count}(\text{iplus}(-1, y)) < \text{count}(y))$

DEFINITION:

induct-hint-cost-hom(x, y)
= **if** $\neg \text{ilessp}(0, x)$ **then t**
 elseif $\neg \text{ilessp}(x, y)$ **then t**
 elseif $x = \text{iplus}(-1, y)$ **then t**
 else induct-hint-cost-hom(x, iplus(-1, y)) **endif**

THEOREM: y > 1-when-y > x & x > 0
 $(\text{ilessp}(0, x) \wedge \text{ilessp}(x, y)) \rightarrow \text{ilessp}(1, y)$

THEOREM: iterk-at-e1-exists-when-iterk-at-e2-exists & $0 < e1 < e2$ -basic-step

$$\begin{aligned}
& (\text{ilessp}(0, e1) \\
& \wedge \text{ilessp}(e1, e2) \\
& \wedge \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(e2, 0), \\
& \quad \quad \text{cons}(x, 0))) \\
\rightarrow & \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(-1, e2), 0), \\
& \quad \quad \text{cons}(x, 0)))
\end{aligned}$$

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

THEOREM: iterk-at-e1-exists-when-iterk-at-e2-exists- $0 < e1 < e2$
 $(\text{ilessp}(0, e1)$
 $\wedge \text{ilessp}(e1, e2)$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e2, 0),$
 $\quad \quad \text{cons}(x, 0)))$
 $\rightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e1, 0),$
 $\quad \quad \text{cons}(x, 0)))$

THEOREM: cost > cost-of-args
 $((fn \neq 'if) \wedge (f \notin args) \wedge \text{v\&c-apply}\$(fn, args))$
 $\rightarrow (\text{sum-cdrs}(args) < \text{cdr}(\text{v\&c-apply}\$(fn, args)))$

THEOREM: sum-cdrs >= member
 $(x \in l) \rightarrow (\text{sum-cdrs}(l) \not< \text{cdr}(x))$

THEOREM: cost>member
 $(\text{v\&c-apply}\$(fn, args) \wedge (fn \neq 'if) \wedge (x \in args))$
 $\rightarrow (\text{cdr}(x) < \text{cdr}(\text{v\&c-apply}\$(fn, args)))$

THEOREM: cost-iterk-at-e1<cost-iterk-at-e2-when-0<e1<e2-basic-step
 $(\text{ilessp}(0, e1)$
 $\wedge \text{ilessp}(e1, e2)$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e2, 0),$
 $\quad \quad \text{cons}(x, 0))))$
 $\rightarrow (\text{cdr}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(\text{iplus}(-1, e2), 0),$
 $\quad \quad \text{cons}(x, 0))))$
 $< \text{cdr}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e2, 0),$
 $\quad \quad \text{cons}(x, 0))))$

EVENT: Disable y>1-when-y>x&x>0.

THEOREM: cost-iterk-at-e1<cost-iterk-at-e2-when-0<e1<e2
 $(\text{ilessp}(0, e1)$
 $\wedge \text{ilessp}(e1, e2)$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e2, 0),$
 $\quad \quad \text{cons}(x, 0))))$
 $\rightarrow (\text{cdr}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$

```

                                cons(c, 0),
                                cons(d, 0),
                                cons(e1, 0),
                                cons(x, 0)))
< cdr(v&c-apply$( 'iterk,
                    list(cons(a, 0),
                           cons(b, 0),
                           cons(c, 0),
                           cons(d, 0),
                           cons(e2, 0),
                           cons(x, 0))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 3. The cost of computing IterK is unbounded
;           when x <= a, d <= 0, and c > 1:

; Assume x <= a, d <= 0, c > 1, i > 0,
; and
; IterK( a,b,c,d,1,x ) exists.

; Then IterK( a,b,c,d,1,x+id ) exists

; and

; cost[ IterK( a,b,c,d,1,x+id ) ] + i
; <
; cost[ IterK( a,b,c,d,1,x ) ].

; Proof. By induction on i.

```

THEOREM: iterk_x+i-1.d_implies_iterk_x+id-when-x<=a&d<=0&c>1&i>0&e=1

```

((¬ ilssp(a, x))
 ∧ (¬ ilssp(0, d))
 ∧ ilssp(1, c)
 ∧ ilssp(0, i)
 ∧ v&c-apply$( 'iterk,
                list(cons(a, 0),
                       cons(b, 0),
                       cons(c, 0),
                       cons(d, 0),
                       '(1 . 0),
                       cons(iplus(x, itimes(iplus(-1, i), d)), 0))))
→ v&c-apply$( 'iterk,

```

```

list (cons (a, 0),
      cons (b, 0),
      cons (c, 0),
      cons (d, 0),
      '(1 . 0),
      cons (iplus (x, itimes (d, i)), 0)))

```

THEOREM: iterk-x-implies-iterk-x+id-when-x<=a&d<=0&c>1&e=1

```

((¬ ilessp (a, x))
 ∧ (¬ ilessp (0, d))
 ∧ ilessp (1, c)
 ∧ ilessp (0, i)
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     '(1 . 0),
                     cons (x, 0))))
→ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     '(1 . 0),
                     cons (iplus (x, itimes (i, d)), 0)))

```

THEOREM: iterk-x+d-cost<cost-iterk-x-when-a>=x&1<c

```

((¬ ilessp (a, x))
 ∧ ilessp (1, c)
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     '(1 . 0),
                     cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          '(1 . 0),
                          cons (iplus (x, d), 0))))

```

```

< cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              cons (x, 0))))))

```

THEOREM: cost-induction-step-when-x<=a&d<=0&c>1

```

((¬ ilessp (a, x))
 ∧ (¬ ilessp (0, d))
 ∧ ilessp (1, c)
 ∧ ilessp (0, i)
 ∧ (i ≠ 1)
 ∧ v&c-apply$ ('iterk,
                list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          '(1 . 0),
                          cons (x, 0)))
 ∧ ((cdr (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                                 cons (b, 0),
                                 cons (c, 0),
                                 cons (d, 0),
                                 '(1 . 0),
                                 cons (iplus (x, itimes (iplus (-1, i), d)), 0))))
    + iplus (-1, i))
 < cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              cons (x, 0))))))
 → ((cdr (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                                 cons (b, 0),
                                 cons (c, 0),
                                 cons (d, 0),
                                 '(1 . 0),
                                 cons (iplus (x, itimes (d, i)), 0))))
    + i)

```

```

< cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              cons (x, 0))))))

```

THEOREM: iterk-x-cost>i+cost-iterk-x+id-when-x<=a&d<=0&c>1&e=1

```

((¬ illessp (a, x))
 ∧ (¬ illessp (0, d))
 ∧ illessp (1, c)
 ∧ illessp (0, i)
 ∧ v&c-apply$ ('iterk,
                list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          '(1 . 0),
                          cons (x, 0))))
→ ((cdr (v&c-apply$ ('iterk,
                     list (cons (a, 0),
                               cons (b, 0),
                               cons (c, 0),
                               cons (d, 0),
                               '(1 . 0),
                               cons (iplus (x, itimes (i, d)), 0))))
    + i)
   < cdr (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                                 cons (b, 0),
                                 cons (c, 0),
                                 cons (d, 0),
                                 '(1 . 0),
                                 cons (x, 0))))))

```

THEOREM: iterk-cost-is-unbounded-when-x<=a&d<=0&c>1

```

((¬ illessp (a, x))
 ∧ (¬ illessp (0, d))
 ∧ illessp (1, c)
 ∧ illessp (0, i)
 ∧ v&c-apply$ ('iterk,
                list (cons (a, 0),
                          cons (b, 0),

```



```

cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (x, 0)))
→ (i < cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (x, 0))))))

```

THEOREM: iterk-does-not-exist-when- $x \leq a \wedge d \leq 0 \wedge c > 1$
 $((\neg \text{ilessp}(a, x)) \wedge (\neg \text{ilessp}(0, d)) \wedge \text{ilessp}(1, c))$

```

→ (¬ v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (x, 0))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 3 of the
; Main Theorem given above in the introduction.

```

```

; 3. If  $x \leq a$ ,  $d \leq 0$ , and  $c > 1$ , then
;  $K(a, b, c, d, x)$  does not exist.

```

```

; Proof. By Lemma 3 and the definition of  $K$ .

```

THEOREM: k-does-not-exist-when- $x \leq a \wedge d \leq 0 \wedge c > 1$
 $((\neg \text{ilessp}(a, x)) \wedge (\neg \text{ilessp}(0, d)) \wedge \text{ilessp}(1, c))$

```

→ (¬ v&c-apply$ ('k,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (x, 0))))

```

THEOREM: k-does-not-halt-when- $x \leq a \wedge d \leq 0 \wedge c > 1$

```

(vc-a
∧ vc-b
∧ vc-c

```

```

 $\wedge$   $vc-d$ 
 $\wedge$   $vc-x$ 
 $\wedge$   $(\neg \text{ilessp}(\text{car}(vc-a), \text{car}(vc-x)))$ 
 $\wedge$   $(\neg \text{ilessp}(0, \text{car}(vc-d)))$ 
 $\wedge$   $\text{ilessp}(1, \text{car}(vc-c))$ 
 $\rightarrow$   $(\neg \text{v\&c-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x)))$ 

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 4. IterK exists when  $d > 0$ ,
;            $c \leq 1$ , and  $e = c$ :
;
; Assume  $d > 0$ , and  $c \leq 1$ .
;
; Then  $\text{IterK}(a, b, c, d, c, x)$  exists.
;
; Proof. Hold the parameter  $a$  fixed and
;         induct on the value given by
;
;         if  $x > a$  then 0
;         else  $1 + a - x$ .

```

```

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

```

```

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

```

EVENT: Disable k-measure.

```

DEFINITION:
induct-hint-k-measure( $a, b, c, d, x$ )
= if  $\neg \text{ilessp}(0, d)$  then t
  elseif  $\text{ilessp}(a, x)$  then t
  else  $\text{induct-hint-k-measure}(a, b, c, d, \text{iplus}(d, x))$  endif

```

```

THEOREM:  $\text{iterk}$ -exists-when- $d > 0 \wedge c \leq 1 \wedge e = c$ 
 $(\text{ilessp}(0, d) \wedge (\neg \text{ilessp}(1, c)))$ 
 $\rightarrow$   $\text{v\&c-apply}\$('iterk,$ 
       $\text{list}(\text{cons}(a, 0),$ 
             $\text{cons}(b, 0),$ 

```

```

cons(c, 0),
cons(d, 0),
cons(c, 0),
cons(x, 0))

```

THEOREM: iterk-exists-when- $x \leq a \wedge d > 0 \wedge c \leq 1 \wedge e = 1$

$((\neg \text{ilessp}(a, x)) \wedge \text{ilessp}(0, d) \wedge (\neg \text{ilessp}(1, c)))$

$\rightarrow \text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$
 $\text{cons}(d, 0),$
 $'(1 . 0),$
 $\text{cons}(x, 0))$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 4 of the
; Main Theorem given above in the introduction.

```

```

; 4. If  $x \leq a$ ,  $d > 0$ , and  $c \leq 1$ , then
;  $K(a, b, c, d, x)$  exists.

```

```

; Proof. By Lemma 4 and the definition of K.

```

THEOREM: k-exists-when- $x \leq a \wedge d > 0 \wedge c \leq 1$

$((\neg \text{ilessp}(a, x)) \wedge \text{ilessp}(0, d) \wedge (\neg \text{ilessp}(1, c)))$

$\rightarrow \text{v\&c-apply}\$('k,$
 $\text{list}(\text{cons}(a, 0), \text{cons}(b, 0), \text{cons}(c, 0), \text{cons}(d, 0), \text{cons}(x, 0)))$

THEOREM: k-halts-when- $x \leq a \wedge d > 0 \wedge c \leq 1$

$(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge vc-x$
 $\wedge (\neg \text{ilessp}(\text{car}(vc-a), \text{car}(vc-x)))$
 $\wedge \text{ilessp}(0, \text{car}(vc-d))$
 $\wedge (\neg \text{ilessp}(1, \text{car}(vc-c))))$
 $\rightarrow \text{v\&c-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x))$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 5. IterK exists and has value > a
; when  $x > a$ ,  $b \leq 0$ , and  $e > 0$ :

```

; Assume $x > a$, $b \leq 0$, and $e > 0$.
; Then $\text{IterK}(a, b, c, d, e, x)$ exists
; and
; $\text{IterK}(a, b, c, d, e, x) > a$.
; Proof. By induction on e .

THEOREM: iterk-exists-when-e<=1&a<x-version-2
 $(vc-x \wedge (\neg \text{ilessp}(1, e)) \wedge \text{ilessp}(a, \text{car}(vc-x)))$
 $\rightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e, 0),$
 $\quad \quad vc-x))$

THEOREM: iterk-value-when-e<=1&a<x-version-2
 $(vc-x \wedge (\neg \text{ilessp}(1, e)) \wedge \text{ilessp}(a, \text{car}(vc-x)))$
 $\rightarrow (\text{car}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e, 0),$
 $\quad \quad vc-x)))$
 $= \text{idifference}(\text{car}(vc-x), b))$

THEOREM: a+b<x-when-a<x&b<=0
 $(\text{ilessp}(a, x) \wedge (\neg \text{ilessp}(0, b))) \rightarrow \text{ilessp}(\text{iplus}(a, b), x)$

THEOREM: a<-b+x-when-a<x&b<=0
 $(\text{ilessp}(a, x) \wedge (\neg \text{ilessp}(0, b))) \rightarrow \text{ilessp}(a, \text{iplus}(\text{ineg}(b), x))$

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

THEOREM: iterk-exists&iterk>a-when-x>a&b<=0&e>0
 $(\text{ilessp}(a, x) \wedge (\neg \text{ilessp}(0, b)) \wedge \text{ilessp}(0, e))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0)))
∧ illessp (a,
car (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 6. The number of iterates in the composition
; of IterK with itself add up:

```

```

; Let e1 > 0 and e2 > 0. Then

; IterK( a,b,c,d,e1,IterK( a,b,c,d,e2,x ) ) exists
; iff
; IterK( a,b,c,d,e1+e2,x ) exists

; and

; IterK( a,b,c,d,e1,IterK( a,b,c,d,e2,x ) )
; =
; IterK( a,b,c,d,e1+e2,x )

```

```

; Proof. By induction on e1.

```

THEOREM: $iplus_{-1}^{-1}x=x$.when $x>0$
 $illessp(0, x) \rightarrow (iplus(1, iplus(-1, x)) = x)$

THEOREM: $x+y>1$ -when- $x>0 \& x<<1 \& 0<y$
 $(illessp(0, x) \wedge (x \neq 1) \wedge illessp(0, y)) \rightarrow illessp(1, iplus(x, y))$

THEOREM: nbr-of-iterates-sum-exists-step-1
 $(illessp(0, e1) \wedge (e1 \neq 1))$
 $\rightarrow (v\&c-apply$ ('iterk,$
list (cons (a, 0),
cons (b, 0),
cons (c, 0),

```

cons (d, 0),
cons (e1, 0),
v&c-apply$ ('iterk,
              list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (e2, 0),
                    cons (x, 0))))
↔ v&c-apply$ ('iterk,
              list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    '(1 . 0),
                    v&c-apply$ ('iterk,
                                  list (cons (a, 0),
                                        cons (b, 0),
                                        cons (c, 0),
                                        cons (d, 0),
                                        cons (iplus (-1, e1), 0),
                                        v&c-apply$ ('iterk,
                                                  list (cons (a, 0),
                                                        cons (b, 0),
                                                        cons (c, 0),
                                                        cons (d, 0),
                                                        cons (e2, 0),
                                                        cons (x, 0))))))))))

```

THEOREM: nbr-of-iterates-sum-values-step-1

(ilessp (0, e1) \wedge (e1 \neq 1))

```

→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (e1, 0),
                          v&c-apply$ ('iterk,
                                        list (cons (a, 0),
                                              cons (b, 0),
                                              cons (c, 0),
                                              cons (d, 0),
                                              cons (e2, 0),
                                              cons (x, 0)))))))

```

```

= car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      '(1 . 0),
                      v&c-apply$ ('iterk,
                                  list (cons (a, 0),
                                        cons (b, 0),
                                        cons (c, 0),
                                        cons (d, 0),
                                        cons (iplus (-1, e1), 0),
                                        v&c-apply$ ('iterk,
                                                    list (cons (a, 0),
                                                          cons (b, 0),
                                                          cons (c, 0),
                                                          cons (d, 0),
                                                          cons (e2,
                                                                0),
                                                          cons (x, 0))))))))))

```

THEOREM: nbr-of-iterates-sum-exists-step-2

(ilessp (0, e1) \wedge (e1 \neq 1) \wedge illessp (0, e2))

\rightarrow (v&c-apply\$ ('iterk,
 list (cons (a, 0),
 cons (b, 0),
 cons (c, 0),
 cons (d, 0),
 cons (iplus (e1, e2), 0),
 cons (x, 0)))

\leftrightarrow v&c-apply\$ ('iterk,
 list (cons (a, 0),
 cons (b, 0),
 cons (c, 0),
 cons (d, 0),
 '(1 . 0),
 v&c-apply\$ ('iterk,
 list (cons (a, 0),
 cons (b, 0),
 cons (c, 0),
 cons (d, 0),
 cons (iplus (iplus (-1, e1), e2),
 0),
 cons (x, 0))))))

THEOREM: nbr-of-iterates-sum-values-step-2

```
(ilessp(0, e1) ∧ (e1 ≠ 1) ∧ illessp(0, e2))
→ (car(v&c-apply$('iterk,
    list(cons(a, 0),
        cons(b, 0),
        cons(c, 0),
        cons(d, 0),
        cons(iplus(e1, e2), 0),
        cons(x, 0))))))
= car(v&c-apply$('iterk,
    list(cons(a, 0),
        cons(b, 0),
        cons(c, 0),
        cons(d, 0),
        '(1 . 0),
        v&c-apply$('iterk,
            list(cons(a, 0),
                cons(b, 0),
                cons(c, 0),
                cons(d, 0),
                cons(iplus(iplus(-1, e1),
                    e2),
                    0),
                cons(x, 0)))))))
```

THEOREM: nbr-of-iterates-sum-exists-step-3

```
((v&c-apply$('iterk,
    list(cons(a, 0),
        cons(b, 0),
        cons(c, 0),
        cons(d, 0),
        cons(iplus(-1, e1), 0),
        v&c-apply$('iterk,
            list(cons(a, 0),
                cons(b, 0),
                cons(c, 0),
                cons(d, 0),
                cons(e2, 0),
                cons(x, 0)))))))
↔ v&c-apply$('iterk,
    list(cons(a, 0),
        cons(b, 0),
        cons(c, 0),
        cons(d, 0),
```



