CS 5130 Homework II

Liang Yan
From Computer Science

1. Answer the following questions related to reaching definitions analysis for the CFG
(a) Give the GEN and PRSV sets for each basic block

$U = \{r1@1, r10@2, r2@3, r11@4, r3@5, r15@6, r4@7, r13@8, r5@9, r14@10, r6@12, r14@13, r4@15, r14@16,$
$r1@17, r7@18, r15@19, r4@21, r11@22, r5@23, r15@24, r6@26, r10@27, r8@28, r11@29\}$

BB	GEN	PRSV
B0	r1@1,r10@2,r2@3,r11@4,r3@5	r15@6,r4@7,r13@8,r5@9,r14@10,r6@12,r14@13,r4@15,r14@16, r7@18,r15@19,r4@21,r5@23,r15@24,r6@26,r10@27,r8@28
B1	r15@6,r4@7,r13@8,r5@9,r14@10	r1@1,r10@2,r2@3,r11@4,r3@5,r6@12, r1@17,r7@18,r11@22,r6@26,r10@27,r8@28,r11@29
B2	r6@12,r14@13	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r4@15 r1@17,r7@18,r15@19,r4@21,r11@22,r5@23,r15@24,r10@27,r8@28,r11@29
B3	r4@15,r14@16,r1@17,r7@18,r15@19	r10@2,r2@3,r3@5,r13@8,r5@9,r14@10, r6@12,r14@13r5@23,r6@26,r10@27,r8@28
B4	r4@21,r11@22,r5@23,r15@24	r1@1,r10@2,r2@3,r3@5,r13@8,r14@10,r6@12,r14@13, r14@16, r1@17,r7@18,r6@26,r10@27,r8@28
B5	r6@26,r10@27,r8@28,r11@29	r1@1,r2@3,r3@5,r15@6,r4@7,r13@8,r5@9,r14@10,r14@13, r4@15,r14@16,r1@17,r7@18, r15@19,r4@21,r5@23,r15@24

(b)
Init.(iter.0)
IN(b) = $\emptyset$, OUT(b)=Gen(b) $\cup$ (IN(b) $\cap$ PRSV(b)) = Gen(b)

BB	IN	OUT
B0	$\emptyset$	r1@1,r10@2,r2@3,r11@4,r3@5
B1	$\emptyset$	r15@6,r4@7,r13@8,r5@9,r14@10
B2	$\emptyset$	r6@12,r14@13
B3	$\emptyset$	r4@15,r14@16,r1@17,r7@18,r15@19
B4	$\emptyset$	r4@21,r11@22,r5@23,r15@24
B5	$\emptyset$	r6@26,r10@27,r8@28,r11@29

iter.1
IN(b) = $\bigcup_{p \in pred(b)} OUT(p)$
OUT(b)=Gen(b) $\cup$ (IN(b) $\cap$ PRSV(b)) = Gen(b)

BB	IN	OUT
B0	$\emptyset$	r1@1,r10@2,r2@3,r11@4,r3@5
B1	r1@1,r10@2,r2@3,r11@4,r3@5,r4@21,r11@22,r5@23,r15@24	r1@1,r10@2,r2@3,r11@4,r3@5, r15@6,r4@7,r13@8,r5@9,r14@10,r11@22
B2	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r14@10,r4@15,r14@16,r1@17,r7@18,r15@19,r11@22	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r6@12,r14@13,r4@15,r1@17,r7@18,r15@19,r11@22
B3	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r6@12,r14@13,r4@15,r1@17,r7@18,r15@19,r11@22	r10@2,r2@3,r3@5,r13@8,r5@9,r6@12, r14@13,r4@15,r14@16,r1@17,r7@18,r15@19
B4	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8, r5@9,r14@10,r6@12,r14@13,r4@15,r1@17,r7@18,r15@19,r11@22	r1@1,r10@2,r2@3,r3@5,r13@8,r14@10, r6@12,r14@13,r1@17,r7@18,r4@21,r11@22,r5@23,r15@24
B5	r1@1,r10@2,r2@3,r3@5,r13@8,r14@10, r6@12,r14@13,r1@17,r7@18,r4@21,r11@22,r5@23,r15@24	r1@1,r2@3,r3@5,r13@8,r14@10,r14@13, r1@17,r7@18,r4@21,r5@23,r15@24,r6@26,r10@27,r8@28,r11@29

iter.2
IN(b) = $\bigcup_{p \in pred(b)} OUT(p)$
OUT(b)=Gen(b) $\cup$ (IN(b) $\cap$ PRSV(b)) = Gen(b)

