CS 352. Jens Palsberg Dec 11,2000

Final Exam – Solutions

Question 1:

If the type checker tries to check the statement C().m(5,6) then it will have to consult the symbol table of class C to get the type of the method m.

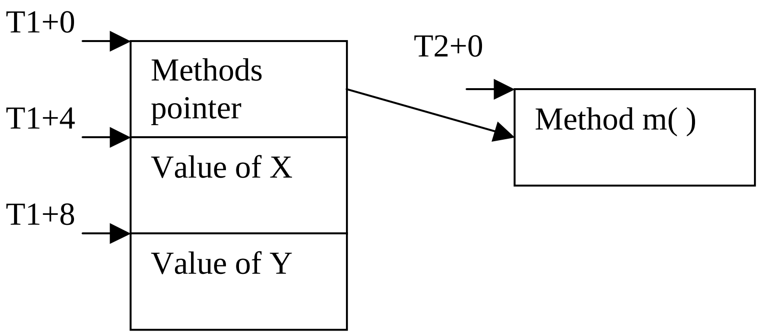
Assuming that we have a 2 pass compiler, the symbol table of class C should at this time look like:

Key	Value
C.x	Int
C.y	Int
C.m.num	Int
C.m.a	Int
C.m.i	Int
C.m.j	Int
Methods	
C.m	Int

Then the type check would just require access to the method field C.m

Question 2:

Heap layout of the C object where T1 and T2 are addresses allocated by HALLOCATE



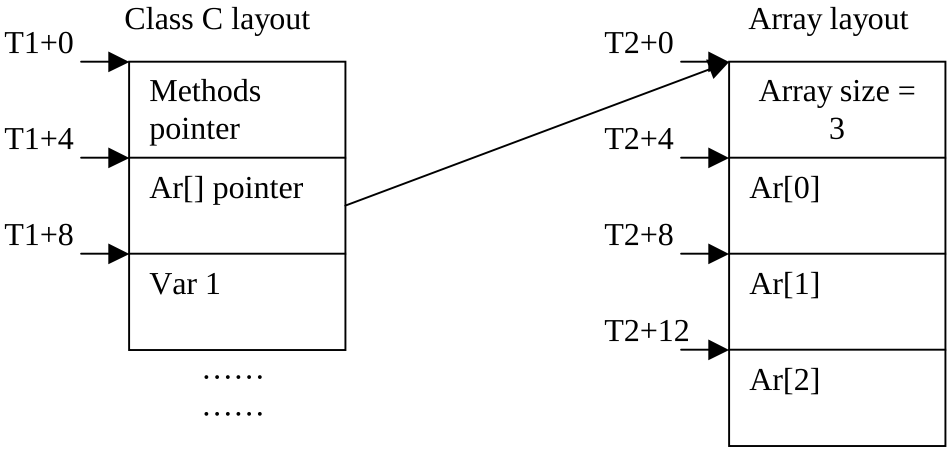
Piglet translation of the code:

```
PRINT CALL
BEGIN
    MOVE TEMP 24
       BEGIN
             MOVE TEMP 25 HALLOCATE  4
             MOVE TEMP 26 HALLOCATE  16
             HSTORE TEMP 25  0 C_m
             MOVE TEMP 27  4
      L0    CJUMP  LT TEMP 27  16 L1
             HSTORE  PLUS TEMP 26 TEMP 27  0  0
             MOVE TEMP 27  PLUS TEMP 27  4
             JUMP L0
      L1    HSTORE TEMP 26  0 TEMP 25
      RETURN
      TEMP 26
      END

    HLOAD TEMP 22 TEMP 24  0
    MOVE TEMP 23  PLUS TEMP 22  0
 RETURN
TEMP 23
END
(TEMP 24  5  6 )
END
```

Question 3:
Let int[] ar be the first field declaration in a class C. Then the offset of the pointer to the array structure will be 4 (from the beginning of the class structure)
In other words if the class C heap layout starts at TEMP 0 then the array pointer will be at TEMP 0 + 4

Assuming HALLOCATE returned address T2 we do the following allocation
for the array:

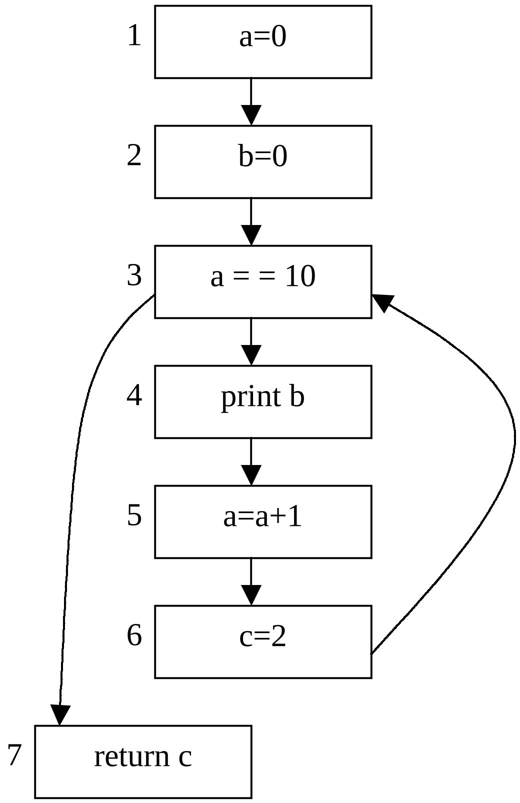


If we want to access the element a[3] and suppose that the pointer to the Array layout is stored in class variable TEMP 20 then the Piglet code for that would look like:

```
    MOVE TEMP 21
      BEGIN
            HLOAD TEMP 32  PLUS TEMP 20  PLUS
                  BEGIN
                        MOVE TEMP 30  TIMES  3  4
                        HLOAD TEMP 31 TEMP 20  0
                        CJUMP  MINUS  1  LT TEMP 30 TEMP 31 L4
                        ERROR
                        L4      NOOP
                  RETURN
                  TEMP 30
                  END
            4  0
      RETURN
      TEMP 32
      END

 RETURN
TEMP 21
```

