Cyclic Hanoi Problem

Presented by Zheng Zangwei

Problem

- We are given three pegs (A, B, C), which are arranged as a circle with the clockwise and the counterclockwise directions being defined as A - B - C - A and A - C - B - A respectively. The moving direction of the disk must be clockwise.

- Input : N

- Output : Series of moving process

- Relation : It should be valid process (according to the problem) to move N disks.

Clockwise

Subroutine Move_clockwise(n,x,y,z)//x to y

1. If (n==1) then Move from x to y,return
2. Move_clockwise(n-1,x,y,z)
3. Move_clockwise(n-1,y,z,x)
4. Move from x to y
5. Move_clockwise(n-1,z,x,y)
6. Move_clockwise(n-1,x,y,z)

Analogy to DH

- We will add the assertion just prior to entering the entire routine.
- We will define our invariants like the DH does
- We will use mathematical induction to prove it.

DH:
the invariant of Hanoi problem

Ha? So long!

- (S): Assume that the peg names A, B, and C are associated, in some order, with the variables X, Y and Z. Then a terminating execution of the call **move N from X to Y using Z** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, correctly moves those N rings from X to Y, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the Towers of Hanoi problem, and it leaves all other rings untouched.

DH: the invariant of Hanoi problem

- (S): Assume that the peg names A, B, and C are associated, in some order, with the variables X, Y and Z. Then a terminating execution of the call **move N from X to Y using Z** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, correctly moves those N rings from X to Y, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the Towers of Hanoi problem, and it leaves all other rings untouched.

Partial Correctness

- (clockwise): Assume that the peg names A, B, and C are associated, in an order that A-B-C are clockwise, with the variables X, Y and Z. Then a terminating execution of the call **Move_clockwise(N, X, Y, Z)** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, correctly moves those N rings from X to Y, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the Cyclic Towers of Hanoi problem, and it leaves all other rings untouched.

Mathematical induction

- Prove (P): for every n, (clockwise) holds
- When n==1, this is trivial.
- Assume now that the statement (P) holds for some arbitrary N-1. …

Clockwise

```
Subroutine Move_clockwise(n,x,y,z)//x to y
 1. If (n==1) then Move from x to y,return
        2. Move_clockwise(n-1,x,y,z)
        3. Move_clockwise(n-1,y,z,x)
             4. Move from x to y
        5. Move_clockwise(n-1,z,x,y)
        6. Move_clockwise(n-1,x,y,z)
```

- the sequence adheres to the rules of the Cyclic Towers of Hanoi problem
- leaves all other rings untouched.

Termination

- This routine terminates for every N.
- Proof : By each recursion, N decreases by 1. And N cannot be less than 1 because if N reaches 1 it will meet the return clause. Consequently the trees of the recursion is finite and execution must terminate.

So far, we have proved the total correctness.

Clockwise

```
Subroutine Move_clockwise(n,x,y,z)//x to y
1. If (n==1) then Move from x to y,return
2. Move_counterclockwise(n-1,x,z,y)
3. Move from x to y
4. Move_counterclockwise(n-1,z,y,x)
```

Counter_Clockwise

```
Subroutine Move_counterclockwise(n,x,y,z)//x to y
1. If (n==1) then Move counterclockwise from x to y,return
2. Move_clockwise(n-1,x,z,y)
3. Move counterclockwise from x to y
4. Move_counterclockwise(n-1,z,y,x)
```

Clockwise

Subroutine Move_clockwise(n,x,y,z)//x to y

1. If (n==1) then Move from x to y,return
2. Move_counterclockwise(n-1,x,z,y)
3. Move from x to y
4. Move_counterclockwise(n-1,z,y,x)

Counter_Clockwise

Subroutine Move_counterclockwise(n,x,y,z)//x to y

1. If (n==1) then Move from x to z, Move from z to y,return
2. Move_counterclockwise(n-1,x,y,z)
3. Move from x to z
4. Move_clockwise(n-1,y,x,z)
5. Move from z to y
6. Move_counterclockwise(n-1,x,y,x)

Partial Correctness

- (clockwise): Assume that the peg names A, B, and C are associated, in an order that A-B-C are clockwise, with the variables X, Y and Z. Then a terminating execution of the call **Move_clockwise(N, X, Y, Z)** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, correctly moves those N rings from X to Y, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the Cyclic Towers of Hanoi problem, and it leaves all other rings untouched.