BB	IN	OUT
B0	$\emptyset$	r1@1,r10@2,r2@3,r11@4,r3@5
B1	r1@1,r10@2,r2@3,r11@4,r3@5,r13@8,r14@10,r6@12,r14@13 r1@17,r7@18, r4@21,r11@22,r5@23,r15@24	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r14@10,r1@17,r7@18,r11@22
B2	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r14@10,r6@12,r14@13,r4@15,r14@16,r1@17,r7@18,r15@19,r11@22	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r6@12,r14@13,r4@15,r1@17,r7@18,r15@19,r11@22
B3	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8,r5@9, r6@12,r14@13,r4@15,r1@17,r7@18,r15@19,r11@22	r10@2,r2@3,r3@5,r13@8,r5@9,r6@12, r14@13,r4@15,r14@16,r1@17,r7@18,r15@19
B4	r1@1,r10@2,r2@3,r11@4,r3@5,r15@6,r4@7,r13@8, r5@9,r14@10,r6@12,r14@13,r4@15,r1@17,r7@18,r15@19,r11@22	r1@1,r10@2,r2@3,r3@5,r13@8,r14@10, r6@12,r14@13,r1@17,r7@18,r4@21,r11@22,r5@23,r15@24
B5	r1@1,r10@2,r2@3,r3@5,r13@8,r14@10, r6@12,r14@13,r1@17,r7@18,r4@21,r11@22,r5@23,r15@24	r1@1,r2@3,r3@5,r13@8,r14@10,r14@13, r1@17,r7@18,r4@21,r5@23,r15@24,r6@26,r10@27,r8@28,r11@29

iter.3
No change
(c)
B0 $r1 \Rightarrow r1@1, r2 \Rightarrow r2@3$
B1 $r3 \Rightarrow r3@5, r10 \Rightarrow r10@2, r11 \Rightarrow r11@4 or r11@22, r4 \Rightarrow r4@7, r13 \Rightarrow r13@8, r10 \Rightarrow r10@2, r5 \Rightarrow r5@9, r14 \Rightarrow r14@10$
B2 $r11 \Rightarrow r11@4 or r11@22, r14 \Rightarrow r14@13, r6 \Rightarrow r6@12, r14 \Rightarrow r14@13$
B3 $r10 \Rightarrow r10@2, r11 \Rightarrow r11@22 or r11@4, r4 \Rightarrow r4@15, r15 \Rightarrow r15@19 or r15@6, r1 \Rightarrow r1@17, r7 \Rightarrow r7@18$
B4 $r10 \Rightarrow r10@2, r11 \Rightarrow r11@22 or r11@4, r4 \Rightarrow r4@21, r13 \Rightarrow r13@8, r10 \Rightarrow r10@2, r5 \Rightarrow r5@23, r15 \Rightarrow r15@24$
B5 $r11 \Rightarrow r11@22, r14 \Rightarrow r14@13, r6 \Rightarrow r6@26, r10 \Rightarrow r10@27, r14 \Rightarrow r14@13, r8 \Rightarrow r8@28$

2.Answer the following questions related to liveness analysis analysis for the CFG
(a) Give the GEN and PRSV sets for each basic block
$U = \{r1, r2, r3, r4, r5, r6, r7, r8, r10, r11, r13, r14, r15\}$

BB	GEN	PRSV
B5	r11,r14	r1,r2,r3,r4,r5,r7,r13,r14,r15
B4	r10,r11,r13	r1,r2,r3,r6,r7,r8,r10,r13,r14
B3	r10,r11,r15	r2,r3,r5,r6,r8,r10,r13
B2	r11,r14	r1,r2,r3,r4,r5,r7,r8,r10,r11,r13,r15
B1	r3,r10,r11	r1,r2,r3,r6,r7,r8,r10,r11
B0	$\emptyset$	r4,r5,r6,r7,r8,r13,r14,r15

(b)Liveness is a backwards problem, That is because to verify an instruction is liveness or not we need to check if it is used in the future, so start from future to current is a good choice. we need to take postOrder to analysis it. and the out set of one block is the union of in set from all its successor.