```

                                cons (iplus (iplus (-1, e1), e2), 0),
                                cons (x, 0)))
^ (car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (iplus (-1, e1), 0),
                    v&c-apply$ ('iterk,
                                list (cons (a, 0),
                                    cons (b, 0),
                                    cons (c, 0),
                                    cons (d, 0),
                                    cons (e2, 0),
                                    cons (x, 0)))))))
= car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (iplus (iplus (-1, e1), e2), 0),
                    cons (x, 0))))))
→ (v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    ' (1 . 0),
                    v&c-apply$ ('iterk,
                                list (cons (a, 0),
                                    cons (b, 0),
                                    cons (c, 0),
                                    cons (d, 0),
                                    cons (iplus (-1, e1), 0),
                                    v&c-apply$ ('iterk,
                                                list (cons (a, 0),
                                                    cons (b, 0),
                                                    cons (c, 0),
                                                    cons (d, 0),
                                                    cons (e2, 0),
                                                    cons (x, 0)))))))
↔ v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),

```

```

cons (c, 0),
cons (d, 0),
'(1 . 0),
v&c-apply$ ('iterk,
              list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (iplus (-1, iplus (e1, e2)),
                          0),
                    cons (x, 0))))))

```

THEOREM: nbr-of-iterates-sum-values-step-3

```

((v&c-apply$ ('iterk,
              list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (iplus (-1, e1), 0),
                    v&c-apply$ ('iterk,
                                list (cons (a, 0),
                                      cons (b, 0),
                                      cons (c, 0),
                                      cons (d, 0),
                                      cons (e2, 0),
                                      cons (x, 0))))))
  ↔ v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      cons (iplus (iplus (-1, e1), e2), 0),
                      cons (x, 0))))
  ∧ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (iplus (-1, e1), 0),
                          v&c-apply$ ('iterk,
                                      list (cons (a, 0),
                                            cons (b, 0),
                                            cons (c, 0),
                                            cons (d, 0),

```

```

                                cons (e2, 0),
                                cons (x, 0))))))
=  car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              cons (iplus (iplus (-1, e1), e2), 0),
                              cons (x, 0))))))
→  (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              v&c-apply$ ('iterk,
                                          list (cons (a, 0),
                                                    cons (b, 0),
                                                    cons (c, 0),
                                                    cons (d, 0),
                                                    cons (iplus (-1, e1), 0),
                                                    v&c-apply$ ('iterk,
                                                                list (cons (a, 0),
                                                                      cons (b, 0),
                                                                      cons (c, 0),
                                                                      cons (d, 0),
                                                                      cons (e2, 0),
                                                                      cons (x, 0))))))))))
=  car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              '(1 . 0),
                              v&c-apply$ ('iterk,
                                          list (cons (a, 0),
                                                    cons (b, 0),
                                                    cons (c, 0),
                                                    cons (d, 0),
                                                    cons (iplus (-1,
                                                                iplus (e1, e2)),
                                                                0),
                                                    cons (x, 0))))))

```

THEOREM: nbr-of-iterates-sum-exists&values
 (ilessp(0, e1) ∧ illessp(0, e2))
 → ((v&c-apply\$('iterk,
 list(cons(a, 0),
 cons(b, 0),
 cons(c, 0),
 cons(d, 0),
 cons(e1, 0),
 v&c-apply\$('iterk,
 list(cons(a, 0),
 cons(b, 0),
 cons(c, 0),
 cons(d, 0),
 cons(e2, 0),
 cons(x, 0))))))
 ↔ v&c-apply\$('iterk,
 list(cons(a, 0),
 cons(b, 0),
 cons(c, 0),
 cons(d, 0),
 cons(iplus(e1, e2), 0),
 cons(x, 0)))
 ∧ (car(v&c-apply\$('iterk,
 list(cons(a, 0),
 cons(b, 0),
 cons(c, 0),
 cons(d, 0),
 cons(e1, 0),
 v&c-apply\$('iterk,
 list(cons(a, 0),
 cons(b, 0),
 cons(c, 0),
 cons(d, 0),
 cons(e2, 0),
 cons(x, 0))))))
 = car(v&c-apply\$('iterk,
 list(cons(a, 0),
 cons(b, 0),
 cons(c, 0),
 cons(d, 0),
 cons(iplus(e1, e2), 0),
 cons(x, 0))))))

;;

```

; Lemma 7. The number of iterates of K accumulate
;       when 1 < c, 0 < d, and x is "small."

; Assume 1 < c, 0 < d, 0 < n, and a+d >= x+nd.

; Then

; IterK( a,b,c,d,1,x ) exists
; iff
; IterK( a,b,c,d,1+n(c-1),x+nd ) exists

; and

; IterK( a,b,c,d,1,x )
; =
; IterK( a,b,c,d,1+n(c-1),x+nd ).

; Proof. By induction on n.

```

THEOREM: $n \cdot c - 1 + 1 = c$ -when- $n=1 \& c > 1$
 $((n = 1) \wedge \text{ilessp}(1, c)) \rightarrow (\text{iplus}(1, \text{itimes}(n, \text{iplus}(-1, c))) = c)$

THEOREM: $nd = d$ -when- $n=1 \& d > 0$
 $((n = 1) \wedge \text{ilessp}(0, d)) \rightarrow (\text{itimes}(n, d) = d)$

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

THEOREM: $\text{iterk}_e = 1 + n \cdot c - 1 \& x + nd$ -exists-base-step
 $((n = 1)$
 $\wedge \text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(n, d))))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad '(1 . 0),$
 $\quad \quad \text{cons}(x, 0)))$
 $\leftrightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$

$\text{cons}(d, 0),$
 $\text{cons}(\text{iplus}(1, \text{itimes}(n, \text{iplus}(-1, c))), 0),$
 $\text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0))$

THEOREM: iterk_e=1+_n.c-1&x+nd-value-base-step

$((n = 1)$
 $\wedge \text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(n, d))))$
 $\rightarrow (\text{car}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \text{cons}(b, 0),$
 $\quad \text{cons}(c, 0),$
 $\quad \text{cons}(d, 0),$
 $\quad '(1 . 0),$
 $\quad \text{cons}(x, 0))))$
 $= \text{car}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \text{cons}(b, 0),$
 $\quad \text{cons}(c, 0),$
 $\quad \text{cons}(d, 0),$
 $\quad \text{cons}(\text{iplus}(1, \text{itimes}(n, \text{iplus}(-1, c))), 0),$
 $\quad \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0))))$

THEOREM: n-1_c-1>0-when-1<c&0<n&n<>1

$(\text{ilessp}(1, c) \wedge \text{ilessp}(0, n) \wedge (n \neq 1))$
 $\rightarrow \text{ilessp}(0, \text{itimes}(\text{iplus}(-1, n), \text{iplus}(-1, c)))$

THEOREM: iterk_e=1+_n.c-1&x+nd-exists-step-1

$(\text{ilessp}(1, c) \wedge \text{ilessp}(0, n) \wedge (n \neq 1))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \text{cons}(b, 0),$
 $\quad \text{cons}(c, 0),$
 $\quad \text{cons}(d, 0),$
 $\quad \text{cons}(\text{iplus}(1, \text{itimes}(\text{iplus}(-1, n), \text{iplus}(-1, c))), 0),$
 $\quad \text{cons}(\text{iplus}(x, \text{itimes}(\text{iplus}(-1, n), d)), 0)))$
 $\leftrightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \text{cons}(b, 0),$
 $\quad \text{cons}(c, 0),$
 $\quad \text{cons}(d, 0),$
 $\quad \text{cons}(\text{itimes}(\text{iplus}(-1, n), \text{iplus}(-1, c)), 0),$
 $\quad \text{v\&c-apply}\$('iterk,$
 $\quad \quad \text{list}(\text{cons}(a, 0),$

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
' (1 . 0),
cons (iplus (x,
           itimes (iplus (-1, n),
                    d)),
      0))))))

```

THEOREM: iterk_e=1+_n.c-1&x+nd-value-step-1

```

(ilessp (1, c) ^ ilessp (0, n) ^ (n ≠ 1))
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                cons (iplus (1, itimes (iplus (-1, n), iplus (-1, c))),
                                    0),
                                cons (iplus (x, itimes (iplus (-1, n), d)), 0))))
    = car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                cons (itimes (iplus (-1, n), iplus (-1, c)), 0),
                                v&c-apply$ ('iterk,
                                            list (cons (a, 0),
                                                        cons (b, 0),
                                                        cons (c, 0),
                                                        cons (d, 0),
                                                        ' (1 . 0),
                                                        cons (iplus (x,
                                                                    itimes (iplus (-1,
                                                                                          n),
                                                                                          d)),
                                                                    0))))))

```

THEOREM: iterk_e=1+_n.c-1&x+nd-exists-step-2

```

(ilessp (0, n)
 ^ ilessp (0, d)
 ^ (¬ ilessp (iplus (a, d), iplus (x, itimes (n, d))))))
→ (v&c-apply$ ('iterk,
                list (cons (a, 0),
                            cons (b, 0),

```

```

cons (c, 0),
cons (d, 0),
cons (itimes (iplus (-1, n), iplus (-1, c)), 0),
v&c-apply$ ('iterk,
  list (cons (a, 0),
        cons (b, 0),
        cons (c, 0),
        cons (d, 0),
        '(1 . 0),
        cons (iplus (x, itimes (iplus (-1, n), d)),
              0))))))
↔ v&c-apply$ ('iterk,
  list (cons (a, 0),
        cons (b, 0),
        cons (c, 0),
        cons (d, 0),
        cons (itimes (iplus (-1, n), iplus (-1, c)), 0),
        v&c-apply$ ('iterk,
          list (cons (a, 0),
                cons (b, 0),
                cons (c, 0),
                cons (d, 0),
                cons (c, 0),
                cons (iplus (x, itimes (n, d)), 0))))))

```

THEOREM: iterk_e=1+_n.c-1&x+nd-value-step-2

```

(ilessp (0, n)
 ^ ilessp (0, d)
 ^ (¬ ilessp (iplus (a, d), iplus (x, itimes (n, d))))))
→ (car (v&c-apply$ ('iterk,
  list (cons (a, 0),
        cons (b, 0),
        cons (c, 0),
        cons (d, 0),
        cons (itimes (iplus (-1, n), iplus (-1, c)), 0),
        v&c-apply$ ('iterk,
          list (cons (a, 0),
                cons (b, 0),
                cons (c, 0),
                cons (d, 0),
                '(1 . 0),
                cons (iplus (x,
                          itimes (iplus (-1, n), d)),
                          0))))))

```



```

= car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      cons (itimes (iplus (-1, n), iplus (-1, c)), 0),
                      v&c-apply$ ('iterk,
                                list (cons (a, 0),
                                      cons (b, 0),
                                      cons (c, 0),
                                      cons (d, 0),
                                      cons (c, 0),
                                      cons (iplus (x, itimes (n, d)),
                                            0))))))

```

THEOREM: iterk_e=1+_n.c-1&x+nd-exists-step-3

```

(ilessp (0, n) & (n ≠ 1) & ilessp (1, c))
→ (v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      cons (itimes (iplus (-1, n), iplus (-1, c)), 0),
                      v&c-apply$ ('iterk,
                                list (cons (a, 0),
                                      cons (b, 0),
                                      cons (c, 0),
                                      cons (d, 0),
                                      cons (c, 0),
                                      cons (iplus (x, itimes (n, d)), 0))))))
↔ v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      cons (iplus (1, itimes (n, iplus (-1, c))), 0),
                      cons (iplus (x, itimes (n, d)), 0))))

```

THEOREM: iterk_e=1+_n.c-1&x+nd-value-step-3

```

(ilessp (0, n) & (n ≠ 1) & ilessp (1, c))
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),

```

$$\begin{aligned}
& \text{cons}(d, 0), \\
& \text{cons}(\text{itimes}(\text{iplus}(-1, n), \text{iplus}(-1, c)), 0), \\
& \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0)))))) \\
= & \text{car}(\text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(1, \text{itimes}(n, \text{iplus}(-1, c))), 0), \\
& \quad \quad \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0))))))
\end{aligned}$$

THEOREM: $a+d \geq x + n - 1 \cdot d$ -when- $a+d \geq x + nd \wedge 0 < d \wedge 0 < n$
 $(\text{ilessp}(0, n)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(n, d))))$
 $\rightarrow (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(\text{iplus}(-1, n), d))))$

THEOREM: $\text{iterk}_e = 1 + n \cdot c - 1 \wedge x + nd$ -exists-when- $1 < c \wedge 0 < d \wedge 0 < n \wedge a + d \geq x + nd$
 $(\text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge \text{ilessp}(0, n)$
 $\wedge (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(n, d))))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad '(1 . 0),$
 $\quad \quad \text{cons}(x, 0)))$
 $\leftrightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(\text{iplus}(1, \text{itimes}(n, \text{iplus}(-1, c))), 0),$
 $\quad \quad \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0))))$

THEOREM: $\text{iterk}_e = 1 + n \cdot c - 1 \wedge x + nd$ -value-when- $1 < c \wedge 0 < d \wedge 0 < n \wedge a + d \geq x + nd$
 $(\text{ilessp}(1, c)$

```

 $\wedge$  ilessp(0, d)
 $\wedge$  ilessp(0, n)
 $\wedge$  ( $\neg$  ilessp(iplus(a, d), iplus(x, itimes(n, d))))
 $\rightarrow$  (car(v&c-apply$('iterk,
      list(cons(a, 0),
            cons(b, 0),
            cons(c, 0),
            cons(d, 0),
            '(1 . 0),
            cons(x, 0))))
      = car(v&c-apply$('iterk,
      list(cons(a, 0),
            cons(b, 0),
            cons(c, 0),
            cons(d, 0),
            cons(iplus(1, itimes(n, iplus(-1, c))), 0),
            cons(iplus(x, itimes(n, d)), 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; 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(0, d) then 0
   elseif ilessp(a, x) then 0
   else iplus(1, n(a, d, iplus(x, d))) endif

```

```

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

```

```

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

```

```

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

```

```

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

```

```

THEOREM: iterk-e=1&x-iff-iterk_e=1+_n_c-1&x+nd-when-1<c&0<d&a>=x
(ilessp(1, c)  $\wedge$  ilessp(0, d)  $\wedge$  ( $\neg$  ilessp(a, x)))

```

```

→ (v&c-apply$('iterk,
      list (cons (a, 0),
            cons (b, 0),
            cons (c, 0),
            cons (d, 0),
            '(1 . 0),
            cons (x, 0)))
   ↔ v&c-apply$('iterk,
      list (cons (a, 0),
            cons (b, 0),
            cons (c, 0),
            cons (d, 0),
            cons (iplus (1, itimes (n (a, d, x), iplus (-1, c))),
                  0),
            cons (iplus (x, itimes (n (a, d, x), d)), 0)))

```

THEOREM: iterk-exists-when-e=1+n.c-1&x=x+nd&b<=0

```

((¬ ilessp (0, b)) ∧ ilessp (1, c) ∧ ilessp (0, d))
→ v&c-apply$('iterk,
      list (cons (a, 0),
            cons (b, 0),
            cons (c, 0),
            cons (d, 0),
            cons (iplus (1, itimes (n (a, d, x), iplus (-1, c))), 0),
            cons (iplus (x, itimes (n (a, d, x), d)), 0)))

```

THEOREM: iterk-exists-when-x<=a&d>0&c<1&b<=0

```

((¬ ilessp (0, b)) ∧ ilessp (1, c) ∧ ilessp (0, d) ∧ (¬ ilessp (a, x)))
→ v&c-apply$('iterk,
      list (cons (a, 0),
            cons (b, 0),
            cons (c, 0),
            cons (d, 0),
            '(1 . 0),
            cons (x, 0)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 5 of the
; Main Theorem given above in the introduction.

```

```

; 5. If x <= a, d > 0, c > 1, and b <= 0, then
;    K( a,b,c,d,x ) exists.

```

```

; Proof. By Lemma 7, the definition of the
; function N, Part 1 of the Main Theorem,

```

; and the definition of K.

THEOREM: k -exists-when- $x \leq a \wedge d > 0 \wedge c < 1 \wedge b \leq 0$
 $((\neg \text{ilessp}(0, b)) \wedge \text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge (\neg \text{ilessp}(a, x)))$
 $\rightarrow \forall k. \text{apply}(k, \text{list}(\text{cons}(a, 0), \text{cons}(b, 0), \text{cons}(c, 0), \text{cons}(d, 0), \text{cons}(x, 0)))$

THEOREM: k -halts-when- $x \leq a \wedge d > 0 \wedge c < 1 \wedge b \leq 0$
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge vc-x$
 $\wedge (\neg \text{ilessp}(0, \text{car}(vc-b)))$
 $\wedge \text{ilessp}(1, \text{car}(vc-c))$
 $\wedge \text{ilessp}(0, \text{car}(vc-d))$
 $\wedge (\neg \text{ilessp}(\text{car}(vc-a), \text{car}(vc-x))))$
 $\rightarrow \forall k. \text{apply}(k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x))$

;;
; Lemma 8. Generalize Lemma 7 to the case when $e > 1$.

; Assume $1 < c$, $0 < d$, $1 < e$, $0 < n$, and $a+d \geq x+nd$.

; Then

; $\text{IterK}(a, b, c, d, e, x)$ exists
; iff
; $\text{IterK}(a, b, c, d, e+n(c-1), x+nd)$ exists

; and

; $\text{IterK}(a, b, c, d, e, x)$
; =
; $\text{IterK}(a, b, c, d, e+n(c-1), x+nd)$.

; Proof. By Lemma 6 (with $e_2 = 1$) and Lemma 7.

