CSE 231 Spring 1998
Marcio Faerman
HW 2

1. Reaching Definitions computation

1(a) Gen and Kill sets for each basic block.

BASIC BLOCKS	BLOCK STATEMENTS	GEN	KILL
Entry		{}	{}
B1	<s1> a := 1 <s2> b := 2	{a1, b2}	{a1, b2, b11, a14, b15}
B2	<s3> L1: c := a + b <s4> d := c - a <s5> if (…) then go to L3	{c3, d4}	{c3, d4, d6, d8}
B3	<s6> L2: d := b * d <s7> if (…) then go to L3	{d6}	{d4, d6, d8}
B4	<s8> d := a + b <s9> e := d + 5 <s10> go to L2	{d8, e9}	{d4, d6, d8, e9, e12, e16}
B5	<s11> L3: b := a + b <s12> e := c − a <s13> if (…) then go to L1	{b11, e12}	{b2, e9, b11, e12, b15, e16}
B6	<s14> a := b * d <s15> b := a − d <s16> L4: e := e − c <s17> f := e − a	{a14, b15, e16, f17}	{a1, b2, e9, b11, a14, b15, e12, e16, f17}
Exit		{}	{}

1(b) Worklist Computation

The reverse postorder used for the initialization of the worklist was <Entry, B1, B2, B3, B4, B5, B6, Exit> and we show two methods to compute the Reachin and Reachout sets. In the first one the computation of the worklist was run according to the algorithm of Lecture 4, slide 5, having new nodes inserted in the worklist following a FIFO discipline. In the second one, all blocks were computed at each iteration. The Reachin and Reachout sets were initalized to {} for all blocks in both methods.

BASIC BLOCKS	REACHIN	REACHOUT	WORKLIST	ITERATION NUMBER
Entry	{}	{}	<Entry, B1, B2, B3, B4, B5, B6, Exit>	1
B1	{}	{a1, b2}	<B1, B2, B3, B4, B5, B6, Exit>	2
B2	{a1, b2}	{a1, b2, c3, d4}	<B2, B3, B4, B5, B6, Exit>	3
B3	{a1, b2, c3, d4}	{d6, a1, b2, c3}	<B3, B4, B5, B6, Exit>	4
B4	{d6, a1, b2, c3}	{d8, e9, a1, b2, c3}	<B4, B5, B6, Exit>	5
B5	{a1, b2, c3, d4, d6}	{b11, e12, a1, c3, d4, d6}	<B5, B6, Exit, B3>	6
B6	{b11, e12, a1, c3, d4, d6}	{a14, b15, e16, f17, c3, d4, d6}	<B6, Exit, B3, B2>	7
Exit	{a14, b15, e16, f17, c3, d4, d6}	{a14, b15, e16, f17, c3, d4, d6}	<Exit, B3, B2>	8
B3	{a1, b2, c3, d4, d8, e9}	{d6, a1, b2, c3, e9}	<B3, B2>	9
B2	{a1, b2, b11, e12, c3, d4, d6}	{c3, d4, a1, b2, b11, e12}	<B2, B4, B5>	10
B4	{d6, a1, b2, c3, e9}	{d8, e9, a1, b2, c3}	<B4, B5, B3>	11
B5	{ a1, b2, c3, d4, b11, e12, d6, e9}	{b11, e12, a1, c3, d4, d6}	<B5, B3>	12
B3	{ a1, b2, c3, d4, b11, e12, d8, e9}	{d6, a1, b2, c3, b11, e12, e9}	<B3>	13
B4	{d6, a1, b2, c3, b11, e12, e9}	{d8, e9, a1, b2, c3, b11}	<B4, B5>	14
B5	{ a1, b2, c3, d4, b11, e12, d6, e9}	{b11, e12, a1, c3, d4, d6}	<B5, B3>	15
B3	{ a1, b2, c3, d4, b11, e12, d8, e9}	{d6, a1, b2, c3, b11, e12, e9}	<B3>	16

Therefore, after the worklist computation the final values for Reachin and Reachout are given by:

BASIC BLOCKS	REACHIN	REACHOUT
Entry	{}	{}
B1	{}	{a1, b2}
B2	{a1, b2, b11, e12, c3, d4, d6}	{c3, d4, a1, b2, b11, e12}
B3	{ a1, b2, c3, d4, b11, e12, d8, e9}	{d6, a1, b2, c3, b11, e12, e9}
B4	{d6, a1, b2, c3, b11, e12, e9}	{d8, e9, a1, b2, c3, b11}
B5	{ a1, b2, c3, d4, b11, e12, d6, e9}	{b11, e12, a1, c3, d4, d6}
B6	{b11, e12, a1, c3, d4, d6}	{a14, b15, e16, f17, c3, d4, d6}
Exit	{a14, b15, e16, f17, c3, d4, d6}	{a14, b15, e16, f17, c3, d4, d6}

1(c) The number of iterations before the computation terminates is equal to 16 according to the worklist table of item (b) and the maximum number of times any node is put on the worklist is 4.