(c)
Init.(iter.0)
OUT(b) = $\emptyset$, IN(b)=Gen(b) $\cup$ (OUT(b) $\cap$ PRSV(b)) = Gen(b)

BB	OUT	IN
B5	$\emptyset$	r11,r14
B4	$\emptyset$	r10,r11,r13
B3	$\emptyset$	r10,r11,r15
B2	$\emptyset$	r11,r14
B1	$\emptyset$	r3,r19,r11
B0	$\emptyset$	$\emptyset$

iter.1
OUT(b) = $\bigcup_{s \in succs(b)} IN(s)$
IN(b)=Gen(b) $\cup$ (OUT(b) $\cap$ PRSV(b))

BB	OUT	IN
B5	$\emptyset$	r11,r14
B4	r3,r11,r14,r10	r3,r10,r11,r13,r14
B3	r11,r14	r10,r11,r15
B2	r3,r10,r11,r13,r14,r15	r3,r10,r11,r13,r14,r15
B1	r3,r10,r11,r13,r14,r15	r3,r10,r11
B0	r3,r10,r11	$\emptyset$

iter.2
OUT(b) = $\bigcup_{s \in succs(b)} IN(s)$

IN(b)=Gen(b) ∪ (OUT(b) ∩ PRSV(b))

BB	OUT	IN
B5	∅	r11,r14
B4	r3,r10,r11,r14	r3,r10,r11,r13,r14
B3	r3,r10,r11,r13,r14,r15	r3,r10,r11,r13,r14,r15
B2	r3,r10,r11,r13,r14,r15	r3,r10,r11,r13,r14,r15
B1	r3,r10,r11,r13,r14,r15	r3,r10,r11
B0	r3,r10,r11	∅

iter.3
OUT(b) = $\bigcup_{s \in succs(b)} IN(s)$
IN(b)=Gen(b) ∪ (OUT(b) ∩ PRSV(b))

No change

3 living analysis
(a)pointer reference.
when facing this problem, we should try not to kill the variable that could be live.
if we could know which variable the pointer reference, we can just use this variable instead, however, sometimes we could not verify until run-time, so we need to find all possible variables, the simple ideas is to think it points to all variables.
Assume pointer on the right, GEN need to be a full set,because any variables could be live,PRSV no change,
Assume pointer on the left, GEN no change ,PRSV becomes empty set,
(b)procedure calls
if the parameter is value type, then we know it is only used as other operands, we can just add the parameter to GEN set. and do not change PRSV. but if the parameter is reference or pointer, the situation is changed, because we do not know if it is used as an operand or assignment. functions can do anything redene variables, use variables.If meeting this situation, I will let all Gen to emptyset and PRSV to be full set, then start over.

4.sinking transformation.
(a)
Sinking is a transformation that move an instruction to its succsor where its result is first needed. So we need to make sure the operands of the instruction could not be changed during its node to sinking point, also we need to find where the lvalue of the instruction is first used, then copy the instruction here and eliment the former one.
We can consider the sinking as an available expression problem,because according to its defination above, assume we find the sinking position, and insert the instruction before eliment the former ones, then the new insert one becomes an available expression, we can delete it, but if there is no defination of the lefe value between these instructions, then we can delete the formes ones. so the equation problem like available expression,only thing we need to notice is

1. the set of expressions that have definitely been computed and have not had operands redefined and the evalution have not used befre entering the current basic block IN

2. The set of expressions that are computed in the basic block whose operands are not changed and evalution is not used later in the same block GEN

3. The set of expressions computed anywhere in the function that do not have their operands defined and evalution used in the current block PRSV

4. The set of expressions that are definitely computed before or in the block and do not have their operands redefined and evalution used in the block - OUT

the equation is:

IN(b) = $\bigcap_{s \in pred(b)} OUT(s)$
OUT(b)=Gen(b) ∪ (OUT(b) ∩ PRSV(b))

(b)
Algorithm

```
EliminateRedundacy(G) {
    for each b in G {
        AVAIL = OUT(b)
        for each i in b in execution order {
          if i.lval() in AVAIL
             remove i
          if i.operand()in AVAIL
             insert the instruction from AVAIL index by i.operand()
        }
    }
}
```