THEOREM: $x + -x + y = \text{fix-int-}y$
 $\text{ilessp}(0, x) \rightarrow (\text{iplus}(x, \text{iplus}(-x, y)) = \text{fix-int}(y))$

THEOREM: $\text{fix-int-}x = x$ -when- $x > 1$
 $\text{ilessp}(1, x) \rightarrow (\text{fix-int}(x) = x)$

THEOREM: iterk-e-x-iff-iterk-e-1-iterk-1-x

ilessp(1, e)

$$\begin{aligned}
 &\rightarrow (\text{v\&c-apply}\$ ('iterk, \\
 &\quad \text{list}(\text{cons}(a, 0), \\
 &\quad \quad \text{cons}(b, 0), \\
 &\quad \quad \text{cons}(c, 0), \\
 &\quad \quad \text{cons}(d, 0), \\
 &\quad \quad \text{cons}(e, 0), \\
 &\quad \quad \text{cons}(x, 0))) \\
 &\leftrightarrow \text{v\&c-apply}\$ ('iterk, \\
 &\quad \text{list}(\text{cons}(a, 0), \\
 &\quad \quad \text{cons}(b, 0), \\
 &\quad \quad \text{cons}(c, 0), \\
 &\quad \quad \text{cons}(d, 0), \\
 &\quad \quad \text{cons}(\text{iplus}(-1, e), 0), \\
 &\quad \quad \text{v\&c-apply}\$ ('iterk, \\
 &\quad \quad \quad \text{list}(\text{cons}(a, 0), \\
 &\quad \quad \quad \quad \text{cons}(b, 0), \\
 &\quad \quad \quad \quad \text{cons}(c, 0), \\
 &\quad \quad \quad \quad \text{cons}(d, 0), \\
 &\quad \quad \quad \quad '(1 . 0), \\
 &\quad \quad \quad \quad \text{cons}(x, 0))))))
 \end{aligned}$$

THEOREM: iterk-e-x=iterk-e-1-iterk-1-x

ilessp(1, e)

$$\begin{aligned}
 &\rightarrow (\text{car}(\text{v\&c-apply}\$ ('iterk, \\
 &\quad \text{list}(\text{cons}(a, 0), \\
 &\quad \quad \text{cons}(b, 0), \\
 &\quad \quad \text{cons}(c, 0), \\
 &\quad \quad \text{cons}(d, 0), \\
 &\quad \quad \text{cons}(e, 0), \\
 &\quad \quad \text{cons}(x, 0)))) \\
 &= \text{car}(\text{v\&c-apply}\$ ('iterk, \\
 &\quad \text{list}(\text{cons}(a, 0), \\
 &\quad \quad \text{cons}(b, 0), \\
 &\quad \quad \text{cons}(c, 0), \\
 &\quad \quad \text{cons}(d, 0), \\
 &\quad \quad \text{cons}(\text{iplus}(-1, e), 0), \\
 &\quad \quad \text{v\&c-apply}\$ ('iterk, \\
 &\quad \quad \quad \text{list}(\text{cons}(a, 0), \\
 &\quad \quad \quad \quad \text{cons}(b, 0), \\
 &\quad \quad \quad \quad \text{cons}(c, 0), \\
 &\quad \quad \quad \quad \text{cons}(d, 0), \\
 &\quad \quad \quad \quad '(1 . 0),
 \end{aligned}$$

cons (x, 0))))))

THEOREM: iterk_e+_n.c-1&x+nd-exists-when-1<c&0<d&1<e&0<n&a+d>=x+nd-step-1
(ilessp (1, c)

∧ illessp (0, d)

∧ illessp (0, n)

∧ (¬ illessp (iplus (a, d), iplus (x, itimes (n, d))))))

→ (v&c-apply\$ ('iterk,

list (cons (a, 0),

cons (b, 0),

cons (c, 0),

cons (d, 0),

cons (iplus (-1, e), 0),

v&c-apply\$ ('iterk,

list (cons (a, 0),

cons (b, 0),

cons (c, 0),

cons (d, 0),

'(1 . 0),

cons (x, 0))))))

↔ v&c-apply\$ ('iterk,

list (cons (a, 0),

cons (b, 0),

cons (c, 0),

cons (d, 0),

cons (iplus (-1, e), 0),

v&c-apply\$ ('iterk,

list (cons (a, 0),

cons (b, 0),

cons (c, 0),

cons (d, 0),

cons (iplus (1,

itimes (n,

iplus (-1, c))),

0),

cons (iplus (x, itimes (n, d)), 0))))))

THEOREM: iterk_e+_n.c-1&x+nd-value-when-1<c&0<d&1<e&0<n&a+d>=x+nd-step-1
(ilessp (1, c)

∧ illessp (0, d)

∧ illessp (0, n)

∧ (¬ illessp (iplus (a, d), iplus (x, itimes (n, d))))))

→ (car (v&c-apply\$ ('iterk,

list (cons (a, 0),

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (-1, e), 0),
v&c-apply$ ('iterk,
            list (cons (a, 0),
                  cons (b, 0),
                  cons (c, 0),
                  cons (d, 0),
                  '(1 . 0),
                  cons (x, 0))))))
= car (v&c-apply$ ('iterk,
                  list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (iplus (-1, e), 0),
                        v&c-apply$ ('iterk,
                                    list (cons (a, 0),
                                            cons (b, 0),
                                            cons (c, 0),
                                            cons (d, 0),
                                            cons (iplus (1,
                                                         itimes (n,
                                                             iplus (-1,
                                                                 c))),
                                                         0),
                                            cons (iplus (x, itimes (n, d)),
                                                  0)))))))

```

THEOREM: $\text{iterk}_e + \text{n}_c - 1 \wedge x + \text{nd}$ -exists-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d = x + \text{nd}$ -step-2
 $(\text{ilessp}(1, c) \wedge \text{ilessp}(1, e) \wedge \text{ilessp}(0, n))$

```

→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (iplus (-1, e), 0),
                       v&c-apply$ ('iterk,
                                   list (cons (a, 0),
                                             cons (b, 0),
                                             cons (c, 0),
                                             cons (d, 0),
                                             cons (iplus (1, itimes (n, iplus (-1, c))),
                                               0))))))

```


$$\begin{aligned} & 0), \\ & \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0)))) \\ \leftrightarrow & \text{v\&c-apply}\$('iterk, \\ & \text{list}(\text{cons}(a, 0), \\ & \text{cons}(b, 0), \\ & \text{cons}(c, 0), \\ & \text{cons}(d, 0), \\ & \text{cons}(\text{iplus}(e, \text{itimes}(n, \text{iplus}(-1, c))), 0), \\ & \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0)))) \end{aligned}$$

THEOREM: iterk_e+_n.c-1&x+nd-value-when-1<c&0<d&1<e&0<n&a+d>=x+nd-step-2
(ilessp(1, c) \wedge illessp(1, e) \wedge illessp(0, n))

$$\begin{aligned} \rightarrow & (\text{car}(\text{v\&c-apply}\$('iterk, \\ & \text{list}(\text{cons}(a, 0), \\ & \text{cons}(b, 0), \\ & \text{cons}(c, 0), \\ & \text{cons}(d, 0), \\ & \text{cons}(\text{iplus}(-1, e), 0), \\ & \text{v\&c-apply}\$('iterk, \\ & \text{list}(\text{cons}(a, 0), \\ & \text{cons}(b, 0), \\ & \text{cons}(c, 0), \\ & \text{cons}(d, 0), \\ & \text{cons}(\text{iplus}(1, \\ & \text{itimes}(n, \text{iplus}(-1, c))), \\ & 0), \\ & \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0)))))) \\ = & \text{car}(\text{v\&c-apply}\$('iterk, \\ & \text{list}(\text{cons}(a, 0), \\ & \text{cons}(b, 0), \\ & \text{cons}(c, 0), \\ & \text{cons}(d, 0), \\ & \text{cons}(\text{iplus}(e, \text{itimes}(n, \text{iplus}(-1, c))), 0), \\ & \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0)))) \end{aligned}$$

THEOREM: iterk_e+_n.c-1&x+nd-exists-when-1<c&0<d&1<e&0<n&a+d>=x+nd
(ilessp(1, c)

$$\begin{aligned} & \wedge \text{ilessp}(0, d) \\ & \wedge \text{ilessp}(1, e) \\ & \wedge \text{ilessp}(0, n) \\ & \wedge (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(n, d)))) \\ \rightarrow & (\text{v\&c-apply}\$('iterk, \\ & \text{list}(\text{cons}(a, 0), \\ & \text{cons}(b, 0), \end{aligned}$$

$$\begin{aligned}
& \text{cons}(c, 0), \\
& \text{cons}(d, 0), \\
& \text{cons}(e, 0), \\
& \text{cons}(x, 0))) \\
\leftrightarrow & \text{v\&c-apply}\$('iterk, \\
& \text{list}(\text{cons}(a, 0), \\
& \text{cons}(b, 0), \\
& \text{cons}(c, 0), \\
& \text{cons}(d, 0), \\
& \text{cons}(\text{iplus}(e, \text{itimes}(n, \text{iplus}(-1, c))), 0), \\
& \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0)))
\end{aligned}$$

THEOREM: iterk_e+_n.c-1&x+nd-value-when-1<c&0<d&1<e&0<n&a+d>=x+nd

$$\begin{aligned}
& (\text{ilessp}(1, c) \\
& \wedge \text{ilessp}(0, d) \\
& \wedge \text{ilessp}(1, e) \\
& \wedge \text{ilessp}(0, n) \\
& \wedge (\neg \text{ilessp}(\text{iplus}(a, d), \text{iplus}(x, \text{itimes}(n, d)))))) \\
\rightarrow & (\text{car}(\text{v\&c-apply}\$('iterk, \\
& \text{list}(\text{cons}(a, 0), \\
& \text{cons}(b, 0), \\
& \text{cons}(c, 0), \\
& \text{cons}(d, 0), \\
& \text{cons}(e, 0), \\
& \text{cons}(x, 0)))))) \\
= & \text{car}(\text{v\&c-apply}\$('iterk, \\
& \text{list}(\text{cons}(a, 0), \\
& \text{cons}(b, 0), \\
& \text{cons}(c, 0), \\
& \text{cons}(d, 0), \\
& \text{cons}(\text{iplus}(e, \text{itimes}(n, \text{iplus}(-1, c))), 0), \\
& \text{cons}(\text{iplus}(x, \text{itimes}(n, d)), 0))))))
\end{aligned}$$

THEOREM: iterk-e&x-iff-iterk_e+_n.c-1&x+nd-when-1<c&0<d&a>=x&e>1

$$\begin{aligned}
& (\text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(1, e) \wedge (\neg \text{ilessp}(a, x))) \\
\rightarrow & (\text{v\&c-apply}\$('iterk, \\
& \text{list}(\text{cons}(a, 0), \\
& \text{cons}(b, 0), \\
& \text{cons}(c, 0), \\
& \text{cons}(d, 0), \\
& \text{cons}(e, 0), \\
& \text{cons}(x, 0))) \\
\leftrightarrow & \text{v\&c-apply}\$('iterk, \\
& \text{list}(\text{cons}(a, 0),
\end{aligned}$$

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (e, itimes (n (a, d, x), iplus (-1, c))), 0),
cons (iplus (x, itimes (n (a, d, x), d)), 0))))

```

THEOREM: $\text{iterk-e} \wedge x = \text{iterk.e} + \text{n.c} - 1 \wedge x + \text{nd} - \text{when} - 1 < c \wedge 0 < d \wedge a > = x \wedge e > 1$
 $(\text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(1, e) \wedge (\neg \text{ilessp}(a, x)))$

```

→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                cons (e, 0),
                                cons (x, 0))))))
=   car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                cons (iplus (e,
                                                itimes (n (a, d, x), iplus (-1, c))),
                                                0),
                                cons (iplus (x, itimes (n (a, d, x), d)), 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 9. The number of iterates of K can be reduced
;           by one if the value of x is reduced by b
;           when a < x.
;
; Assume 1 < e, and a < x.
;
; Then
;
; IterK( a,b,c,d,e,x ) exists
; iff
; IterK( a,b,c,d,e-1,x-b ) exists
;
; and
;
; IterK( a,b,c,d,e,x )
; =
; IterK( a,b,c,d,e-1,x-b ).

```

; Proof. By Lemma 6 (with $e_2 = 1$), and the definition
 ; of IterK for values of x larger than a .

THEOREM: iterk_e_x-iff-iterk_e-1_x-b-when-1<e&x>a-step-1

(ilessp(a, x) \wedge illessp(1, e))
 \rightarrow (v&c-apply\$('iterk,
 list(cons($a, 0$),
 cons($b, 0$),
 cons($c, 0$),
 cons($d, 0$),
 cons(iplus(-1, e), 0),
 v&c-apply\$('iterk,
 list(cons($a, 0$),
 cons($b, 0$),
 cons($c, 0$),
 cons($d, 0$),
 '(1 . 0),
 cons($x, 0$))))))
 \leftrightarrow v&c-apply\$('iterk,
 list(cons($a, 0$),
 cons($b, 0$),
 cons($c, 0$),
 cons($d, 0$),
 cons(iplus(-1, e), 0),
 cons(idifference(x, b), 0))))

THEOREM: iterk_e_x=iterk_e-1_x-b-when-1<e&x>a-step-1

(ilessp(a, x) \wedge illessp(1, e))
 \rightarrow (car(v&c-apply\$('iterk,
 list(cons($a, 0$),
 cons($b, 0$),
 cons($c, 0$),
 cons($d, 0$),
 cons(iplus(-1, e), 0),
 v&c-apply\$('iterk,
 list(cons($a, 0$),
 cons($b, 0$),
 cons($c, 0$),
 cons($d, 0$),
 '(1 . 0),
 cons($x, 0$))))))
 $=$ car(v&c-apply\$('iterk,
 list(cons($a, 0$),

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (-1, e), 0),
cons (idifference (x, b), 0))))

```

THEOREM: iterk_e_x-iff-iterk_e-1_x-b-when-1<e&x>a

```

(ilessp (a, x) ^ illessp (1, e))
→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (e, 0),
                       cons (x, 0)))
    ↔ v&c-apply$ ('iterk,
                  list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (iplus (-1, e), 0),
                          cons (idifference (x, b), 0))))

```

THEOREM: iterk_e_x=iterk_e-1_x-b-when-1<e&x>a

```

(ilessp (a, x) ^ illessp (1, e))
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (e, 0),
                            cons (x, 0))))
    = car (v&c-apply$ ('iterk,
                      list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              cons (iplus (-1, e), 0),
                              cons (idifference (x, b), 0))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 10. A key fact noted by Knuth in his proof.

```

```

; Assume 1 < c, 0 < d, 1 < e, and x <= a.

```

```

; Then

; IterK( a,b,c,d,e,x ) exists
; iff
; IterK( a,b,c,d,e-1+N( a,d,x )(c-1),x+N( a,d,x )d-b )
; exists

; and

; IterK( a,b,c,d,e,x )
; =
; IterK( a,b,c,d,e-1+N( a,d,x )(c-1),x+N( a,d,x )d-b ).

; Proof. By Lemma 8, the definition of the function N,
; and Lemma 9.

```

THEOREM: $e+n \cdot c-1 > 1$ -when- $1 < c \leq 1 < e \leq 0 < n$
 $(\text{ilessp}(1, c) \wedge \text{ilessp}(1, e) \wedge \text{ilessp}(0, n))$
 $\rightarrow \text{ilessp}(1, \text{iplus}(e, \text{itimes}(n, \text{iplus}(-1, c))))$

THEOREM: $\text{iterk-}e \leq x$ -iff- $\text{iterk-}1+e+n \cdot c-1 \leq x+nd-b$ -when- $1 < c \leq 0 < d \wedge a = x \wedge e > 1$
 $(\text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(1, e) \wedge (\neg \text{ilessp}(a, x)))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \text{cons}(b, 0),$
 $\quad \text{cons}(c, 0),$
 $\quad \text{cons}(d, 0),$
 $\quad \text{cons}(e, 0),$
 $\quad \text{cons}(x, 0)))$
 $\leftrightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \text{cons}(b, 0),$
 $\quad \text{cons}(c, 0),$
 $\quad \text{cons}(d, 0),$
 $\quad \text{cons}(\text{iplus}(-1,$
 $\quad \quad \text{iplus}(e,$
 $\quad \quad \quad \text{itimes}(n(a, d, x), \text{iplus}(-1, c))))),$
 $\quad \quad \quad 0),$
 $\quad \text{cons}(\text{idifference}(\text{iplus}(x, \text{itimes}(n(a, d, x), d)),$
 $\quad \quad \quad b),$
 $\quad \quad \quad 0))))$

THEOREM: $\text{iterk-}e \leq x = \text{iterk-}1+e+n \cdot c-1 \leq x+nd-b$ -when- $1 < c \leq 0 < d \wedge a = x \wedge e > 1$
 $(\text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(1, e) \wedge (\neg \text{ilessp}(a, x)))$

```

→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                cons (e, 0),
                                cons (x, 0))))
    = car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                cons (iplus (-1,
                                              iplus (e,
                                                    itimes (n (a, d, x),
                                                            iplus (-1, c))))),
                                0),
                                cons (idifference (iplus (x,
                                                            itimes (n (a, d, x), d)),
                                                    b),
                                0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 11. Another key fact noted by Knuth.

```

```

; Assume  $0 < b$ ,  $1 < c$ ,  $0 < d$ ,  $1 < e$ , and  $x \leq a$ .

```

```

; Then

```

```

; IterK( a,b,c,d,e-1,x-b ) exists
; iff
; IterK( a,b,c,d,e-1+N( a,d,x )(c-1),x-b+N( a,d,x )d )
; exists

```

```

; and

```

```

; IterK( a,b,c,d,e-1,x-b )
; =
; IterK( a,b,c,d,e-1+N( a,d,x )(c-1),x-b+N( a,d,x )d ).

```

```

; Proof. By Lemma 7 (if  $e = 2$ ), Lemma 8 (if  $e > 2$ ), and
; the definition of the function N.

```

THEOREM: $a+d \geq x-b+nd$ -when- $b > 0 \& a+d \geq x+nd \& d > 0$

```

(ilessp (0, b)
  ^ ilessp (0, d)
  ^ ilessp (0, n)
  ^ (¬ ilessp (iplus (a, d), iplus (x, itimes (n, d))))
  → (¬ ilessp (iplus (a, d), iplus (idifference (x, b), itimes (n, d))))

```

THEOREM: $\text{iterk_e-1} \wedge x=b \text{ iff } \text{iterk_e-1} + _n.c-1 \wedge x=b+nd \text{ when } -1 < c \wedge 0 < d \wedge a = x \wedge b > 0 \wedge e > 2$

```

(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ ilessp (2, e)
  ^ (¬ ilessp (a, x))
  → (v&c-apply$ ('iterk,
                 list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (iplus (-1, e), 0),
                        cons (idifference (x, b), 0)))
     ↔ v&c-apply$ ('iterk,
                   list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (iplus (-1,
                                       iplus (e,
                                             itimes (n (a, d, x), iplus (-1, c))))),
                          0),
                          cons (idifference (iplus (x, itimes (n (a, d, x), d)),
                                             b),
                              0))))

```

THEOREM: $\text{iterk_e-1} \wedge x=b = \text{iterk_e-1} + _n.c-1 \wedge x=b+nd \text{ when } -1 < c \wedge 0 < d \wedge a = x \wedge e > 2$