Partial Correctness

- (counterclockwise): Assume that the peg names A, B, and C are associated, <u>in an order that A-B-C are counterclockwise</u>, with the variables X, Y and Z. Then a terminating execution of the call **Move_counterclockwise(N, X, Y, Z)** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, <u>correctly moves those N rings from X to Y</u>, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the <u>Cyclic Towers of Hanoi problem</u>, and it leaves all other rings untouched.

Mathematical induction

- Prove (P): for every n, (clockwise) and (counterclockwise) holds
- When n==1, this is trivial.
- Assume now that the statement (P) holds for some arbitrary N-1. …

Clockwise

Subroutine Move_clockwise(n, x, y, z)//x to y

1. If (n==1) then Move from x to y, return
2. Move_counterclockwise(n-1, x, z, y)
3. Move from x to y
4. Move_counterclockwise(n-1, z, y, x)

Counter_Clockwise

Subroutine Move_counterclockwise(n, x, y, z)//x to y

1. If (n==1) then Move from x to z, Move from z to y, return
2. Move_counterclockwise(n-1, x, y, z)
3. Move from x to z
4. Move_clockwise(n-1, y, x, z)
5. Move from z to y
6. Move_counterclockwise(n-1, x, z, y)

- the sequence adheres to the rules of the Cyclic Towers of Hanoi problem
- leaves all other rings untouched.

Termination

- This routine terminates for every N.
- Proof : We denote each time Move_clockwise or Move_counterclockwise being called one operation. By each operation, N decreases by 1. And N cannot be less than 1 because if N reaches 1 it will meet the return clause. Consequently the trees of the operation is finite and execution must terminate.

So far, we have proved the total correctness.

Now the answer need to be best.

Partial Correctness

- (clockwise): Assume that the peg names A,B, and C are associated, in an order that A-B-C are clockwise, with the variables X,Y and Z. Then a terminating execution of the call **Move_clockwise(N,X,Y,Z)** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, correctly moves those N rings from X to Y, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the Cyclic Towers of Hanoi problem, and it leaves all other rings untouched. In addition, it is costs the least operation.

Partial Correctness

- (counterclockwise): Assume that the peg names A, B, and C are associated, in an order that A-B-C are counterclockwise, with the variables X, Y and Z. Then a terminating execution of the call **Move_counterclockwise(N, X, Y, Z)** lists a sequence of ring-moving instructions, which, if started (and followed faithfully) in and legal configuration of the rings and pegs in which at least the N smallest rings are on peg X, correctly moves those N rings from X to Y, possibly using Z as temporary storage. Moreover, the sequence adheres to the rules of the Cyclic Towers of Hanoi problem, and it leaves all other rings untouched. In addition, it is costs the least operation.

Clockwise

Subroutine Move_clockwise(n, x, y, z)//x to y

1. If (n==1) then Move from x to y, return
2. Move_counterclockwise(n-1, x, z, y)
3. Move from x to y
4. Move_counterclockwise(n-1, z, y, x)

Counter_Clockwise

Subroutine Move_counterclockwise(n, x, y, z)//x to y

1. If (n==1) then Move from x to z, Move from z to y, return
2. Move_counterclockwise(n-1, x, y, z)
3. Move from x to z
4. Move_clockwise(n-1, y, x, z)
5. Move from z to y
6. Move_counterclockwise(n-1, x, z, y)

- the sequence adheres to the rules of the Cyclic Towers of Hanoi problem
- leaves all other rings untouched.

- In addition, it is costs the less operation.

So far, we have proved the total correctness.

Clockwise

```
Subroutine Move_clockwise(n,x,y,z)//x to y
1. If (n==1) then Move from x to y,return
2. Move_clockwise(n-1,x,y,z)
3. Move_clockwise(n-1,y,z,x)
4. Move from x to y
5. Move_clockwise(n-1,z,x,y)
6. Move_clockwise(n-1,x,y,z)
```

N=3, A->B:21; Otherwise latter one is 15

Calculate the number

- The first one : An = 4An + 1, A1=1; An = 4^(n-1)*(4/3) - 1/3
- The second one : Cn = 2A(n-1) +1; An = 2+C(n-1)+2A(n-1)
- $$c_n = \frac{1}{2\sqrt{3}}\{(1+\sqrt{3})^{n+1} - (1-\sqrt{3})^{n+1}\} - 1$$

  $$a_n = \frac{1}{4\sqrt{3}}\{(1+\sqrt{3})^{n+2} - (1-\sqrt{3})^{n+2}\} - 1$$

That’ s all for today.
THANK YOU!