1(b') In this second method we generate the Reachin and Reachout sets, computing all the blocks at each iteration. Following, the results for our CFG applying this second technique:

Iteration 1:

Reachout(Entry) = {}	Reachin(Entry)= {}
Reachout(B1) = {1, 2}	Reachin(B1) = {}
Reachout(B2) = {1, 2, 3, 4}	Reachin(B2) = {1, 2}
Reachout(B3) = {1, 2, 3, 6}	Reachin(B3) = {1, 2, 3, 4}
Reachout(B4) = {1, 2, 3, 8, 9}	Reachin(B4) = {1, 2, 3, 6}
Reachout(B5) = {1, 3, 4 ,6 ,11, 12}	Reachin(B5) = {1, 2, 3, 4 ,6}
Reachout(B6) = {3, 4, 6 ,14, 15, 16, 17}	Reachin(B6) = {1, 3, 4, 6 ,11, 12}
Reachout(Exit) = {3, 4, 6 ,14, 15, 16, 17}	Reachin(Exit) = {3, 4, 6 ,14, 15, 16, 17}

Iteration 2:

Reachout(Entry) = {}	Reachin(Entry)= {}
Reachout(B1) = {1, 2}	Reachin(B1) = {}
Reachout(B2) = {1, 2, 3, 4, 11, 12}	Reachin(B2) = {1, 2, 3 ,4, 6 , 11, 12}
Reachout(B3) = {1, 2, 3, 6, 9, 11, 12}	Reachin(B3) = {1, 2, 3, 4, 8, 9, 11, 12}
Reachout(B4) = {1, 2, 3, 8, 9, 11}	Reachin(B4) = {1, 2, 3, 6, 9, 11, 12}
Reachout(B5) = {1, 3, 4 ,6 , 11, 12}	Reachin(B5) = {1, 2, 3, 4 ,6, 9, 11, 12}
Reachout(B6) = {3, 4, 6 ,14, 15, 16, 17}	Reachin(B6) = {1, 3, 4, 6 ,11, 12}
Reachout(Exit) = {3, 4, 6 ,14, 15, 16, 17}	Reachin(Exit) = {3, 4, 6 ,14, 15, 16, 17}

Iteration 3:

Reachout(Entry) = {}	Reachin(Entry)= {}
Reachout(B1) = {1, 2}	Reachin(B1) = {}
Reachout(B2) = {1, 2, 3, 4, 11, 12}	Reachin(B2) = {1, 2, 3 ,4, 6 , 11, 12}
Reachout(B3) = {1, 2, 3, 6, 9, 11, 12}	Reachin(B3) = {1, 2, 3, 4, 8, 9, 11, 12}
Reachout(B4) = {1, 2, 3, 8, 9, 11}	Reachin(B4) = {1, 2, 3, 6, 9, 11, 12}
Reachout(B5) = {1, 3, 4 ,6 , 11, 12}	Reachin(B5) = {1, 2, 3, 4 ,6, 9, 11, 12}
Reachout(B6) = {3, 4, 6 ,14, 15, 16, 17}	Reachin(B6) = {1, 3, 4, 6 ,11, 12}
Reachout(Exit) = {3, 4, 6 ,14, 15, 16, 17}	Reachin(Exit) = {3, 4, 6 ,14, 15, 16, 17}

1(c') The computation terminates after 3 iterations, having found the answer after 2 iterations.

2. Upwards Exposed Problem

We characterize this problem as a **backward** problem. The local sets Gen and Kill are given by:

Gen(B): set of local uses in B not defined in B before their use (or in other words - the uses *upward exposed* until the beginning of block B).

Kill(B): set of uses of variables **defined** in B (these uses become no longer upward exposed).

Pres(B): complement of **Kill(B) –** set of uses of variables **not** defined in B.

The global sets UpExIn and UpExOut are given by the following data flow equations:

$$\mathbf{UpExOut(B)} = \varnothing \qquad \text{for B = Exit}$$

$$\mathbf{UpExOut(B)} = \bigcup_{S \in \text{succ of B}} \mathbf{UpExIn(S)} \qquad \text{otherwise}$$

$$\mathbf{UpExIn(B) = Gen(B)} \cup \mathbf{(\ UpExOut(B)} \cap \mathbf{Pres\ (B)\ )}$$