```

(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ ilessp (2, e)
  ^ (¬ ilessp (a, x))
  → (car (v&c-apply$ ('iterk,
                     list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (iplus (-1, e), 0),

```


$$\begin{aligned}
& \text{cons}(\text{idifference}(x, b), 0))) \\
= & \text{car}(\text{v\&c-apply}\$('iterk, \\
& \text{list}(\text{cons}(a, 0), \\
& \quad \text{cons}(b, 0), \\
& \quad \text{cons}(c, 0), \\
& \quad \text{cons}(d, 0), \\
& \quad \text{cons}(\text{iplus}(-1, \\
& \quad \quad \text{iplus}(e, \\
& \quad \quad \quad \text{itimes}(\text{n}(a, d, x), \\
& \quad \quad \quad \text{iplus}(-1, c))))), \\
& \quad 0), \\
& \text{cons}(\text{idifference}(\text{iplus}(x, \\
& \quad \quad \text{itimes}(\text{n}(a, d, x), d)), \\
& \quad \quad b), \\
& \quad 0))))))
\end{aligned}$$

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

THEOREM: $\text{iterk}_e-1 \& x=b$ iff $\text{iterk}_e-1 + \text{n}_c-1 \& x=b+nd$ when $1 < c \& 0 < d \& a = x \& b > 0 \& e=2$

$$\begin{aligned}
& (\text{ilessp}(0, b) \\
& \wedge \text{ilessp}(1, c) \\
& \wedge \text{ilessp}(0, d) \\
& \wedge (e = 2) \\
& \wedge (\neg \text{ilessp}(a, x))) \\
\rightarrow & (\text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(-1, e), 0), \\
& \quad \quad \text{cons}(\text{idifference}(x, b), 0))) \\
\leftrightarrow & \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(-1, \\
& \quad \quad \quad \text{iplus}(e, \\
& \quad \quad \quad \quad \text{itimes}(\text{n}(a, d, x), \text{iplus}(-1, c))))), \\
& \quad \quad 0), \\
& \quad \text{cons}(\text{idifference}(\text{iplus}(x, \text{itimes}(\text{n}(a, d, x), d)), \\
& \quad \quad b), \\
& \quad \quad 0))))))
\end{aligned}$$

THEOREM: $\text{iterk-e-1} \wedge x=b = \text{iterk-e-1} + \text{n-c-1} \wedge x=b + \text{nd-when-1} < c \wedge 0 < d \wedge a > = x \wedge b > 0 \wedge e=2$

$$\begin{aligned}
& (\text{ilessp}(0, b) \\
& \wedge \text{ilessp}(1, c) \\
& \wedge \text{ilessp}(0, d) \\
& \wedge (e = 2) \\
& \wedge (\neg \text{ilessp}(a, x))) \\
\rightarrow & (\text{car}(\text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(-1, e), 0), \\
& \quad \quad \text{cons}(\text{idifference}(x, b), 0)))) \\
= & \text{car}(\text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(-1, \\
& \quad \quad \quad \text{iplus}(e, \\
& \quad \quad \quad \quad \text{itimes}(\text{n}(a, d, x), \\
& \quad \quad \quad \quad \quad \text{iplus}(-1, c))))), \\
& \quad \quad \quad 0), \\
& \quad \quad \text{cons}(\text{idifference}(\text{iplus}(x, \\
& \quad \quad \quad \quad \text{itimes}(\text{n}(a, d, x), d)), \\
& \quad \quad \quad \quad b), \\
& \quad \quad \quad 0))))))
\end{aligned}$$

THEOREM: $\text{iterk-e-1} \wedge x=b \text{ iff } \text{iterk-e-1} + \text{n-c-1} \wedge x=b + \text{nd-when-1} < c \wedge 0 < d \wedge a > = x \wedge b > 0 \wedge e=1$

$$\begin{aligned}
& (\text{ilessp}(0, b) \\
& \wedge \text{ilessp}(1, c) \\
& \wedge \text{ilessp}(0, d) \\
& \wedge \text{ilessp}(1, e) \\
& \wedge (\neg \text{ilessp}(a, x))) \\
\rightarrow & (\text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(-1, e), 0), \\
& \quad \quad \text{cons}(\text{idifference}(x, b), 0))) \\
\leftrightarrow & \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0),
\end{aligned}$$

```

cons (c, 0),
cons (d, 0),
cons (iplus (-1,
             iplus (e,
                    itimes (n (a, d, x), iplus (-1, c)))))
0),
cons (idifference (iplus (x, itimes (n (a, d, x), d)),
                  b),
0))))

```

THEOREM: $\text{iterk}_{-e-1}(x, b) = \text{iterk}_{-e-1} + n_{-c-1}(x, b) + nd$ -when $-1 < c < 0 < d \wedge a \geq x \wedge b > 0 \wedge e > 1$
 $(\text{ilessp}(0, b)$

$\wedge \text{ilessp}(1, c)$

$\wedge \text{ilessp}(0, d)$

$\wedge \text{ilessp}(1, e)$

$\wedge (\neg \text{ilessp}(a, x))$

$\rightarrow (\text{car}(\text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$
 $\text{cons}(d, 0),$
 $\text{cons}(\text{iplus}(-1, e), 0),$
 $\text{cons}(\text{idifference}(x, b), 0))))$

$= \text{car}(\text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$
 $\text{cons}(d, 0),$
 $\text{cons}(\text{iplus}(-1,$
 $\text{iplus}(e,$
 $\text{itimes}(n(a, d, x),$
 $\text{iplus}(-1, c))))),$
 $0),$
 $\text{cons}(\text{idifference}(\text{iplus}(x,$
 $\text{itimes}(n(a, d, x), d)),$
 $b),$
 $0))))$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 12. The number of iterates of K can be reduced
; by one if the value of x is reduced by b
; when a >= x and restrictions are placed
; on the parameters a,b,c, and d.

```

```

; Assume 0 < b, 1 < c, 0 < d, 1 < e, and a >= x.

; Then

; IterK( a,b,c,d,e,x ) exists
; iff
; IterK( a,b,c,d,e-1,x-b ) exists

; and

; IterK( a,b,c,d,e,x )
; =
; IterK( a,b,c,d,e-1,x-b ).

; Proof. By Lemma 10 and Lemma 11.

```

THEOREM: iterk_e_x-iff-iterk_e-1_x-b-when-0<b&1<c&0<d&1<e&a>=x

```

(ilessp (0, b)
 ^ illessp (1, c)
 ^ illessp (0, d)
 ^ illessp (1, e)
 ^ (¬ illessp (a, x)))
→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (e, 0),
                       cons (x, 0)))
   ↔ v&c-apply$ ('iterk,
                 list (cons (a, 0),
                         cons (b, 0),
                         cons (c, 0),
                         cons (d, 0),
                         cons (iplus (-1, e), 0),
                         cons (idifference (x, b), 0))))

```

THEOREM: iterk_e_x=iterk_e-1_x-b-when-0<b&1<c&0<d&1<e&a>=x

```

(ilessp (0, b)
 ^ illessp (1, c)
 ^ illessp (0, d)
 ^ illessp (1, e)
 ^ (¬ illessp (a, x)))

```

```

→ (car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (e, 0),
                    cons (x, 0))))
    = car (v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (iplus (-1, e), 0),
                    cons (idifference (x, b), 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 13. Combine Lemma 9 and Lemma 12 about reducing
;           the number of iterates of K by one.

```

```

; Assume 0 < b, 1 < c, 0 < d, and 1 < e.

```

```

; Then

```

```

; IterK( a,b,c,d,e,x ) exists
; iff
; IterK( a,b,c,d,e-1,x-b ) exists

```

```

; and

```

```

; IterK( a,b,c,d,e,x )
; =
; IterK( a,b,c,d,e-1,x-b ).

```

```

; Proof. By Lemma 9 (if x > a) and Lemma 12 (if x <= a).

```

THEOREM: iterk_e_x-iff-iterk_e-1_x-b-when-0<b&1<c&0<d&1<e
 (ilessp(0, b) ∧ illessp(1, c) ∧ illessp(0, d) ∧ illessp(1, e))

```

→ (v&c-apply$ ('iterk,
                list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    cons (e, 0),

```

$$\leftrightarrow \text{v\&c-apply}\$ ('iterk, \text{list}(\text{cons}(a, 0), \text{cons}(b, 0), \text{cons}(c, 0), \text{cons}(d, 0), \text{cons}(\text{iplus}(-1, e), 0), \text{cons}(\text{idifference}(x, b), 0))))$$

THEOREM: $\text{iterk}_e x = \text{iterk}_{e-1} x$ when $0 < b < 1 < c < 0 < d < 1 < e$
 $(\text{ilessp}(0, b) \wedge \text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(1, e))$

$$\begin{aligned} &\rightarrow (\text{car}(\text{v\&c-apply}\$ ('iterk, \text{list}(\text{cons}(a, 0), \text{cons}(b, 0), \text{cons}(c, 0), \text{cons}(d, 0), \text{cons}(e, 0), \text{cons}(x, 0)))))) \\ &= \text{car}(\text{v\&c-apply}\$ ('iterk, \text{list}(\text{cons}(a, 0), \text{cons}(b, 0), \text{cons}(c, 0), \text{cons}(d, 0), \text{cons}(\text{iplus}(-1, e), 0), \text{cons}(\text{idifference}(x, b), 0)))))) \end{aligned}$$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 14. Generalize Lemma 13 by reducing the number
;           of iterates of K by more than one.

; Assume 0 < b, 1 < c, 0 < d, 1 < e, and 0 < j < e.

; Then

; IterK( a,b,c,d,e,x ) exists
; iff
; IterK( a,b,c,d,e-j,x-jb ) exists

; and

; IterK( a,b,c,d,e,x )
; =
; IterK( a,b,c,d,e-j,x-jb ).

```

; Proof. By induction on j .

THEOREM: $\text{iterk_e_x-iff-iterk_e_j_x-jb-when-}0 < b & 1 < c & 0 < d & 1 < e & 0 < j < e$

```
(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ ilessp (1, e)
  ^ ilessp (0, j)
  ^ ilessp (j, e))
→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (e, 0),
                       cons (x, 0)))
   ↔ v&c-apply$ ('iterk,
                 list (cons (a, 0),
                         cons (b, 0),
                         cons (c, 0),
                         cons (d, 0),
                         cons (idifference (e, j), 0),
                         cons (idifference (x, itimes (b, j)), 0))))
```

THEOREM: $\text{iterk_e_x=iterk_e_j_x-jb-when-}0 < b & 1 < c & 0 < d & 1 < e & 0 < j < e$

```
(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ ilessp (1, e)
  ^ ilessp (0, j)
  ^ ilessp (j, e))
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (e, 0),
                            cons (x, 0))))
   = car (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                               cons (b, 0),
                               cons (c, 0),
                               cons (d, 0),
```

```

cons (idifference (e, j), 0),
cons (idifference (x, itimes (b, j)), 0))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 15. A special case of Lemma 14.

```

```

; Assume 0 < b, 1 < c, 0 < d, and a >= x.

```

```

; Then

```

```

; IterK( a,b,c,d,1,x ) exists
; iff
; IterK( a,b,c,d,1,x+d-(c-1)b ) exists

```

```

; and

```

```

; IterK( a,b,c,d,1,x )
; =
; IterK( a,b,c,d,1,x+d-(c-1)b ).

```

```

; Proof. By Lemma 14 with e = c, j = c-1, and
; x replaced by x + d.

```

THEOREM: iterk_x-iff-iterk_x+d_c-1_b-when-0<b&1<c&0<d&a>=x
 (ilessp(0, b) ^ illessp(1, c) ^ illessp(0, d) ^ (¬ illessp(a, x)))

```

→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       '(1 . 0),
                       cons (x, 0)))
   ↔ v&c-apply$ ('iterk,
                 list (cons (a, 0),
                         cons (b, 0),
                         cons (c, 0),
                         cons (d, 0),
                         '(1 . 0),
                         cons (iplus (x,
                                      idifference (d,
                                                    itimes (b, iplus (-1, c))),
                                      0))))

```

THEOREM: iterk_x=iterk_x+d_c-1_b-when-0<b&1<c&0<d&a>=x


```

(ilessp(0, b) ∧ illessp(1, c) ∧ illessp(0, d) ∧ (¬ illessp(a, x)))
→ (car(v&c-apply$('iterk,
                list(cons(a, 0),
                    cons(b, 0),
                    cons(c, 0),
                    cons(d, 0),
                    '(1 . 0),
                    cons(x, 0))))
    = car(v&c-apply$('iterk,
                list(cons(a, 0),
                    cons(b, 0),
                    cons(c, 0),
                    cons(d, 0),
                    '(1 . 0),
                    cons(iplus(x,
                            idifference(d,
                                    itimes(b,
                                            iplus(-1, c))))),
                    0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events give a version of Part 6 of
; the Main Theorem given above in the introduction.

```

```

; 6. If  $x \leq a$ ,  $d > 0$ ,  $c > 1$ , and  $b > 0$ , then
;  $K(a, b, c, d, x)$  exists if and only if
;  $K(a, b, c, d, x + d - (c - 1)b)$  exists and
;  $K(a, b, c, d, x) = K(a, b, c, d, x + d - (c - 1)b)$ .

```

```

; Proof. By Lemma 15 and the definition of K.

```

THEOREM: k_x -iff- $k_{x+d-c-1b}$ -when- $0 < b & 1 < c & 0 < d & a \geq x$
 $(ilessp(0, b) \wedge illessp(1, c) \wedge illessp(0, d) \wedge (\neg illessp(a, x)))$

```

→ (v&c-apply$('k,
                list(cons(a, 0), cons(b, 0), cons(c, 0), cons(d, 0), cons(x, 0)))
    ↔ v&c-apply$('k,
                list(cons(a, 0),
                    cons(b, 0),
                    cons(c, 0),
                    cons(d, 0),
                    cons(iplus(x,
                            idifference(d,
                                    itimes(b, iplus(-1, c))))),
                    0))))

```



```

^  vc-b
^  vc-c
^  vc-d
^  vc-x
^  ilessp (0, car (vc-b))
^  ilessp (1, car (vc-c))
^  ilessp (0, car (vc-d))
^  (¬ ilessp (car (vc-a), car (vc-x)))
→ (car (v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))
    =  car (v&c-apply$ ('k,
                    list (cons (car (vc-a), 0),
                                cons (car (vc-b), 0),
                                cons (car (vc-c), 0),
                                cons (car (vc-d), 0),
                                cons (iplus (car (vc-x),
                                            idifference (car (vc-d),
                                                         itimes (car (vc-b),
                                                             iplus (-1,
                                                                 car (vc-c))))),
                                            0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events give a more general version
; of Part 6 of the Main Theorem.

```

THEOREM: k_x -iff- $k_x+d.c-1_b$ -when- $0 < b < 1 < c < 0 < d & a > = x$ -version-3

```

(vc-a
^  vc-b
^  vc-c
^  vc-d
^  vc-x
^  ilessp (0, car (vc-b))
^  ilessp (1, car (vc-c))
^  ilessp (0, car (vc-d))
^  (¬ ilessp (car (vc-a), car (vc-x)))
→ (v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x))
    ↔ v&c-apply$ ('k,
                  list (vc-a,
                        vc-b,
                        vc-c,
                        vc-d,
                        cons (iplus (car (vc-x),
                                    idifference (car (vc-d),

```

```

itimes(car(vc-b),
        iplus(-1,
              car(vc-c))))),
cost))))

```

THEOREM: $k_x = k_x + d_c - 1$ when $0 < b & 1 < c & 0 < d & a \geq x$ version-3

```

(vc-a
 ^ vc-b
 ^ vc-c
 ^ vc-d
 ^ vc-x
 ^ ilessp(0, car(vc-b))
 ^ ilessp(1, car(vc-c))
 ^ ilessp(0, car(vc-d))
 ^ (¬ ilessp(car(vc-a), car(vc-x))))
 → (car(v&c-apply$('k, list(vc-a, vc-b, vc-c, vc-d, vc-x)))
     = car(v&c-apply$('k,
                    list(vc-a,
                        vc-b,
                        vc-c,
                        vc-d,
                        cons(iplus(car(vc-x),
                                idifference(car(vc-d),
                                        itimes(car(vc-b),
                                                iplus(-1,
                                                        car(vc-c))))),
                                cost))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 16. The parameter c can be replaced by 1 if
;           the parameter d is modified in a suitable
;           way, when certain restrictions are placed
;           on the parameters.

```

```

; Assume 0 < b, 1 < c, 0 < d, and (c-1)b < d.

```

```

; Then

```

```

; IterK( a,b,c,d,1,x ) exists
; iff
; IterK( a,b,1,d-(c-1)b,1,x ) exists

```

```

; and

```

```

; IterK( a,b,c,d,1,x )
; =
; IterK( a,b,1,d-(c-1)b,1,x ).

```

; Proof. By induction on the value given by

```

;     if x > a then 0
;     else 1 + a - x.

```

; Also use Lemma 15.

THEOREM: $k\text{-measure}_{x+d-c-1}b < k\text{-measure}_x\text{-when-}c-1.b < d \& a > = x$
 $(\text{ilessp}(\text{itimes}(b, c), \text{iplus}(b, d)) \wedge (\neg \text{ilessp}(a, x)))$
 $\rightarrow (k\text{-measure}(a, \text{iplus}(b, \text{iplus}(d, \text{iplus}(x, \text{ineg}(\text{itimes}(b, c))))))$
 $< k\text{-measure}(a, x))$

DEFINITION:

```

induct-hint-1-k-measure(a, b, c, d, x)
= if ¬ illessp(itimes(b, iplus(-1, c)), d) then t
  elseif illessp(a, x) then t
  else induct-hint-1-k-measure(a,
                               b,
                               c,
                               d,
                               iplus(x,
                                     idifference(d,
                                                  itimes(b,
                                                       iplus(-1,
                                                            c)))))) endif

```

THEOREM: $\text{iterk}_{c \& d} \text{iff-iterk}_{1 \& d-c-1} b \text{-when-} 0 < b \& 1 < c \& 0 < d \& b-c-1 < d$
 $(\text{ilessp}(0, b)$
 $\wedge \text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge \text{ilessp}(\text{itimes}(b, \text{iplus}(-1, c)), d))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$
 $\text{cons}(d, 0),$
 $'(1 . 0),$
 $\text{cons}(x, 0)))$
 $\leftrightarrow \text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$

```

cons (b, 0),
'(1 . 0),
cons (idifference (d, itimes (b, iplus (-1, c))), 0),
'(1 . 0),
cons (x, 0)))

```

THEOREM: $\text{iterk}_c d = \text{iterk}_{1-d-c-1} b$ when $0 < b < 1 < c < 0 < d & b-c-1 < d$

```

(ilessp (0, b)
 ^ ilessp (1, c)
 ^ ilessp (0, d)
 ^ ilessp (itimes (b, iplus (-1, c)), d))
→ (car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                '(1 . 0),
                                cons (x, 0))))))
= car (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                '(1 . 0),
                                cons (idifference (d, itimes (b, iplus (-1, c))),
                                                0),
                                '(1 . 0),
                                cons (x, 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events give a version of Part 7 of
; the Main Theorem given above in the introduction.

```

```

; 7. If  $d > 0$ ,  $c > 1$ ,  $b > 0$ , and  $(c-1)b < d$ , then
;  $K(a, b, c, d, x)$  exists if and only if
;  $K(a, b, 1, d-(c-1)b, x)$  exists and
;  $K(a, b, c, d, x) = K(a, b, 1, d-(c-1)b, x)$ .

```

```

; Proof. By Lemma 16.

```

THEOREM: $k_c d \text{ iff } k_{1-d-c-1} b$ when $0 < b < 1 < c < 0 < d & b-c-1 < d$

