a) precondition: n is an integer

postcondition:

1> s is an integer in [0, 10]

2> if n is a positive integer with even number of digits, (n rem 11) = (11-s); if n is a positive integer with odd number, (n rem 11) = s. if n is a negative integer with even number of digits, (n rem 11) = s; if n is a negative integer with odd number, (n rem 11) = (11-s). if n is 0, n=s=0.

3> DivisibleBy11(n) returns whether the alternating sum of the digits in n, read from left to right is equal to zero.

奇数同余 偶数11-s同余

0

负数

let P(i) = ” when the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, (notice: for every positive integer, DivisibleBy11(n) performs the while loop at least 1 time) s=0 if and only if ((m[i]…m[2]m[1] rem 11) = 0).

Base case:

1.When i = 1, s = (n rem 10) = m[1], k = (n div 10) = m[i]… m[3]m[2].

2.(m[1] rem 11) = m[1] , so (m[1] rem 11) =s. So, s=0 if and only if (m[1] rem 11)=0. 3. P(1) is true (direct prove from 1 and 2)

Constructor cases:

3.Assume for any i q, P(i) is true.

4.When the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, we have k and s.

5.k and s satisfies: {(k = m[q]…m[i+1]) AND (s = 0 if and only if (m[i]…m[2]m[1] rem 11 = 0))}. (from the algorithm DivisibleBy11(n))

6.If the i+1 th while loop does not exist, then P(i) is true for every iq.

7.Else if the i+1 th while loop does exist, then in the i+1 th while loop, after executed line number 5 before executing line number 6, we have s’ and k’.

8.s’ and k’ satisfies:{(s’ = (k rem 10)-s = m[i+1]-s) AND (k’ = m[q]…m[i+2])}.

(from the algorithm DivisibleBy11(n))

9.If m[i+1]…m[2]m[1] rem 11 = 0:

10. (m[i+1]x+m[i]…m[2]m[1]) rem 11 =0

(from the property of decimal system)

11. {(m[i]…m[2]m[1] rem 11) + (m[i+1]x rem 11) = 0} OR {(m[i]…m[2]m[1] rem 11) + (m[i+1]x rem 11) = 11}

(from enumeration)

12 (m[i]…m[2]m[1] rem 11) + (m[i+1]x rem 11)0 if and only if m[i]…m[2]m[1] rem 11=0 and m[i+1]x rem 11 =0 ( from the property of rem, remainders are nonnegative integers.)

13. m[i+1] (from the property of decimal system)

14. (m[i+1]x rem 11)0 (from the property of decimal system and integer multiplication)

15. (m[i]…m[2]m[1] rem 11) + (m[i+1]x rem 11)0 (direct prove from 12 and 14)

so (m[i]…m[2]m[1] rem 11) + (m[i+1]x rem 11) = 11. (direct prove from 11 and 15)

16. (m[i+1]x rem 11) < 11 (from the property of rem)

17. m[i]…m[2]m[1] rem 11

b) lemma: for an arbitrary positive integer n, we write n in decimal system, meanly, if n has k digits, n = m[k]…m[2]m[1],when the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, (notice: for every positive integer, DivisibleBy11(n) performs the while loop at least 1 time) if i is odd, , (m[i]… m[2]m[1] rem 11) = s. if i is even, (m[i]… m[2]m[1] rem 11) = (11-s).

proof in induction:

let P(i) = ” when the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, (notice: for every positive integer, DivisibleBy11(n) performs the while loop at least 1 time) if i is odd, (m[i]… m[2]m[1] rem 11) = s. if i is even, (m[i]… m[2]m[1] rem 11) = (11-s).”

for an arbitrary positive integer n, assume n has q digits in decimal system.

Base case:

When i = 1, s = (n rem 10) = m[1], k = (n div 10) = m[i]… m[3]m[2].

(m[1] rem 11) = m[1] = s

p(1) is true.

When i = 2, when executed the line number 5 and before the execution of line number 6, s = ((n div 10) rem 10) – (n rem 10) = (m[i]… m[3]m[2] rem 10) – m[1] = m[2] – m[1], k = ((n div 10) div 10))= m[i]… m[4]m[3].

When executing line number 6, if s <0 previously, then (m[2] < m[1]) AND (s = 11 - m[2] + m[1]),

If s>0 previously, then (m[2] > m[1]) AND (s = m[2] - m[1]),

Constructor case:

Assume for an arbitrary i q, p(i) is true. When the algorithm DivisibleBy11(n) performs the while loop the i th time just finished the 6th line, we have k and s,

Assume i is odd.

k = m[q]…m[i+1], s = (m[i]… m[2]m[1] rem 11)

i+1 is even,

if the i+1 th while loop does not exist, then p(i) is true for every iq.

else, for the i+1 th while loop, before the execution of line number 6,

s’ = (k rem 10) -s = m[i+1]-s

k’ = k div 10 = m[q]…m[i+2],