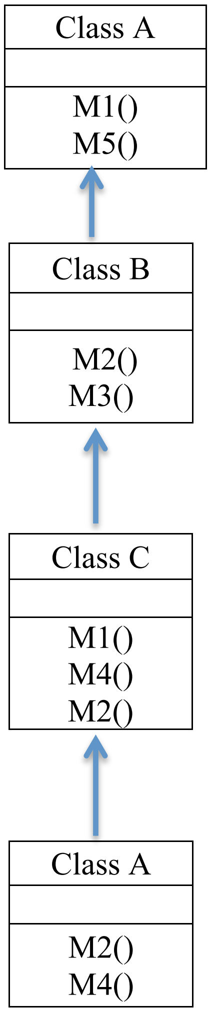
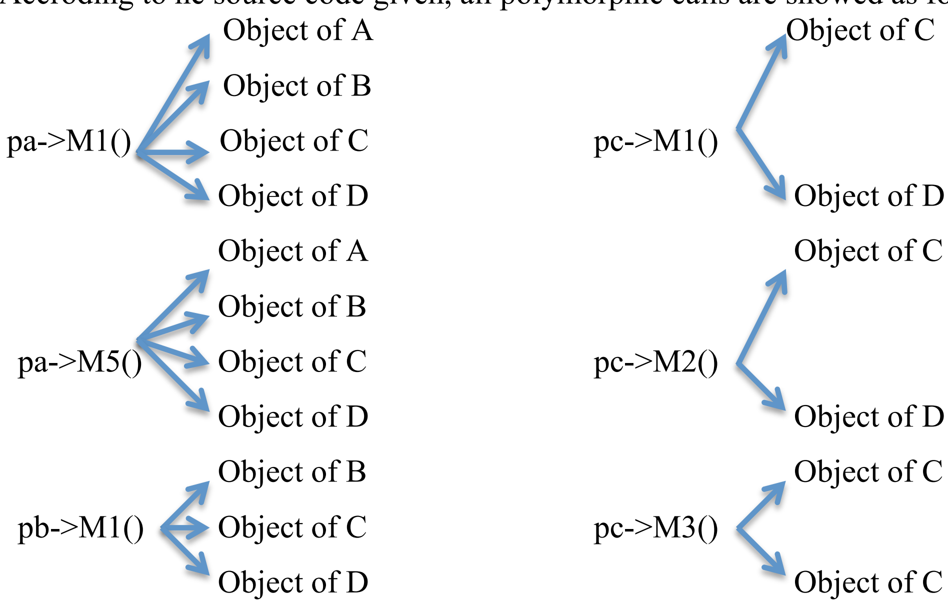
CS589 HW3 A20315603 Xiaoyang Lu

Problem 1 Testing Polymorphism



Accroding to he source code given, all polymorphic calls are showed as following:



pb->M5() → Object of B, Object of C, Object of D

pc->M4() → Object of C, Object of D

pb->M2() → Object of B, Object of C, Object of D

pc->M5() → Object of C, Object of D

Test #1: x=1, y=3, z=-1

pb->M1()----object of D, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#5, msg#5

Test #2: x=1, y=3, z=0

pb->M5()----object of D, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#2, msg#5

Test #3: x=1, y=3, z=1

pb->M2()----object of D, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#8, msg#5

Test #4: x=1, y=2, z=6

pc->M1()----object of D is executed.
On this test the following messages is printed: msg#5

Test #5: x=2, y=2, z=6

pc->M2()----object of D is executed.
On this test the following messages is printed: msg#8

Test #6 x=3, y=2, z=6

pc->M3()----object of D is executed.
On this test the following messages is printed: msg#4

Test #7: x=4, y=2, z=6

pc->M4()----object of D is executed.
On this test the following messages is printed: msg#9

Test #8: x=1, y=2, z=-2

pa->M1()----object of A, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#1, msg#5

Test #9: x=1, y=2, z=-1

pa->M5()----object of A, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#2, msg#5

Test #10: x=1, y=2, z=0

pb->M1()----object of C, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#5, msg#5

Test #11: x=1, y=2, z=1

pb->M5()----object of C, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#2, msg#5

Test #12: x=1, y=2, z=2

pb->M2()----object of C, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#7, msg#5

Test #13: x=1, y=1, z=-1
pa->M1()----object of B, pc->M1()----object of C are executed.
On this test the following messages are printed: msg#1, msg#5

Test #14: x=2, y=1, z=0
pa->M5()----object of B, pc->M2()----object of C are executed.
On this test the following messages are printed: msg#2, msg#7

Test #15: x=3, y=1, z=1
pb->M1()----object of B, pc->M3()----object of C are executed.
On this test the following messages are printed: msg#1, msg#4

Test #16: x=4, y=1, z=2
pb->M5()----object of B, pc->M4()----object of C are executed.
On this test the following messages are printed: msg#2, msg#6

Test #17: x=5, y=1, z=3
pb->M2()----object of B, pc->M5()----object of C are executed.
On this test the following messages are printed: msg#3, msg#2

pa->M1()---object of C and object of D, pa->M5()---object of C and object of D are non-executable.