```

(ilessp (0, b)
 ^ ilessp (1, c)
 ^ ilessp (0, d)
 ^ ilessp (itimes (b, iplus (-1, c)), d))
→ (v&c-apply$ ('k,

```

$$\begin{aligned}
& \text{list (cons (a, 0), cons (b, 0), cons (c, 0), cons (d, 0), cons (x, 0))} \\
\leftrightarrow & \text{v\&c-apply\$ ('k,} \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{'(1 . 0),} \\
& \quad \quad \text{cons (idifference (d, itimes (b, iplus (-1, c))), 0),} \\
& \quad \quad \text{cons (x, 0))}
\end{aligned}$$

THEOREM: $k \cdot c + d = k \cdot 1 + d - c - 1 \cdot b$ -when- $0 < b & 1 < c & 0 < d & b - c - 1 < d$

$$\begin{aligned}
& (\text{ilessp (0, b)} \\
& \wedge \text{ilessp (1, c)} \\
& \wedge \text{ilessp (0, d)} \\
& \wedge \text{ilessp (itimes (b, iplus (-1, c)), d)} \\
\rightarrow & (\text{car (v\&c-apply\$ ('k,} \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{cons (c, 0),} \\
& \quad \quad \text{cons (d, 0),} \\
& \quad \quad \text{cons (x, 0))}) \\
= & \text{car (v\&c-apply\$ ('k,} \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{'(1 . 0),} \\
& \quad \quad \text{cons (idifference (d, itimes (b, iplus (-1, c))),} \\
& \quad \quad \quad \text{0),} \\
& \quad \quad \text{cons (x, 0))})
\end{aligned}$$

THEOREM: $k \cdot c + d$ -iff- $k \cdot 1 + d - c - 1 \cdot b$ -when- $0 < b & 1 < c & 0 < d & b - c - 1 < d$ -version-2

$$\begin{aligned}
& (vc-a \\
& \wedge vc-b \\
& \wedge vc-c \\
& \wedge vc-d \\
& \wedge vc-x \\
& \wedge \text{ilessp (0, car (vc-b))} \\
& \wedge \text{ilessp (1, car (vc-c))} \\
& \wedge \text{ilessp (0, car (vc-d))} \\
& \wedge \text{ilessp (itimes (car (vc-b), iplus (-1, car (vc-c))), car (vc-d))} \\
\rightarrow & (\text{v\&c-apply\$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)} \\
& \leftrightarrow \text{v\&c-apply\$ ('k,} \\
& \quad \text{list (cons (car (vc-a), 0),} \\
& \quad \quad \text{cons (car (vc-b), 0),} \\
& \quad \quad \text{'(1 . 0),} \\
& \quad \quad \text{cons (idifference (car (vc-d),} \\
& \quad \quad \quad \text{itimes (car (vc-b),}
\end{aligned}$$

$$\begin{aligned} & \text{iplus}(-1, \text{car}(vc-c))), \\ & 0), \\ & \text{cons}(\text{car}(vc-x), 0))) \end{aligned}$$

THEOREM: $k_c \& d = k_1 \& d - c - 1$ when $0 < b \& 1 < c \& 0 < d \& b - c - 1 < d$ version-2

$$\begin{aligned} & (vc-a \\ & \wedge vc-b \\ & \wedge vc-c \\ & \wedge vc-d \\ & \wedge vc-x \\ & \wedge \text{ilessp}(0, \text{car}(vc-b)) \\ & \wedge \text{ilessp}(1, \text{car}(vc-c)) \\ & \wedge \text{ilessp}(0, \text{car}(vc-d)) \\ & \wedge \text{ilessp}(\text{itimes}(\text{car}(vc-b), \text{iplus}(-1, \text{car}(vc-c))), \text{car}(vc-d))) \\ \rightarrow & (\text{car}(\text{v\&c-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x))) \\ & = \text{car}(\text{v\&c-apply}\$('k, \\ & \quad \text{list}(\text{cons}(\text{car}(vc-a), 0), \\ & \quad \quad \text{cons}(\text{car}(vc-b), 0), \\ & \quad \quad ' (1 . 0), \\ & \quad \quad \text{cons}(\text{idifference}(\text{car}(vc-d), \\ & \quad \quad \quad \text{itimes}(\text{car}(vc-b), \\ & \quad \quad \quad \quad \text{iplus}(-1, \\ & \quad \quad \quad \quad \quad \text{car}(vc-c))), \\ & \quad \quad \quad 0), \\ & \quad \quad \text{cons}(\text{car}(vc-x), 0)))))) \end{aligned}$$

;;
; The next two events give a more general version
; of Part 7 of the Main Theorem.

THEOREM: $k_c \& d$ iff $k_1 \& d - c - 1$ when $0 < b \& 1 < c \& 0 < d \& b - c - 1 < d$ version-3

$$\begin{aligned} & (vc-a \\ & \wedge vc-b \\ & \wedge vc-c \\ & \wedge vc-d \\ & \wedge vc-x \\ & \wedge \text{ilessp}(0, \text{car}(vc-b)) \\ & \wedge \text{ilessp}(1, \text{car}(vc-c)) \\ & \wedge \text{ilessp}(0, \text{car}(vc-d)) \\ & \wedge \text{ilessp}(\text{itimes}(\text{car}(vc-b), \text{iplus}(-1, \text{car}(vc-c))), \text{car}(vc-d))) \\ \rightarrow & (\text{v\&c-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x)) \\ & \leftrightarrow \text{v\&c-apply}\$('k, \\ & \quad \text{list}(vc-a, \\ & \quad \quad vc-b, \end{aligned}$$


```

cons (1, cost1),
cons (idifference (car (vc-d),
                  itimes (car (vc-b),
                          iplus (-1, car (vc-c)))),
      cost2),
vc-x)))

```

THEOREM: k_c&d=k_1&d-c-1_b-when-0<b&1<c&0<d&b_c-1_<d-version-3

```

(vc-a
 ^ vc-b
 ^ vc-c
 ^ vc-d
 ^ vc-x
 ^ illessp (0, car (vc-b))
 ^ illessp (1, car (vc-c))
 ^ illessp (0, car (vc-d))
 ^ illessp (itimes (car (vc-b), iplus (-1, car (vc-c))), car (vc-d)))
 → (car (v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))
    = car (v&c-apply$ ('k,
                      list (vc-a,
                          vc-b,
                          cons (1, cost1),
                          cons (idifference (car (vc-d),
                                              itimes (car (vc-b),
                                                  iplus (-1,
                                                       car (vc-c))))),
                              cost2),
                          vc-x))))

```

;;;

```

; Lemma 17. Essentially the if-part of Knuth's
; theorem characterizing the parameters
; when K is total.

```

```

; Assume 0 < b, 1 < c, 0 < d, and (c-1)b < d.

```

```

; Then IterK( a,b,c,d,1,x ) exists.

```

```

; Proof. By Lemma 4 and Lemma 16.

```

THEOREM: iterk-exists-when-0<b&1<c&0<d&b_c-1_<d

```

(illessp (0, b)
 ^ illessp (1, c)
 ^ illessp (0, d)

```

```

 $\wedge$  ilessp(itimes( $b$ , iplus(-1,  $c$ )),  $d$ )
 $\rightarrow$  v&c-apply$('iterk,
                list(cons( $a$ , 0),
                      cons( $b$ , 0),
                      cons( $c$ , 0),
                      cons( $d$ , 0),
                      '(1 . 0),
                      cons( $x$ , 0)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 8 of the
; Main Theorem given above in the introduction.

```

```

; 8. If  $d > 0$ ,  $c > 1$ ,  $b > 0$ , and  $(c-1)b < d$ , then
;     $K(a, b, c, d, x)$  exists.

;    Proof. By Lemma 17.

```

```

THEOREM: k-exists-when- $0 < b < 1 < c < 0 < d & b \cdot c - 1 < d$ 
(ilessp(0,  $b$ )
 $\wedge$  ilessp(1,  $c$ )
 $\wedge$  ilessp(0,  $d$ )
 $\wedge$  ilessp(itimes( $b$ , iplus(-1,  $c$ )),  $d$ )
 $\rightarrow$  v&c-apply$('k,
                list(cons( $a$ , 0), cons( $b$ , 0), cons( $c$ , 0), cons( $d$ , 0), cons( $x$ , 0)))

```

```

THEOREM: k-halts-when- $0 < b < 1 < c < 0 < d & b \cdot c - 1 < d$ 
( $vc-a$ 
 $\wedge$   $vc-b$ 
 $\wedge$   $vc-c$ 
 $\wedge$   $vc-d$ 
 $\wedge$   $vc-x$ 
 $\wedge$  ilessp(0, car( $vc-b$ ))
 $\wedge$  ilessp(1, car( $vc-c$ ))
 $\wedge$  ilessp(0, car( $vc-d$ ))
 $\wedge$  ilessp(itimes(car( $vc-b$ ), iplus(-1, car( $vc-c$ ))), car( $vc-d$ )))
 $\rightarrow$  v&c-apply$('k, list( $vc-a$ ,  $vc-b$ ,  $vc-c$ ,  $vc-d$ ,  $vc-x$ ))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 18. The cost of computing  $K$  is lowered when  $d$  is
;    added to  $x$ .

```

```

; Assume  $1 < c$ ,  $a \geq x$ , and IterK( $a, b, c, d, 1, x$ ) exists.

```

```

; Then IterK( a,b,c,d,1,x+d ) exists

; and

; cost[ IterK( a,b,c,d,1,x+d ) ]
; <
; cost[ IterK( a,b,c,d,1,x ) ]

; Proof. By the definition of IterK and Lemma 2.

```

THEOREM: iterk-x+d-exists-when-iterk-exists&1<c&a>=x

```

(ilessp(1, c)
 ^ (¬ illessp(a, x))
 ^ v&c-apply$('iterk,
               list(cons(a, 0),
                    cons(b, 0),
                    cons(c, 0),
                    cons(d, 0),
                    '(1 . 0),
                    cons(x, 0))))
→ v&c-apply$('iterk,
              list(cons(a, 0),
                   cons(b, 0),
                   cons(c, 0),
                   cons(d, 0),
                   '(1 . 0),
                   cons(iplus(d, x), 0)))

```

THEOREM: iterk-x+d-cost<cost-iterk-when-iterk-exists&1<c&a>=x

```

(ilessp(1, c)
 ^ (¬ illessp(a, x))
 ^ v&c-apply$('iterk,
               list(cons(a, 0),
                    cons(b, 0),
                    cons(c, 0),
                    cons(d, 0),
                    '(1 . 0),
                    cons(x, 0))))
→ (cdr(v&c-apply$('iterk,
                  list(cons(a, 0),
                       cons(b, 0),
                       cons(c, 0),
                       cons(d, 0),

```

```

                                '(1 . 0),
                                cons(iplus(d, x), 0)))
< cdr(v&c-apply$( 'iterk,
                  list(cons(a, 0),
                        cons(b, 0),
                        cons(c, 0),
                        cons(d, 0),
                        '(1 . 0),
                        cons(x, 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 19. The cost of computing K is lowered
;           when K is iterated c times and
;           "enough" d's are added to x.
;
; Assume 1 < c, a >= x, 0 < d, and
; IterK( a,b,c,d,1,x ) exists.
;
; Then IterK( a,b,c,d,c,x+N( a,d,x )d ) exists
;
; and
;
; cost[ IterK( a,b,c,d,c,x+N( a,d,x )d ) ]
; <
; cost[ IterK( a,b,c,d,1,x ) ]
;
; Proof. Hold a fixed and induct on the
; value given by
;
; if x > a then 0
; else 1 + a - x.

```

THEOREM: $n=1$ -when- $x+d > a \wedge a \geq x \wedge d > 0$
 $(\text{ilessp}(0, d) \wedge (\neg \text{ilessp}(a, x)) \wedge \text{ilessp}(a, \text{iplus}(d, x)))$
 $\rightarrow (n(a, d, x) = 1)$

DEFINITION:
 $\text{induct-hint-2-k-measure}(a, d, x)$
 $=$ **if** $\neg \text{ilessp}(0, d)$ **then** t
elseif $\text{ilessp}(a, x)$ **then** t
elseif $\text{ilessp}(a, \text{iplus}(d, x))$ **then** t
else $\text{induct-hint-2-k-measure}(a, d, \text{iplus}(d, x))$ **endif**

THEOREM: $\text{iterk-x+nd-exists-when-iterk-exists} \wedge 1 < c \wedge a \geq x \wedge 0 < d$

```

(ilessp (1, c)
  ^  illessp (0, d)
  ^  (¬ illessp (a, x))
  ^  v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      '(1 . 0),
                      cons (x, 0))))
→  v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      cons (c, 0),
                      cons (iplus (x, itimes (d, n (a, d, x))), 0)))

```

THEOREM: iterk-x+nd-cost<cost-iterk-when-iterk-exists&1<c&a>=x&0<d

```

(ilessp (1, c)
  ^  illessp (0, d)
  ^  (¬ illessp (a, x))
  ^  v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      '(1 . 0),
                      cons (x, 0))))
→  (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          cons (c, 0),
                          cons (iplus (x, itimes (d, n (a, d, x))), 0))))
    <  cdr (v&c-apply$ ('iterk,
                      list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            '(1 . 0),
                            cons (x, 0))))))

```

;;;

```

; Lemma 20. The number of iterates of K can be
;           reduced to 1 if "enough" b's are
;           subtracted from x.

; Assume 0 < b, 1 < c, 0 < d, 1 < e, 0 < i < e,
; and IterK( a,b,c,d,e,x ) exists.

; Then IterK( a,b,c,d,1,x-ib ) exists.

; Proof. By Lemma 14 and Lemma 2.

```

THEOREM: $e-i=1$ -when- $1+i \geq e$ & $i < e$
 $((\neg \text{ilessp}(\text{iplus}(1, i), e)) \wedge \text{ilessp}(i, e)) \rightarrow (\text{iplus}(e, \text{ineg}(i)) = 1)$

THEOREM: $\text{iterk}_1(x-ib)$ -exists-when- $\text{iterk}_e(x)$ -exists & $0 < b < 1 < c & 0 < d < 1 < e & 0 < i < e$

```

(ilessp(0, b)
 ^ illessp(1, c)
 ^ illessp(0, d)
 ^ illessp(1, e)
 ^ illessp(0, i)
 ^ illessp(i, e)
 ^ v&c-apply$('iterk,
               list(cons(a, 0),
                    cons(b, 0),
                    cons(c, 0),
                    cons(d, 0),
                    cons(e, 0),
                    cons(x, 0))))
→ v&c-apply$('iterk,
              list(cons(a, 0),
                   cons(b, 0),
                   cons(c, 0),
                   cons(d, 0),
                   '(1 . 0),
                   cons(idifference(x, itimes(b, i)), 0)))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 21. Another lemma reducing the number of
;           iterates of K and subtracting b's
;           from x.

```

```

; Assume 0 < b, 1 < c, 0 < d, 0 < i, a+ib < x+b,
; and IterK( a,b,c,d,i+1,x ) exists.

```

```

; Then IterK( a,b,c,d,1,x-ib ) exists

; and

; cost[ IterK( a,b,c,d,1,x-ib ) ]
; <
; cost[ IterK( a,b,c,d,i+1,x ) ]

; Proof. By Lemma 20 (for the exists portion)
; and induction on i (for the cost portion).

```

THEOREM: iterk_1_x-ib_cost<cost-iterk_i+1_x-base-case

```

(ilessp (0, b)
 ^ ilessp (1, c)
 ^ ilessp (0, d)
 ^ (i = 1)
 ^ ilessp (iplus (a, itimes (b, i)), iplus (x, b))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (iplus (1, i), 0),
                       cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
                   list (cons (a, 0),
                           cons (b, 0),
                           cons (c, 0),
                           cons (d, 0),
                           '(1 . 0),
                           cons (idifference (x, itimes (b, i)), 0))))
   < cdr (v&c-apply$ ('iterk,
                     list (cons (a, 0),
                             cons (b, 0),
                             cons (c, 0),
                             cons (d, 0),
                             cons (iplus (1, i), 0),
                             cons (x, 0))))))

```

THEOREM: fix-int-x=x-when-0<x
 ilessp (0, x) → (fix-int (x) = x)

THEOREM: iterk_1_x-ib_cost<cost-iterk_i+1_x-step-1
 (ilessp (0, i))

$$\begin{aligned}
& \wedge \text{v\&c-apply}\$ ('iterk, \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{cons (c, 0),} \\
& \quad \quad \text{cons (d, 0),} \\
& \quad \quad \text{cons (iplus (1, i), 0),} \\
& \quad \quad \text{cons (x, 0))})) \\
\rightarrow & \text{ (cdr (v\&c-apply}\$ ('iterk, \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{cons (c, 0),} \\
& \quad \quad \text{cons (d, 0),} \\
& \quad \quad '(1 . 0),} \\
& \quad \quad \text{v\&c-apply}\$ ('iterk, \\
& \quad \quad \quad \text{list (cons (a, 0),} \\
& \quad \quad \quad \quad \text{cons (b, 0),} \\
& \quad \quad \quad \quad \text{cons (c, 0),} \\
& \quad \quad \quad \quad \text{cons (d, 0),} \\
& \quad \quad \quad \quad \text{cons (i, 0),} \\
& \quad \quad \quad \quad \text{cons (x, 0))}))))) \\
& < \text{cdr (v\&c-apply}\$ ('iterk, \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{cons (c, 0),} \\
& \quad \quad \text{cons (d, 0),} \\
& \quad \quad \text{cons (iplus (1, i), 0),} \\
& \quad \quad \text{cons (x, 0))}))
\end{aligned}$$

THEOREM: iterk_i_x-iff-iterk_1_x-ib-when-0<b&1<c&0<d&i>0&i<>1
(ilessp (0, b) \wedge ilessp (1, c) \wedge ilessp (0, d) \wedge ilessp (0, i) \wedge (i \neq 1))

$$\begin{aligned}
\rightarrow & \text{ (v\&c-apply}\$ ('iterk, \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{cons (c, 0),} \\
& \quad \quad \text{cons (d, 0),} \\
& \quad \quad \text{cons (i, 0),} \\
& \quad \quad \text{cons (x, 0))})) \\
\leftrightarrow & \text{ v\&c-apply}\$ ('iterk, \\
& \quad \text{list (cons (a, 0),} \\
& \quad \quad \text{cons (b, 0),} \\
& \quad \quad \text{cons (c, 0),} \\
& \quad \quad \text{cons (d, 0),} \\
& \quad \quad '(1 . 0),} \\
& \quad \quad \text{cons (iplus (x, ineg (itimes (b, iplus (-1, i))), 0))}))
\end{aligned}$$

THEOREM: $\text{iterk}_i x = \text{iterk}_{i-1} x$ when $0 < b < 1 < c < 0 < d < i > 0 \wedge i < 1$
 $(\text{ilessp}(0, b) \wedge \text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(0, i) \wedge (i \neq 1))$
 $\rightarrow (\text{car}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(i, 0),$
 $\quad \quad \text{cons}(x, 0))))$
 $= \text{car}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad '(1 . 0),$
 $\quad \quad \text{cons}(\text{iplus}(x, \text{ineg}(\text{itimes}(b, \text{iplus}(-1, i))),$
 $\quad \quad \quad 0))))))$

THEOREM: $\text{iterk}_{i-1} x$ $\text{ib_cost} < \text{cost-iterk}_{i+1} x$ exists-step-2
 $(\text{ilessp}(0, b) \wedge \text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(0, i) \wedge (i \neq 1))$
 $\rightarrow (\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad '(1 . 0),$
 $\quad \quad \text{v\&c-apply}\$('iterk,$
 $\quad \quad \quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \quad \quad \text{cons}(b, 0),$
 $\quad \quad \quad \quad \text{cons}(c, 0),$
 $\quad \quad \quad \quad \text{cons}(d, 0),$
 $\quad \quad \quad \quad \text{cons}(i, 0),$
 $\quad \quad \quad \quad \text{cons}(x, 0))))))$
 $\leftrightarrow \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad '(1 . 0),$
 $\quad \quad \text{v\&c-apply}\$('iterk,$
 $\quad \quad \quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \quad \quad \text{cons}(b, 0),$
 $\quad \quad \quad \quad \text{cons}(c, 0),$
 $\quad \quad \quad \quad \text{cons}(d, 0),$