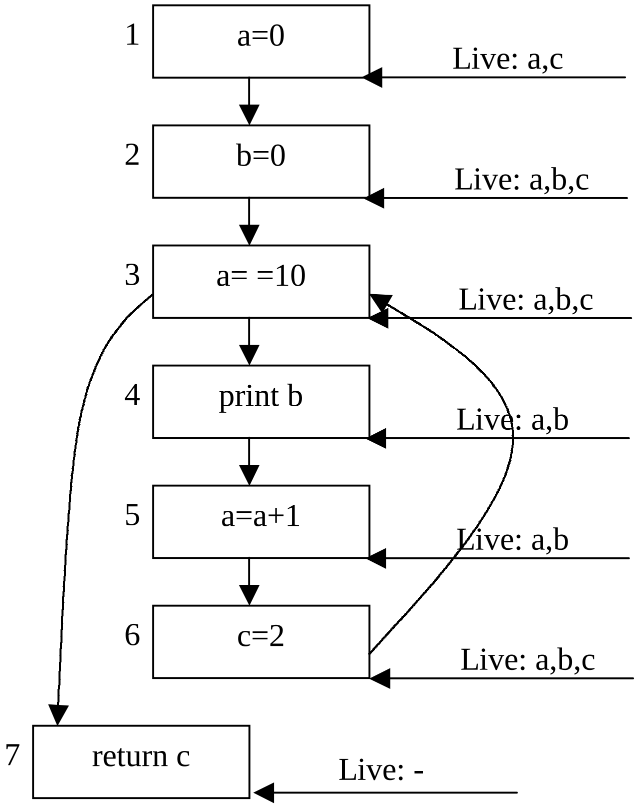
Question 4:



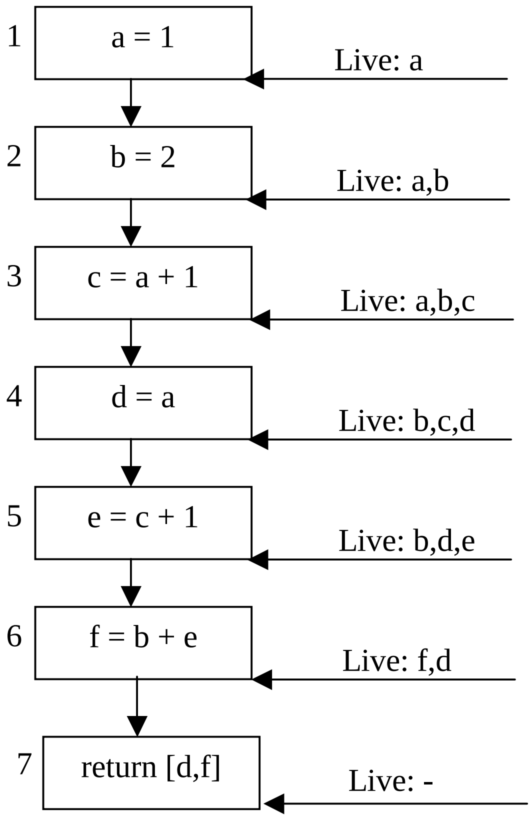
Using the algorithm 10.4 in the book we construct the following table

state	use	def	in	out	in	out	in	out	in	out	in	out	in	out
1		a						a		ac	c	ac	c	ac
2		b				a	a	abc	ac	abc	ac	abc	ac	abc
3	a		a		a	bc	abc	abc	abc	abc	abc	abc	abc	abc
4	b		b		b	a	ab	a	ab	a	ab	ab	ab	ab
5	a	a	a		a		a	a	a	ab	ab	ab	ab	ab
6		c				a	a	abc	ab	abc	ab	abc	ab	abc
7	c		c		c		c		c		c		c	

Then we complete the diagram:



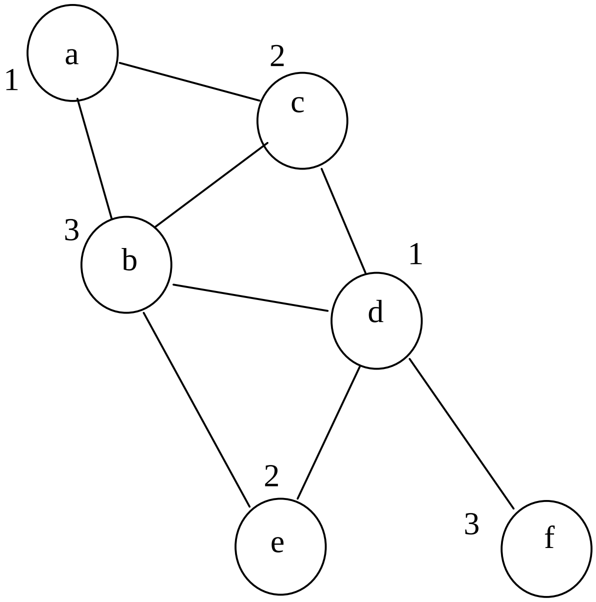
Question 5



As we see from the diagram above we can always have at most 3 live variables which we can store in the three registers without spilling.

The following coloring scheme simplifies the initial program to:

```
r1=1
r3=2
r2=r1+1
r1=r1
r2=r2+1
r3=r3+r2
return r1,r3
```


a
1
2
c
3
b
1
d
2
e
3
f