Problem 2 Symbolic Evaluation

It is in hand writing paper.

Problem 3 Program Proving

Identity a loop invariant at 6, for all $(1 \le t \le i-1):(min \le |a[t]|)$

Prove the correctness of this loop invariant using mathematical induction.

1. Loop entry

   at 6, i=2, min=a[1]

   for all $(1 \le t \le i-1):(min \le |a[t]|)$

   $\equiv$ for all $(1 \le t \le 2-1):(a[1] \le |a[t]|)$

   $\equiv$ t=1 and $a[1] \le |a[1]|$ which is true

2. Assume that for some k, the loop invariant is true.

   It means that for all $(1 \le t \le i_k-1):(min_k \le |a[t]|)$ is true.

   i_k, min_k are values of i and min when execution reaches point 6 for k-th times.

3. We have to prove that the loop invariant is true for k+1.

   It means that for all $(1 \le t \le i_{k+1}-1):(min_{k+1} \le |a[t]|)$ is true.

i_{k+1}, min_{k+1} are values of i and min when execution reaches point 6 for (k+1)-th times.

There are 4 subpath inside the loop.

Path #1: 6,7,8,9,10,11,6

Path #2: 6,7,8,9,11, 6

Path #3: 6,7,9,10,11,6

Path #4: 6,7,9,11,6

Case # 1:

From the source code, we can know that

(1) $i_{k+1}=i_k+1$

(2) $min_{k+1} =|a[i_k]|$

Because the condition 9 is satisfied, (3)$min_k>|a[i_k]|$ is true.

From (2) and (3), we can get (4) $min_k>min_{k+1}$

For all $(1\le t\le i_{k+1}-1):(min_{k+1}\le|a[t]|)$

$\equiv$for all $(1\le t\le i_k+1-1):(min_{k+1}\le|a[t]|)$

$\equiv$for all $(1\le t\le i_k-1+1):(min_{k+1}\le|a[t]|)$

$\equiv$for all $(1\le t\le i_k-1):(min_{k+1}\le|a[t]|)$ and $min_{k+1}\le|a[i_k]|$

From (4) and the assumption for all $(1\le t\le i_k-1):(min_k\le|a[t]|)$,

for all $(1\le t\le i_k-1):(min_{k+1}\le|a[t]|)$ is true.

From (2), $min_{k+1}\le|a[i_k]| \equiv |a[i_k]| \le|a[i_k]|$ is true.

So, for all $(1\le t\le i_{k+1}-1):(min_{k+1}\le|a[t]|)$ is true.

Case # 2:

From the source code, we will know that

(1) $i_{k+1}=i_k+1$

(2) $min_{k+1}= min_k$

because the condition 9 is not satisfied, (3) $min_k \le |a[i_k] |$ is ture.

From(2) and (3), we will get (4) $min_{k+1} \le |a[i_k] |$

For all $(1\le t\le i_{k+1}-1):(min_{k+1}\le|a[t]|)$

$\equiv$for all $(1\le t\le i_k+1-1):(min_{k+1}\le|a[t]|)$

$\equiv$for all $(1\le t\le i_k-1+1):(min_{k+1}\le|a[t]|)$

$\equiv$for all $(1\le t\le i_k-1):(min_{k+1}\le|a[t]|)$ and $min_{k+1}\le|a[i_k]|$

From (2) and the assumption for all $(1 \leq t \leq i_k - 1):(min_k \leq |a[t]|)$,

For all $(1 \leq t \leq i_k - 1):(min_{k+1} \leq |a[t]|)$ is true.

From(4), $min_{k+1} \leq |a[i_k]|$ is obviously true.

So, for all $(1 \leq t \leq i_{k+1} - 1):(min_{k+1} \leq |a[t]|)$ is true.

Case # 3:

It is easy to see that the conditions in the case are same as conditions in case # 1, so the whole process of proving for case # 3 is the same as the case # 1.

Case # 4:

It is easy to see that the conditions in the case are same as conditions in case # 2, so the whole process of proving for case # 4 is the same as the case # 2.

4. On termination

$i = n+1$

For all $(1 \leq t \leq i_{k+1} - 1):(min_{k+1} \leq |a[t]|)$

$\equiv$for all $(1 \leq t \leq n+1-1):(min_{k+1} \leq |a[t]|)$

$\equiv$for all $(1 \leq t \leq n):(min_{k+1} \leq |a[t]|)$

which is exactly post condition.

So the function is correct with respect to the given pre-condition and post-condition.