```

'(1 . 0),
cons (iplus (x,
            ineg (itimes (b,
                        iplus (-1,
                              i))))),
0))))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_i+1_x-step-2

```

(ilessp (0, b)
 ^ ilessp (1, c)
 ^ ilessp (0, d)
 ^ ilessp (0, i)
 ^ (i ≠ 1)
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       '(1 . 0),
                       v&c-apply$ ('iterk,
                                   list (cons (a, 0),
                                           cons (b, 0),
                                           cons (c, 0),
                                           cons (d, 0),
                                           cons (i, 0),
                                           cons (x, 0))))))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       '(1 . 0),
                       v&c-apply$ ('iterk,
                                   list (cons (a, 0),
                                           cons (b, 0),
                                           cons (c, 0),
                                           cons (d, 0),
                                           '(1 . 0),
                                           cons (iplus (x,
                                                         ineg (itimes (b,
                                                             iplus (-1, i))))),
                                                         0))))))
 ^ (cdr (v&c-apply$ ('iterk,
                     list (cons (a, 0),

```

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (iplus (x, ineg (itimes (b, iplus (-1, i)))), 0))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (i, 0),
cons (x, 0))))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (iplus (x,
ineg (itimes (b,
iplus (-1,
i))))),
0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (i, 0),
cons (x, 0))))))

```

THEOREM: iterk_{i+1}-exists-iff-iterk₁-iterk_i-exists

```

(ilessp (0, i) ∧ (i ≠ 1))
→ (v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (iplus (1, i), 0),
                       cons (x, 0)))
   ↔ v&c-apply$ ('iterk,
                 list (cons (a, 0),
                         cons (b, 0),
                         cons (c, 0),
                         cons (d, 0),
                         '(1 . 0),
                         v&c-apply$ ('iterk,
                                     list (cons (a, 0),
                                             cons (b, 0),
                                             cons (c, 0),
                                             cons (d, 0),
                                             cons (i, 0),
                                             cons (x, 0))))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk.i+1_x-step-1&2

```

(ilessp (0, b)
 ∧ ilessp (1, c)
 ∧ ilessp (0, d)
 ∧ ilessp (0, i)
 ∧ (i ≠ 1)
 ∧ v&c-apply$ ('iterk,
               list (cons (a, 0),
                       cons (b, 0),
                       cons (c, 0),
                       cons (d, 0),
                       cons (iplus (1, i), 0),
                       cons (x, 0)))
 ∧ (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            '(1 . 0),
                            cons (iplus (x, ineg (itimes (b, iplus (-1, i))), 0))))
      < cdr (v&c-apply$ ('iterk,
                        list (cons (a, 0),

```

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (i, 0),
cons (x, 0))))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (iplus (x,
ineg (itimes (b,
iplus (-1,
i))))),
0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (1, i), 0),
cons (x, 0))))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_i+1_x-exists-step-3

```

(ilessp (0, b)
^ illessp (1, c)
^ illessp (0, d)
^ illessp (0, i)
^ (i ≠ 1)
^ illessp (iplus (a, itimes (b, i)), iplus (x, b)))
→ (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
v&c-apply$ ('iterk,

```

```

list (cons (a, 0),
      cons (b, 0),
      cons (c, 0),
      cons (d, 0),
      '(1 . 0),
      cons (iplus (x,
                  ineg (itimes (b,
                              iplus (-1, i))),
                  0))))))
↔ v&c-apply$ ('iterk,
              list (cons (a, 0),
                    cons (b, 0),
                    cons (c, 0),
                    cons (d, 0),
                    '(1 . 0),
                    cons (iplus (x, ineg (itimes (b, i))), 0))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_i+1_x-step-3

```

(ilessp (0, b)
 ^ ilessp (1, c)
 ^ ilessp (0, d)
 ^ ilessp (0, i)
 ^ (i ≠ 1)
 ^ ilessp (iplus (a, itimes (b, i)), iplus (x, b))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     '(1 . 0),
                     cons (iplus (x, ineg (itimes (b, i))), 0)))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     '(1 . 0),
                     v&c-apply$ ('iterk,
                                   list (cons (a, 0),
                                           cons (b, 0),
                                           cons (c, 0),
                                           cons (d, 0),
                                           '(1 . 0),
                                           cons (iplus (x,
                                                         ineg (itimes (b, i))),
                                                         0))))))

```

```

                                ineg (itimes (b,
                                iplus (-1, i))),
                                0))))))
→ (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                '(1 . 0),
                                cons (iplus (x, ineg (itimes (b, i))), 0))))
    <  cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                                cons (b, 0),
                                cons (c, 0),
                                cons (d, 0),
                                '(1 . 0),
                                v&c-apply$ ('iterk,
                                            list (cons (a, 0),
                                                    cons (b, 0),
                                                    cons (c, 0),
                                                    cons (d, 0),
                                                    '(1 . 0),
                                                    cons (iplus (x,
                                                                ineg (itimes (b,
                                                                iplus (-1,
                                                                i))),
                                                                0))))))
                                0))))))

```

THEOREM: $\text{iterk}_{i+1} _x \text{-iff-iterk}_{i-1} _x \text{-ib-when-} 0 < b & 1 < c & 0 < d \& i > 0$
 $(\text{ilessp}(0, b) \wedge \text{ilessp}(1, c) \wedge \text{ilessp}(0, d) \wedge \text{ilessp}(0, i))$

```

→ (v&c-apply$ ('iterk,
                list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            cons (iplus (1, i), 0),
                            cons (x, 0)))
    ↔ v&c-apply$ ('iterk,
                list (cons (a, 0),
                            cons (b, 0),
                            cons (c, 0),
                            cons (d, 0),
                            '(1 . 0),
                            cons (iplus (x, ineg (itimes (b, i))), 0))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_i+1_x-step-1&2&3

```

(ilessp (0, b)
 ^ ilessp (1, c)
 ^ ilessp (0, d)
 ^ ilessp (0, i)
 ^ (i ≠ 1)
 ^ iplus (iplus (a, itimes (b, i)), iplus (x, b))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     cons (iplus (1, i), 0),
                     cons (x, 0)))
 ^ (cdr (v&c-apply$ ('iterk,
                    list (cons (a, 0),
                          cons (b, 0),
                          cons (c, 0),
                          cons (d, 0),
                          '(1 . 0),
                          cons (iplus (x, ineg (itimes (b, iplus (-1, i))), 0))))
      <  cdr (v&c-apply$ ('iterk,
                        list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              cons (i, 0),
                              cons (x, 0))))))
 →  (cdr (v&c-apply$ ('iterk,
                     list (cons (a, 0),
                           cons (b, 0),
                           cons (c, 0),
                           cons (d, 0),
                           '(1 . 0),
                           cons (iplus (x, ineg (itimes (b, i))), 0))))
      <  cdr (v&c-apply$ ('iterk,
                        list (cons (a, 0),
                              cons (b, 0),
                              cons (c, 0),
                              cons (d, 0),
                              cons (iplus (1, i), 0),
                              cons (x, 0))))))

```

THEOREM: a+bi < x+b-when-a+bi < b+b+x&0 < b

$$\begin{aligned}
& (\text{ilessp}(0, b) \wedge \text{ilessp}(\text{iplus}(a, \text{itimes}(b, i)), \text{iplus}(x, b))) \\
\rightarrow & \text{ilessp}(\text{iplus}(a, \text{itimes}(b, i)), \text{iplus}(b, \text{iplus}(b, x)))
\end{aligned}$$

THEOREM: iterk-e=i-exists-when-iterk-e=i+1-exists

$$\begin{aligned}
& (\text{ilessp}(0, i) \\
& \wedge (i \neq 1) \\
& \wedge \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(1, i), 0), \\
& \quad \quad \text{cons}(x, 0))) \\
\rightarrow & \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(i, 0), \\
& \quad \quad \text{cons}(x, 0)))
\end{aligned}$$

THEOREM: iterk_1_x-ib_cost<cost-iterk_i+1_x-induction-step

$$\begin{aligned}
& (\text{ilessp}(0, b) \\
& \wedge \text{ilessp}(1, c) \\
& \wedge \text{ilessp}(0, d) \\
& \wedge \text{ilessp}(0, i) \\
& \wedge (i \neq 1) \\
& \wedge \text{ilessp}(\text{iplus}(a, \text{itimes}(b, i)), \text{iplus}(x, b)) \\
& \wedge \text{v\&c-apply}\$('iterk, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \text{cons}(c, 0), \\
& \quad \quad \text{cons}(d, 0), \\
& \quad \quad \text{cons}(\text{iplus}(1, i), 0), \\
& \quad \quad \text{cons}(x, 0))) \\
& \wedge ((\text{ilessp}(0, b) \\
& \quad \wedge \text{ilessp}(1, c) \\
& \quad \wedge \text{ilessp}(0, d) \\
& \quad \wedge \text{ilessp}(0, \text{iplus}(-1, i)) \\
& \quad \wedge \text{ilessp}(\text{iplus}(a, \text{itimes}(b, \text{iplus}(-1, i))), \text{iplus}(x, b)) \\
& \quad \wedge \text{v\&c-apply}\$('iterk, \\
& \quad \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \quad \text{cons}(b, 0), \\
& \quad \quad \quad \text{cons}(c, 0),
\end{aligned}$$

```

cons (d, 0),
cons (iplus (1, iplus (-1, i)), 0),
cons (x, 0)))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (idifference (x,
itimes (b, iplus (-1, i))),
0))))))
< (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (1, iplus (-1, i)), 0),
cons (x, 0))))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (idifference (x, itimes (b, i)), 0))))))
< (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (1, i), 0),
cons (x, 0))))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_i+1_x
(ilessp (0, b)
^ ilessp (1, c)
^ ilessp (0, d)
^ ilessp (0, i)
^ ilessp (iplus (a, itimes (b, i)), iplus (x, b))
^ v&c-apply\$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),

```

cons (d, 0),
cons (iplus (1, i), 0),
cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (idifference (x, itimes (b, i)), 0))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (iplus (1, i), 0),
cons (x, 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 22. A generalization of Lemma 21.

```

```

; Assume 0 < b, 1 < c, 0 < d, 0 < i, a+ib < x+b,
; i < e, and IterK( a,b,c,d,e,x ) exists.

```

```

; Then IterK( a,b,c,d,1,x-ib ) exists

```

```

; and

```

```

; cost[ IterK( a,b,c,d,1,x-ib ) ]
; <
; cost[ IterK( a,b,c,d,e,x ) ]

```

```

; Proof. Lemma 2 and Lemma 21.

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_e_x-case-1

```

(ilessp (0, b)
^ illessp (1, c)
^ illessp (0, d)
^ illessp (0, i)
^ illessp (iplus (1, i), e)
^ illessp (iplus (a, itimes (b, i)), iplus (x, b))
^ v&c-apply$ ('iterk,
list (cons (a, 0),

```

```

cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (idifference (x, itimes (b, i), 0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0))))))

```

THEOREM: iterk_1_x-ib_cost < cost-iterk_e_x-case-2

```

(ilessp (0, b)
^ ilessp (1, c)
^ ilessp (0, d)
^ ilessp (0, i)
^ (iplus (1, i) = e)
^ ilessp (iplus (a, itimes (b, i)), iplus (x, b))
^ v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (idifference (x, itimes (b, i), 0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),

```

```

cons (c, 0),
cons (d, 0),
cons (e, 0),
cons (x, 0))))))

```

THEOREM: $e=i+1$ -or- $e>i+1$ -when- $e>i>0$
 $(\text{ilessp}(0, i) \wedge \text{ilessp}(i, e))$
 $\rightarrow ((e = \text{iplus}(1, i)) \vee \text{ilessp}(\text{iplus}(1, i), e))$

THEOREM: $\text{iterk_1_x_ib_cost} < \text{cost_iterk_e_x}$
 $(\text{ilessp}(0, b)$
 $\wedge \text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge \text{ilessp}(0, i)$
 $\wedge \text{ilessp}(i, e)$
 $\wedge \text{ilessp}(\text{iplus}(a, \text{itimes}(b, i)), \text{iplus}(x, b))$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e, 0),$
 $\quad \quad \text{cons}(x, 0))))$
 $\rightarrow (\text{cdr}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad '(1 . 0),$
 $\quad \quad \text{cons}(\text{idifference}(x, \text{itimes}(b, i)), 0))))$
 $< \text{cdr}(\text{v\&c-apply}\$('iterk,$
 $\quad \text{list}(\text{cons}(a, 0),$
 $\quad \quad \text{cons}(b, 0),$
 $\quad \quad \text{cons}(c, 0),$
 $\quad \quad \text{cons}(d, 0),$
 $\quad \quad \text{cons}(e, 0),$
 $\quad \quad \text{cons}(x, 0))))))$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define the function J( a,b,x ) recursively,
; so that whenever b > 0, J( a,b,x ) is the smallest
; nonnegative integer i such that x <= a + ib.

```

DEFINITION:

how-far-above-a (a, x)
= **if** \neg **ilessp** (a, x) **then** 0
else **iplus** ($x, \text{ineg}(a)$) **endif**

THEOREM: numberp_x-a_when_x>a
ilessp (a, x) \rightarrow (**iplus** ($x, \text{ineg}(a)$) $\in \mathbf{N}$)

THEOREM: x-a-b<x-a-when-0<b&a<x&a+b<x
(**ilessp** (0, b) \wedge **ilessp** (a, x) \wedge **ilessp** (**iplus** (a, b), x))
 \rightarrow (**iplus** ($x, \text{iplus}(\text{ineg}(a), \text{ineg}(b))$) $<$ **iplus** ($x, \text{ineg}(a)$))

THEOREM: a<>x-when-a<x
ilessp (a, x) \rightarrow (**fix-int** (x) \neq **fix-int** (a))

DEFINITION:
j (a, b, x)
= **if** \neg **ilessp** (0, b) **then** 0
elseif \neg **ilessp** (a, x) **then** 0
else **iplus** (1, **j** ($a, b, \text{iplus}(x, \text{ineg}(b))$)) **endif**

THEOREM: j>=0
 \neg **ilessp** (**j** (a, b, x), 0)

THEOREM: j>0-when-x>a&b>0
(**ilessp** (0, b) \wedge **ilessp** (a, x)) \rightarrow **ilessp** (0, **j** (a, b, x))

THEOREM: x<=a+jb-when-0<b
ilessp (0, b) \rightarrow (\neg **ilessp** (**iplus** ($a, \text{itimes}(b, \text{j}(a, b, x))$), x))

THEOREM: a+jb<x+b-when-0<b&a<x
(**ilessp** (0, b) \wedge **ilessp** (a, x))
 \rightarrow **ilessp** (**iplus** ($a, \text{itimes}(b, \text{j}(a, b, x))$), **iplus** (b, x))

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

THEOREM: j<c-when-0<b&a<x&c>1&a+_c-1_b>=x
(**ilessp** (0, b)
 \wedge **ilessp** (1, c)
 \wedge **ilessp** (a, x)
 \wedge (\neg **ilessp** (**iplus** ($a, \text{itimes}(b, \text{iplus}(-1, c))$), x))
 \rightarrow **ilessp** (**j** (a, b, x), c))

;;;;;;;;;;;;;
; Lemma 23. A special case of Lemma 22.

```

; Assume 0 < b, 1 < c, 0 < d, a < x, a+(c-1)b >= x,
; and IterK( a,b,c,d,c,x ) exists.

; Then IterK( a,b,c,d,1,x-J( a,b,x )b ) exists

; and

; cost[ IterK( a,b,c,d,1,x-J( a,b,x )b ) ]
; <
; cost[ IterK( a,b,c,d,c,x ) ]

; Proof. By Lemma 22 and the definition of J.

```

THEOREM: iterk_1_x-jb-exists-when-iterk_e=c_x-exists&0<b&1<c&0<d&etc

```

(ilessp (0, b)
 ^ illessp (1, c)
 ^ illessp (0, d)
 ^ illessp (a, x)
 ^ (¬ illessp (iplus (a, itimes (b, iplus (-1, c))), x))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     cons (c, 0),
                     cons (x, 0))))
→ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),
                     cons (c, 0),
                     cons (d, 0),
                     '(1 . 0),
                     cons (idifference (x, itimes (b, j (a, b, x))), 0)))

```

THEOREM: iterk_1_x-jb_cost<cost-iterk_e=c_x

```

(ilessp (0, b)
 ^ illessp (1, c)
 ^ illessp (0, d)
 ^ illessp (a, x)
 ^ (¬ illessp (iplus (a, itimes (b, iplus (-1, c))), x))
 ^ v&c-apply$ ('iterk,
               list (cons (a, 0),
                     cons (b, 0),

```

```

cons (c, 0),
cons (d, 0),
cons (c, 0),
cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (idifference (x, itimes (b, j (a, b, x))), 0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
cons (c, 0),
cons (x, 0))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define the function L( a,b,d,x ) to be the following
; combination of x, b, d, and the functions J and N:
;
; x + N( a,d,x ) d - J( a,b,x+N( a,d,x )d ) b.

```

DEFINITION:

```

l(a, b, d, x)
= iplus(iplus(x, itimes(d, n(a, d, x))),
      ineg(itimes(b, j(a, b, iplus(x, itimes(d, n(a, d, x)))))))

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 24. Computing K with L( a,b,d,x ) is cheaper
; than computing K with x.

```

```

; Assume 0 < b, 1 < c, 0 < d, a >= x, (c-1)b >= d,
; and IterK( a,b,c,d,1,x ) exists.

```

```

; Then IterK( a,b,c,d,1,L( a,b,d,x ) ) exists

```

```

; and

```

```

; cost[ IterK( a,b,c,d,1,L( a,b,d,x ) ) ]
; <
; cost[ IterK( a,b,c,d,1,x ) ]

```


; Proof. By Lemma 19 and Lemma 23.

THEOREM: $x \geq z$ -when- $x \geq y$ & $y \geq z$
 $((\neg \text{ilessp}(x, y)) \wedge (\neg \text{ilessp}(y, z))) \rightarrow (\neg \text{ilessp}(x, z))$

THEOREM: $a + c - 1 \cdot b \geq x + nd$ -when- $0 < d$ & $a \geq x$ & $c - 1 \cdot b \geq d$
 $(\text{ilessp}(0, d)$
 $\wedge (\neg \text{ilessp}(a, x))$
 $\wedge (\neg \text{ilessp}(\text{itimes}(b, \text{iplus}(-1, c)), d))$
 $\rightarrow (\neg \text{ilessp}(\text{iplus}(a, \text{itimes}(b, \text{iplus}(-1, c))),$
 $\text{iplus}(x, \text{itimes}(d, n(a, d, x))))$

THEOREM: iterk-l-exists -when- iterk-x-exists & $0 < b & 1 < c & 0 < d$ & etc
 $(\text{ilessp}(0, b)$
 $\wedge \text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge (\neg \text{ilessp}(a, x))$
 $\wedge (\neg \text{ilessp}(\text{itimes}(b, \text{iplus}(-1, c)), d))$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$
 $\text{cons}(d, 0),$
 $'(1 . 0),$
 $\text{cons}(x, 0)))$
 $\rightarrow \text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$
 $\text{cons}(d, 0),$
 $'(1 . 0),$
 $\text{cons}(l(a, b, d, x), 0)))$

THEOREM: $\text{iterk-l-cost} < \text{cost-iterk-x}$ & $0 < b & 1 < c & 0 < d$ & etc
 $(\text{ilessp}(0, b)$
 $\wedge \text{ilessp}(1, c)$
 $\wedge \text{ilessp}(0, d)$
 $\wedge (\neg \text{ilessp}(a, x))$
 $\wedge (\neg \text{ilessp}(\text{itimes}(b, \text{iplus}(-1, c)), d))$
 $\wedge \text{v\&c-apply}\$('iterk,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$

```

cons (d, 0),
'(1 . 0),
cons (x, 0))))
→ (cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (l(a, b, d, x), 0))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
'(1 . 0),
cons (x, 0))))))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Define the function IterL( i,a,b,d,x ) which iterates
; the function L i times.

```

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

DEFINITION:
 $\text{iterl}(i, a, b, d, x)$
= **if** $\text{ilessp}(1, i)$ **then** $l(a, b, d, \text{iterl}(\text{iplus}(-1, i), a, b, d, x))$
else $l(a, b, d, x)$ **endif**

THEOREM: $l \leq a$
 $\text{ilessp}(0, b) \rightarrow (\neg \text{ilessp}(a, l(a, b, d, x)))$

EVENT: Disable l.

THEOREM: $\text{iterl} \leq a$
 $\text{ilessp}(0, b) \rightarrow (\neg \text{ilessp}(a, \text{iterl}(i, a, b, d, x)))$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Lemma 25. The cost of computing K is unbounded
; when (c-1)b >= d

; Assume 0 < b, 1 < c, 0 < d, a >= x, (c-1)b >= d,
; i > 0, and IterK( a,b,c,d,1,x ) exists.

```

```

; Then IterK( a,b,c,d,1,IterL( i,a,b,d,x ) ) exists

; and

; cost[ IterK( a,b,c,d,1,IterL( i,a,b,d,x ) ) ] + i-1
; <
; cost[ IterK( a,b,c,d,1,x ) ]

; Proof. By induction on i using Lemma 24.

```

THEOREM: iterk-iterl-exists-when-iterk-x-exists&0<b&1<c&0<d&etc

```

(ilessp(0, b)
 ^ illessp(1, c)
 ^ illessp(0, d)
 ^ illessp(0, i)
 ^ (¬ illessp(a, x))
 ^ (¬ illessp(itimes(b, iplus(-1, c)), d))
 ^ v&c-apply$( 'iterk,
                list(cons(a, 0),
                      cons(b, 0),
                      cons(c, 0),
                      cons(d, 0),
                      '(1 . 0),
                      cons(x, 0))))
 → v&c-apply$( 'iterk,
                list(cons(a, 0),
                      cons(b, 0),
                      cons(c, 0),
                      cons(d, 0),
                      '(1 . 0),
                      cons(iterl(i, a, b, d, x), 0)))

```

THEOREM: fix-int-cost=cost

```
fix-int(cdr(v&c-apply$(fn, args))) = cdr(v&c-apply$(fn, args))
```

THEOREM: x+i-1<z-when-x<y&-1-1+i+y<z

```
((x < y) ^ (iplus(-1, iplus(-1, iplus(i, y))) < z))
 → (iplus(-1, iplus(i, x)) < z)

```

THEOREM: iterk-iterl-cost+i-1<cost-iterk-x&0<b&1<c&0<d-etc-induction-step

```

(ilessp(0, i)
 ^ (i ≠ 1)
 ^ (iplus(-1,

```

```

    iplus(-1,
      iplus(i,
        cdr(v&c-apply$('iterk,
          list(cons(a, 0),
              cons(b, 0),
              cons(c, 0),
              cons(d, 0),
              '(1 . 0),
              cons(iterl(iplus(-1, i),
                a,
                b,
                d,
                x),
                0))))))
      < cdr(v&c-apply$('iterk,
        list(cons(a, 0),
            cons(b, 0),
            cons(c, 0),
            cons(d, 0),
            '(1 . 0),
            cons(x, 0))))
    ^ ilessp(0, b)
    ^ ilessp(1, c)
    ^ ilessp(0, d)
    ^ (¬ ilessp(a, x))
    ^ (¬ ilessp(itimes(b, c), iplus(b, d)))
    ^ v&c-apply$('iterk,
      list(cons(a, 0),
          cons(b, 0),
          cons(c, 0),
          cons(d, 0),
          '(1 . 0),
          cons(x, 0)))
  → (iplus(-1,
    iplus(i,
      cdr(v&c-apply$('iterk,
        list(cons(a, 0),
            cons(b, 0),
            cons(c, 0),
            cons(d, 0),
            '(1 . 0),
            cons(l(a,
              b,
              d,

```

```

iterl (iplus (-1, i), a, b, d, x),
0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
' (1 . 0),
cons (x, 0))))))

```

THEOREM: iterk-iterl-cost+i-1<cost-iterk-x&0<b&1<c&0<d-etc

```

(ilessp (0, b)
^ ilessp (1, c)
^ ilessp (0, d)
^ ilessp (0, i)
^ (¬ ilessp (a, x))
^ (¬ ilessp (itimes (b, iplus (-1, c)), d))
^ v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
' (1 . 0),
cons (x, 0))))
→ (iplus (-1,
iplus (i,
cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
' (1 . 0),
cons (iterl (i, a, b, d, x), 0))))))
< cdr (v&c-apply$ ('iterk,
list (cons (a, 0),
cons (b, 0),
cons (c, 0),
cons (d, 0),
' (1 . 0),
cons (x, 0))))))

```

THEOREM: i<y-when-0<x&-1+i+x<y
((0 < x) ∧ (iplus (-1, iplus (i, x)) < y)) → (i < y)

EVENT: Disable iterl.

THEOREM: iterk-cost-is-unbounded-when- $(c-1)b \geq d$ & $x \leq a$ & etc

```
(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ ilessp (0, i)
  ^ (¬ ilessp (a, x))
  ^ (¬ ilessp (itimes (b, iplus (-1, c)), d))
  ^ v&c-apply$ ('iterk,
                list (cons (a, 0),
                      cons (b, 0),
                      cons (c, 0),
                      cons (d, 0),
                      '(1 . 0),
                      cons (x, 0))))
→ (i < cdr (v&c-apply$ ('iterk,
                       list (cons (a, 0),
                             cons (b, 0),
                             cons (c, 0),
                             cons (d, 0),
                             '(1 . 0),
                             cons (x, 0))))))
```

THEOREM: iterk-does-not-exist-when- $(c-1)b \geq d$ & $x \leq a$ & etc

```
(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ (¬ ilessp (a, x))
  ^ (¬ ilessp (itimes (b, iplus (-1, c)), d)))
→ (¬ v&c-apply$ ('iterk,
                  list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        '(1 . 0),
                        cons (x, 0))))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; The next two events are versions of Part 9 of the
; Main Theorem given above in the introduction.
```

```
; 9. If  $x \leq a$ ,  $d > 0$ ,  $c > 1$ ,  $b > 0$ , and
;     $(c-1)b \geq d$ , then  $K(a, b, c, d, x)$  does
;    not exist.
```

; Proof. By Lemma 25 and the definition of K.

THEOREM: k-does-not-exist-when- $c-1 \leq d$ & $x \leq a$ etc

```
(ilessp (0, b)
  ^ ilessp (1, c)
  ^ ilessp (0, d)
  ^ (¬ ilessp (a, x))
  ^ (¬ ilessp (itimes (b, iplus (-1, c)), d)))
→ (¬ v&c-apply$ ('k,
                list (cons (a, 0),
                        cons (b, 0),
                        cons (c, 0),
                        cons (d, 0),
                        cons (x, 0))))
```

THEOREM: k-does-not-halt-when- $c-1 \leq d$ & $x \leq a$ etc

```
(vc-a
  ^ vc-b
  ^ vc-c
  ^ vc-d
  ^ vc-x
  ^ ilessp (0, car (vc-b))
  ^ ilessp (1, car (vc-c))
  ^ ilessp (0, car (vc-d))
  ^ (¬ ilessp (car (vc-a), car (vc-x)))
  ^ (¬ ilessp (itimes (car (vc-b), iplus (-1, car (vc-c))), car (vc-d))))
→ (¬ v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Here a quantifier is used to define the concept that K
; is a total function of x with parameters a,b,c, and d.
```

```
; This is the only place in this file of events that
; an explicit quantifier is used.
```

DEFINITION:

```
k-is-total (vc-a, vc-b, vc-c, vc-d)
↔  $\forall vc-x (vc-x \rightarrow v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))$ 
```

THEOREM: k-is-total-suff

```
(vc-x (vc-a, vc-b, vc-c, vc-d)
  → v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x (vc-a, vc-b, vc-c, vc-d))))
→ k-is-total (vc-a, vc-b, vc-c, vc-d)
```

THEOREM: k-is-total-necc
 $(\neg (vc-x \rightarrow v\&c\text{-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x))))$
 $\rightarrow (\neg \text{k-is-total}(vc-a, vc-b, vc-c, vc-d))$

EVENT: Disable k.

EVENT: Disable k-is-total.

THEOREM: knuth-theorem-if-part-case-1
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge \text{ilessp}(0, \text{car}(vc-b))$
 $\wedge \text{ilessp}(1, \text{car}(vc-c))$
 $\wedge \text{ilessp}(0, \text{car}(vc-d)))$
 $\rightarrow (\text{ilessp}(\text{itimes}(\text{car}(vc-b), \text{iplus}(-1, \text{car}(vc-c))), \text{car}(vc-d))$
 $\rightarrow \text{k-is-total}(vc-a, vc-b, vc-c, vc-d))$

THEOREM: knuth-theorem-if-part-case-2
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge \text{ilessp}(0, \text{car}(vc-b))$
 $\wedge (\neg \text{ilessp}(1, \text{car}(vc-c)))$
 $\wedge \text{ilessp}(0, \text{car}(vc-d)))$
 $\rightarrow (\text{ilessp}(\text{itimes}(\text{car}(vc-b), \text{iplus}(-1, \text{car}(vc-c))), \text{car}(vc-d))$
 $\rightarrow \text{k-is-total}(vc-a, vc-b, vc-c, vc-d))$

THEOREM: knuth-theorem-if-part
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge \text{ilessp}(0, \text{car}(vc-b))$
 $\wedge \text{ilessp}(0, \text{car}(vc-d)))$
 $\rightarrow (\text{ilessp}(\text{itimes}(\text{car}(vc-b), \text{iplus}(-1, \text{car}(vc-c))), \text{car}(vc-d))$
 $\rightarrow \text{k-is-total}(vc-a, vc-b, vc-c, vc-d))$

THEOREM: knuth-theorem-only-if-part-case-1
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$


```

 $\wedge$   $vc-d$ 
 $\wedge$   $i\text{lessp}(0, \text{car}(vc-b))$ 
 $\wedge$   $i\text{lessp}(1, \text{car}(vc-c))$ 
 $\wedge$   $i\text{lessp}(0, \text{car}(vc-d))$ 
 $\rightarrow$   $(k\text{-is-total}(vc-a, vc-b, vc-c, vc-d)$ 
       $\rightarrow i\text{lessp}(i\text{times}(\text{car}(vc-b), i\text{plus}(-1, \text{car}(vc-c))), \text{car}(vc-d)))$ 

```

THEOREM: knuth-theorem-only-if-part-case-2
 $(i\text{lessp}(0, b) \wedge (\neg i\text{lessp}(1, c)) \wedge i\text{lessp}(0, d))$
 $\rightarrow i\text{lessp}(i\text{times}(b, i\text{plus}(-1, c)), d)$

THEOREM: knuth-theorem-only-if-part
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge i\text{lessp}(0, \text{car}(vc-b))$
 $\wedge i\text{lessp}(0, \text{car}(vc-d))$
 $\rightarrow (k\text{-is-total}(vc-a, vc-b, vc-c, vc-d)$
 $\rightarrow i\text{lessp}(i\text{times}(\text{car}(vc-b), i\text{plus}(-1, \text{car}(vc-c))), \text{car}(vc-d)))$

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Here is a version of Knuth's theorem:

```

```

; 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.

```

THEOREM: knuth-theorem
 $(vc-a$
 $\wedge vc-b$
 $\wedge vc-c$
 $\wedge vc-d$
 $\wedge i\text{lessp}(0, \text{car}(vc-b))$
 $\wedge i\text{lessp}(0, \text{car}(vc-d))$
 $\rightarrow (k\text{-is-total}(vc-a, vc-b, vc-c, vc-d)$
 $\leftrightarrow i\text{lessp}(i\text{times}(\text{car}(vc-b), i\text{plus}(-1, \text{car}(vc-c))), \text{car}(vc-d)))$

THEOREM: k-value-when-a<x
 $i\text{lessp}(a, x)$
 $\rightarrow (\text{car}(\text{v\&c-apply}\$('k,$
 $\text{list}(\text{cons}(a, 0),$
 $\text{cons}(b, 0),$
 $\text{cons}(c, 0),$

$$\begin{aligned}
& \text{cons}(d, 0), \\
& \text{cons}(x, 0))) \\
= & \text{idifference}(x, b)
\end{aligned}$$

THEOREM: k-value-when-a<x-version-2

$$\begin{aligned}
& (vc-a \wedge vc-b \wedge vc-c \wedge vc-d \wedge vc-x \wedge \text{ilessp}(\text{car}(vc-a), \text{car}(vc-x))) \\
\rightarrow & (\text{car}(\text{v\&c-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x))) \\
& = \text{idifference}(\text{car}(vc-x), \text{car}(vc-b)))
\end{aligned}$$

THEOREM: knuth-theorem-part-2-case-1

$$\begin{aligned}
& (vc-a \\
& \wedge vc-b \\
& \wedge vc-c \\
& \wedge vc-d \\
& \wedge vc-x \\
& \wedge \text{ilessp}(0, \text{car}(vc-b)) \\
& \wedge \text{ilessp}(1, \text{car}(vc-c)) \\
& \wedge \text{ilessp}(0, \text{car}(vc-d))) \\
\rightarrow & (\text{car}(\text{v\&c-apply}\$('k, \text{list}(vc-a, vc-b, vc-c, vc-d, vc-x))) \\
& = \text{if } \text{ilessp}(\text{car}(vc-a), \text{car}(vc-x)) \\
& \quad \text{then } \text{idifference}(\text{car}(vc-x), \text{car}(vc-b)) \\
& \quad \text{else } \text{car}(\text{v\&c-apply}\$('k, \\
& \quad \quad \text{list}(vc-a, \\
& \quad \quad \quad vc-b, \\
& \quad \quad \quad vc-c, \\
& \quad \quad \quad vc-d, \\
& \quad \quad \text{cons}(\text{iplus}(\text{car}(vc-x), \\
& \quad \quad \quad \text{idifference}(\text{car}(vc-d), \\
& \quad \quad \quad \quad \text{itimes}(\text{car}(vc-b), \\
& \quad \quad \quad \quad \quad \text{iplus}(-1, \\
& \quad \quad \quad \quad \quad \quad \text{car}(vc-c))))), \\
& \quad \quad \quad \text{cost})))) \text{endif})
\end{aligned}$$

THEOREM: k-value=body-value-when-a>=x&c=1

$$\begin{aligned}
& (\neg \text{ilessp}(a, x)) \\
\rightarrow & (\text{car}(\text{v\&c-apply}\$('k, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \quad '(1 . 0), \\
& \quad \quad \quad \text{cons}(d, 0), \\
& \quad \quad \quad \text{cons}(x, 0)))) \\
= & \text{car}(\text{v\&c-apply}\$('k, \\
& \quad \text{list}(\text{cons}(a, 0), \\
& \quad \quad \text{cons}(b, 0), \\
& \quad \quad \quad '(1 . 0),
\end{aligned}$$

```

cons (d, 0),
cons (iplus (x, d), 0))))))

```

THEOREM: k-value=body-value-when-a>=x&c=1-version-2

```

(vc-a
  ^  vc-b
  ^  vc-c
  ^  vc-d
  ^  vc-x
  ^  (car (vc-c) = 1)
  ^  (¬ ilessp (car (vc-a), car (vc-x))))
→ (car (v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))
   = car (v&c-apply$ ('k,
                      list (vc-a,
                          vc-b,
                          vc-c,
                          vc-d,
                          cons (iplus (car (vc-x), car (vc-d), cost))))))

```

THEOREM: knuth-theorem-part-2-case-2

```

(vc-a ^ vc-b ^ vc-c ^ vc-d ^ vc-x ^ (car (vc-c) = 1))
→ (car (v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))
   = if ilessp (car (vc-a), car (vc-x))
     then idifference (car (vc-x), car (vc-b))
     else car (v&c-apply$ ('k,
                          list (vc-a,
                              vc-b,
                              vc-c,
                              vc-d,
                              cons (iplus (car (vc-x),
                                           idifference (car (vc-d),
                                                         itimes (car (vc-b),
                                                         iplus (-1,
                                                         car (vc-c))))),
                                  cost)))))) endif)

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Here is a version of that part of Knuth's theorem
; which gives the circumstances when K satisfies a
; simpler recurrence.

```

```

; Theorem. 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 ).
```

THEOREM: knuth-theorem-part-2

```
(vc-a
  ^ vc-b
  ^ vc-c
  ^ vc-d
  ^ vc-x
  ^ ilssp (0, car (vc-b))
  ^ ilssp (0, car (vc-c))
  ^ ilssp (0, car (vc-d)))
→ (car (v&c-apply$ ('k, list (vc-a, vc-b, vc-c, vc-d, vc-x)))
   =  if ilssp (car (vc-a), car (vc-x))
      then idifference (car (vc-x), car (vc-b))
      else car (v&c-apply$ ('k,
                          list (vc-a,
                              vc-b,
                              vc-c,
                              vc-d,
                              cons (iplus (car (vc-x),
                                          idifference (car (vc-d),
                                                         itimes (car (vc-b),
                                                                iplus (-1,
                                                                    car (vc-c))))),
                                  cost)))) endif)
```

Index

- a \langle x-when-a \langle x, 102
- a \langle -b+x-when-a \langle x&b \leq 0, 36
- a \langle x+nd, 51
- a \rangle =x-when-a+d \rangle =x+d, 45
- a+_c-1_b \rangle =x+nd-when-0 \langle d&a \rangle =x&_c-1_b \rangle =d, 105
- a+b \langle x-when-a \langle x&b \leq 0, 36
- a+bi \langle x+b-when-a+bi \langle b+b+x&0 \langle b, 97
- a+d \rangle =x+_n-1_d-when-a+d \rangle =x+_nd&0 \langle d&0 \langle n, 50
- a+d \rangle =x+nd-when-d \rangle 0&a \rangle =x, 51
- a+d \rangle =x-b+nd-when-b \rangle 0&a+d \rangle =x+nd&d \rangle 0, 64
- a+jb \langle x+b-when-0 \langle b&a \langle x, 102
- args-exist-when-fn-exists, 12

- cost \rangle 0-if-fn-exists, 6
- cost \rangle cost-of-args, 27
- cost \rangle member, 28
- cost-base-step-when-x \leq a&d \leq 0&c \leq 1, 23
- cost-induction-step-when-x \leq a&d \leq 0&c \leq 1, 22
- &d \leq 0&c \leq 1, 31
- cost-is-a-numberp, 6
- cost-iterk-at-e1 \langle cost-iterk-at-e2-when-0 \langle e1 \langle e2, 28
- e2-when-0 \langle e1 \langle e2-basic-step, 28
- count-sub1-x \langle count-x-when-x \langle 0&x \langle 1, 21
- count-sub1-y \langle count-y-when-0 \langle x \langle y, 26
- count_x-1 \langle count_x-when-1 \langle x, 106

- e+n_c-1_>1-when-1 \langle c&1 \langle e&0 \langle n, 62
- e-i=1-when-1+i \rangle =e&i \langle e, 86
- e=i+1-or-e \rangle i+1-when-e \rangle i \rangle 0, 101
- eq-args-cost-depends-on-cost-of-args, 10
- eq-args-give-eq-existence, 10
- eq-args-give-eq-values, 10

- fix-int, 53, 87, 102, 107
- fix-int-cost=cost, 107
- fix-int-x=x-when-0 \langle x, 87
- fix-int-x=x-when-x \rangle 1, 53
- forall, 111

- how-far-above-a, 101, 102

- i \langle y-when-0 \langle x&-1+i+x \langle y, 109
- idifference, 10–17, 36, 60–81, 86, 87, 98–101, 103, 104, 114–116
- ilessp, 8, 9, 11–40, 44–62, 64–90, 92–103, 105–116
- ilessp-0-add1-x, 25
- induct-hint-1-k-measure, 77
- induct-hint-2-k-measure, 84
- induct-hint-cost-hom, 26
- induct-hint-k-measure, 34
- induct-hint-positive-int, 21
- ineg, 34, 36, 77, 86, 88–96, 102, 104
- iplus, 19, 21–24, 26–32, 34, 36–70, 72–116
- iplus_1-1_x=x-when_x \rangle 0, 37
- iterk, 4
- iterk-at-e1-exists-when-iterk-at-e2-exists&0 \langle e1 \langle e2, 27
- e2-exists&0 \langle e1 \langle e2-basic-step, 27
- iterk-cost \rangle body-cost, 7
- iterk-cost \rangle body-cost-when-e \leq 1&a \rangle =x, 18
- =1&a \rangle =x-version-2, 19
- iterk-cost \rangle body-cost-when-e \leq 1-version-2, 13
- 1, 9
- iterk-cost-is-unbounded-when-_c-1_b \rangle =d&x \leq a&etc, 110

iterk-cost-is-unbounded-when-x< =a&d<=0&c<=1, 24 =a&d<=0&c>1, 32	&x+nd-b-when-1<c&0<d&a>=x&e>1, 62
iterk-does-not-exist-when-_c-1 _b>=d&x<=a&etc, 110	iterk-e&x=iterk_e+_n.c-1&x +nd-when-1<c&0<d&a>=x&e>1, 59
iterk-does-not-exist-when-x<= a&d<=0&c<=1, 25 a&d<=0&c>1, 33	iterk-exists-iff-body-exists, 4 iterk-exists-iff-body-exists-whe n-e<=1&a>=x, 17
iterk-e-1&x-b-iff-iterk_e-1+_ n.c-1&x-b+nd-when-1<c&0<d&a>=x&b>0&e>1,8 66 n.c-1&x-b+nd-when-1<c&0<d&a>=x&b>0&e>2,version-3, 14 64 n.c-1&x-b+nd-when-1<c&0<d&a>=x&b>0&e>4,version-3-case-2, 14 65	n-e<=1&a>=x-version-2, 18 n-e>1-version-2, 11 n-e>1-version-3, 14 n-e>1-version-3-case-1, 13 n-e>1-version-3-case-2, 14 iterk-exists-when-0<b&1<c& 0<d&b.c-1.<d, 81
iterk-e-1&x-b=iterk_e-1+_n. c-1&x-b+nd-when-1<c&0<d&a>=x&b>0&e>1, 67 c-1&x-b+nd-when-1<c&0<d&a>=x&b>0&e>2, 66 c-1&x-b+nd-when-1<c&0<d&a>=x&b>0&e>4, 64	iterk-exists-when-d>0&c<=1 &e=c, 34 iterk-exists-when-e<=1&a<x, 16 -version-2, 36 iterk-exists-when-e=1+n.c-1&x =x+nd&b<=0, 52
iterk-e-x-iff-iterk-e-1-iterk-1 -x, 54	iterk-exists-when-x<=a&d>0 &c<1&b<=0, 52 &c<=1&e=1, 35
iterk-e-x=iterk_e-1-iterk-1-x, 54	iterk-exists&iterk>a-when-x> rk-e=c&x+id-when-x<=a&d<=0&c<=1, a&b<=0&e>0, 36
iterk-e=1&x-cost>i-1+cost-ite rk-e=c&x+id-when-x<=a&d<=0&c<=1, 24	iterk-iterl-cost+i-1<cost-ite rk-x&0<b&1<c&0<d-etc, 109 rk-x&0<b&1<c&0<d-etc-induction- step, 107
iterk-e=1&x-iff-iterk_e=1+_n _c-1&x+nd-when-1<c&0<d&a>=x, 51	iterk-iterl-exists-when-iterk-x -exists&0<b&1<c&0<d&etc, 107 iterk-l-cost<cost-iterk-x&0< b&1<c&0<d-etc, 105 iterk-l-exists-when-iterk-x-exi sts&0<b&1<c&0<d&etc, 105
iterk-e=1&x-implies-iterk-e=c& x+id-when-x<=a&d<=0&c<=1, 22	iterk-v&c-apply\$idifference- cost, 11
iterk-e=i-exists-when-iterk-e=i +1-exists, 97	iterk-v&c-apply\$idifference-e xists, 10 iterk-v&c-apply\$idifference-v alue, 11
iterk-e&x-iff-iterk_-1+e+_n. c-1&x+nd-b-when-1<c&0<d&a>=x&e>1, 62	iterk-v&c-apply\$idifference- value, 11
iterk-e&x-iff-iterk_e+_n.c- 1&x+nd-when-1<c&0<d&a>=x&e>1, 58	iterk-value-when-e<=1&a<x, 17
iterk-e&x=iterk_-1+e+_n.c-1	

iterk-value-when- $e \leq 1 \wedge a < x$ -version-2, 36
 iterk-value=body-value, 5
 iterk-value=body-value-when- $e < 1 \wedge a > x$, 17
 iterk-value=body-value-when- $e < 1 \wedge a > x$ -version-2, 19
 iterk-value=body-value-when- $e > 1$ -version-2, 12
 iterk-value=body-value-when- $e > 1$ -version-3, 16
 iterk-value=body-value-when- $e > 1$ -version-3-case-1, 15
 iterk-value=body-value-when- $e > 1$ -version-3-case-2, 15
 iterk-value=body-value-when- $e > 1$, 8
 iterk-x+d-cost<cost-iterk-when-iterk-exists $\wedge 1 < c \wedge a > x$, 83
 iterk-x+d-cost<cost-iterk-when- $a > x \wedge 1 < c$, 30
 iterk-x+d-exists-when-iterk-exists $\wedge 1 < c \wedge a > x$, 83
 iterk-x+nd-cost<cost-iterk-when-iterk-exists $\wedge 1 < c \wedge a > x \wedge 0 < d$, 85
 iterk-x+nd-exists-when-iterk-exists $\wedge 1 < c \wedge a > x \wedge 0 < d$, 85
 iterk-x-cost $> i$ +cost-iterk-x+id-when- $x < a \wedge d < 0 \wedge c > 1 \wedge e = 1$, 32
 iterk-x-implies-iterk-x+id-when- $x < a \wedge d < 0 \wedge c > 1 \wedge e = 1$, 40
 iterk_1_x-ib-exists-when-iterk_e_x-exists $\wedge 0 < b \wedge 1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < i < e$,s-step-2, 47
 iterk_1_x-ib-exists-when-iterk_e_x-exists $\wedge 0 < b \wedge 1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < i < e$,s-step-3, 49
 iterk_1_x-ib-cost<cost-iterk_e_x, 101
 iterk_1_x-ib-cost<cost-iterk_e_x-case-1, 99
 iterk_1_x-ib-cost<cost-iterk_e_x-case-2, 100
 iterk_1_x-ib-cost<cost-iterk_e_x-base-case, 87
 iterk_1_x-ib-exists-step-2, 89
 iterk_1_x-ib-exists-step-3, 93
 iterk_1_x-ib-induction-step, 97
 iterk_1_x-ib-step-1, 87
 iterk_1_x-ib-step-1&2, 92
 iterk_1_x-ib-step-1&2&3, 96
 iterk_1_x-ib-step-2, 90
 iterk_1_x-ib-step-3, 94
 iterk_1_x-ib-exists-when-iterk_e_x-exists $\wedge 0 < b \wedge 1 < c \wedge 0 < d$ &etc, 103
 iterk_1_x-ib-cost<cost-iterk_e_x, 103
 iterk_c&d-iff-iterk_1&d-c-1_b-when- $0 < b \wedge 1 < c \wedge 0 < d \wedge b_c-1_< d$, 77
 iterk_c&d=iterk_1&d-c-1_b-when- $0 < b \wedge 1 < c \wedge 0 < d \wedge b_c-1_< d$, 78
 iterk_e+_n.c-1&x+nd-exists-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d > x + nd$, 57
 iterk_e+_n.c-1&x+nd-exists-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d > x + nd$ -step-1, 55
 iterk_e+_n.c-1&x+nd-exists-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d > x + nd$ -step-2, 56
 iterk_e+_n.c-1&x+nd-value-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d > x + nd$, 58
 iterk_e+_n.c-1&x+nd-value-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d > x + nd$ -step-1, 55
 iterk_e+_n.c-1&x+nd-value-when- $1 < c \wedge 0 < d \wedge 1 < e \wedge 0 < n \wedge a + d > x + nd$ -step-2, 57
 iterk_e=1+_n.c-1&x+nd-exists-base-step, 45
 iterk_e=1+_n.c-1&x+nd-exists-step-1, 46
 iterk_e=1+_n.c-1&x+nd-exists-step-2, 47
 iterk_e=1+_n.c-1&x+nd-exists-step-3, 49
 iterk_e=1+_n.c-1&x+nd-exists-when- $1 < c \wedge 0 < d \wedge 0 < n \wedge a + d > x + nd$, 50
 iterk_e=1+_n.c-1&x+nd-value-base-step, 46
 iterk_e=1+_n.c-1&x+nd-value-step-1, 47
 iterk_e=1+_n.c-1&x+nd-value-step-2, 48
 iterk_e=1+_n.c-1&x+nd-value-step-3, 49
 iterk_e=1+_n.c-1&x+nd-value-when- $1 < c \wedge 0 < d \wedge 0 < n \wedge a + d > x + nd$, 50
 iterk_e_x-iff-iterk_e_1_x-b-when- $0 < b \wedge 1 < c \wedge 0 < d \wedge 1 < e$, 69
 iterk_e_x-iff-iterk_e_1_x-b-when- $0 < b \wedge 1 < c \wedge 0 < d \wedge 1 < e \wedge a > x$, 68

-when-1<e&x>a, 61	k, 4
-when-1<e&x>a-step-1, 60	k-does-not-exist-when-_c-1_b>
iterk_e_x-iff-iterk_e-j_x-j	=d&x<=a&etc, 111
b-when-0<b&1<c&0<d&1<e&0<j<e,	k-does-not-exist-when-x<=a&d
71	<=0&c<=1, 25
iterk_e_x=iterk_e-1_x-b-whe	<=0&c>1, 33
n-0<b&1<c&0<d&1<e, 70	k-does-not-halt-when-_c-1_b>
n-0<b&1<c&0<d&1<e&a>=x,	=d&x<=a&etc, 111
68	k-does-not-halt-when-x<=a&d<
n-1<e&x>a, 61	=0&c<=1, 26
n-1<e&x>a-step-1, 60	=0&c>1, 33
iterk_e_x=iterk_e-j_x-jb-whe	k-exists-when-0<b&1<c&0<
n-0<b&1<c&0<d&1<e&0<j<e,	d&b_c-1_<d, 82
71	k-exists-when-x<=a&d>0&c<
iterk_i+1_exists-iff-iterk_1	=1, 35
_iterk_i_exists, 92	1&b<=0, 53
iterk_i+1_x-iff-iterk_1_x-i	k-exists-when-x>a, 20
b-when-0<b&1<c&0<d&i>0, 95	k-halts-when-0<b&1<c&0<
iterk_i_x-iff-iterk_1_x-ib-	d&b_c-1_<d, 82
when-0<b&1<c&0<d&i>0&i<>1,	k-halts-when-x<=a&d>0&c<
88	=1, 35
iterk_i_x=iterk_1_x-ib-when	1&b<=0, 53
-0<b&1<c&0<d&i>0&i<>1, 89	k-halts-when-x>a, 20
iterk_x+i-1_d_implies_ite	k-is-total, 111–113
rk_x+id-when-x<=a&d<=0&c<=1&i>0,	k-is-total-necc, 112
21	k-is-total-suff, 111
rk_x+id-when-x<=a&d<=0&c>1&i>0,	k-measure, 34, 77
29	k-measure_x+d<k-measure_x-w
iterk_x-iff-iterk_x+d_c-1_b	hen-0<d&a>=x, 34
-when-0<b&1<c&0<d&a>=x,	k-measure_x+d_c-1_b<k-mea
72	sure_x-when-_c-1_b<d&a>=x, 77
iterk_x=iterk_x+d_c-1_b-whe	k-value-when-a<x, 113
n-0<b&1<c&0<d&a>=x, 73	k-value-when-a<x-version-2, 114
iterl, 106–109	k-value=body-value-when-a>=x&
iterl<=a, 106	c=1, 114
itimes, 21–24, 29–32, 45–52, 55–59,	c=1-version-2, 115
62–67, 71–82, 85–105, 107–	k.c&d-iff-k_1&d-c-1_b-when
116	-0<b&1<c&0<d&b_c-1_<d, 78
j, 102–104	-0<b&1<c&0<d&b_c-1_<d-version-
j<c-when-0<b&a<x&c>1&	2, 79
a+_c-1_b>=x, 102	-0<b&1<c&0<d&b_c-1_<d-version-
j>0-when-x>a&b>0, 102	3, 80
j>=0, 102	k.c&d=k_1&d-c-1_b-when-0<
	b&1<c&0<d&b_c-1_<d, 79

$b < c & 0 < d \wedge b \cdot c - 1 < d$ -version-2, 80
 $b < c & 0 < d \wedge b \cdot c - 1 < d$ -version-3, 81
 $k \cdot x$ -iff- $k \cdot x + d \cdot c - 1 \cdot b$ -when- $0 < b < c & 0 < d \wedge a > = x$, 73
 $b < c & 0 < d \wedge a > = x$ -version-2, 74
 $b < c & 0 < d \wedge a > = x$ -version-3, 75
 $k \cdot x = k \cdot x + d \cdot c - 1 \cdot b$ -when- $0 < b < c & 0 < d \wedge a > = x$, 74
 $1 < c & 0 < d \wedge a > = x$ -version-2, 74
 $1 < c & 0 < d \wedge a > = x$ -version-3, 76
knuth-theorem, 113
knuth-theorem-if-part, 112
knuth-theorem-if-part-case-1, 112
knuth-theorem-if-part-case-2, 112
knuth-theorem-only-if-part, 113
knuth-theorem-only-if-part-case-1, 112
-2, 113
knuth-theorem-part-2, 116
knuth-theorem-part-2-case-1, 114
knuth-theorem-part-2-case-2, 115
 l , 104–106, 109
 $l < = a$, 106
 n , 51, 52, 59, 62–67, 84, 85, 104, 105
 $n > 0$ -when- $d > 0 \wedge x < = a$, 51
 $n > = 0$, 51
 $n - 1 \cdot c - 1 > 0$ -when- $1 < c & 0 < n \wedge n < > 1$, 46
 $n = 1$ -when- $x + d > a \wedge a > = x \wedge d > 0$, 84
 $n \cdot c - 1 + 1 = c$ -when- $n = 1 \wedge c > 1$, 45
nbr-of-iterates-sum-exists-step-1, 37
-2, 39
-3, 40
nbr-of-iterates-sum-exists-values, 44
nbr-of-iterates-sum-values-step-1, 38
-2, 40
-3, 42
 $nd = d$ -when- $n = 1 \wedge d > 0$, 45
numberp- x -a-when- $x > a$, 102
sub1- $x > 0$ -when- $x > 0 \wedge x < > 1$, 21
sum-cdrs- $> =$ member, 27
v&c-apply-if, 8
vc- x , 111
 $w - 1 + y < z \wedge x < y$ -implies- $w + x < z$, 22
 $x < = a + j \cdot b$ -when- $0 < b$, 102
 $x < y - 1$ -when- $0 < x \wedge x < y \wedge x < > y - 1$, 27
 $x < z$ -when- $x < y \wedge y < = z$, 102
 $x > 1$ -when- $x > 0 \wedge x < > 1$, 36
 $x > = z$ -when- $x > = y \wedge y > = z$, 105
 $x + -x + y = \text{fix-int-}y$, 53
 $x + 2 - 1 = x + 1$, 65
 $x + i - 1 \cdot d < = a$ -when- $x < = a \wedge d < = 0 \wedge i > 0$, 21
 $x + i - 1 < z$ -when- $x < y \wedge -1 - 1 + i + y < z$, 107
 $x + y > 1$ -when- $x > 0 \wedge x < > 1 \wedge 0 < y$, 37
 $x - a - b < x - a$ -when- $0 < b \wedge a < x \wedge a + b < x$, 102
 $y < = z$ -when- $x + y - 1 < z \wedge x > = 0 \wedge y > 0$, 24
 $y > 1$ -when- $y > x \wedge x > 0$, 26