

Assignment 2 - Linguaggi e compilatori

Marco Pini (169105), Ronchetti Gian Marco (164907)

	Dataflow problem 1 - Very Busy Expressions
Domain	Expressions
Direction	Backward: $IN[B] = OUT[B]$ $OUT[B] = \cap IN[succ(B)]$
Transfer Function	$f_B(x) = GEN[B] \cup (OUT[B] - KILL[B])$
Meet Operation(\wedge)	\cup
Boundary Condition	$IN[EXIT] = \emptyset$
Initial interior points	$IN[B] = \emptyset$

Date le espressioni a-b e b-a definiamo un bit vector composto da due bit, il primo per l'espressione a-b e il secondo per b-a. Quando uno dei due valori sarà a 1 vorrà dire che la corrispondente espressione sarà very busy

	Iterazione 1		Iterazione 2	
	IN[B]	OUT[B]	IN[B]	OUT[B]
BB1	<1,1>	<1,1>	<1,1>	<1,1>
BB2	<1,1>	<1,1>	<1,1>	<1,1>
BB3	<1,1>	<1,0>	<1,1>	<1,0>
BB4	<1,0>	\emptyset	<1,0>	\emptyset
BB5	<0,1>	\emptyset	<0,1>	\emptyset
BB6	\emptyset	<1,0>	\emptyset	<1,0>
BB7	<1,0>	\emptyset	<1,0>	\emptyset
BB8	\emptyset	<>	\emptyset	<>

	Dataflow problem 2 - Dominator Analysis
Domain	Basic Blocks
Direction	Forward: DOMOUT[B]= GEN[B] U DOMIN[B] DOMIN[B]= \cap DOMOUT[pred[b]]
Transfer Function	$f_B(x) = \{B\} \cup \cap \text{DOMOUT}[\text{pred}(B)]$
Meet Operation(\wedge)	\cap
Boundary Condition	DOMOUT[entry]= \emptyset
Initial interior points	DOMIN[B]=u

Il bit vector in questo caso sarà composto da un bit per ogni blocco in ordine alfabetico.

	IN[B]	OUT[B]
ENTRY	\emptyset	\emptyset
A	\emptyset	<1,0,0,0,0,0,0>
B	<1,0,0,0,0,0,0>	<1,1,0,0,0,0,0>
C	<1,0,1,0,0,0,0>	<1,0,1,0,0,0,0>
D	<1,0,1,0,0,0,0>	<1,0,1,1,0,0,0>
E	<1,0,1,0,0,0,0>	<1,0,1,0,1,0,0>
F	<1,0,1,0,0,0,0>	<1,0,1,0,0,1,0>
G	<1,0,0,0,0,0,0>	<1,0,0,0,0,0,1>

	Dataflow problem 3 - Constant Propagation
Domain	tuple <variabile, costante>
Direction	Forward: $OUT[B] = GEN[B] \cup IN[B]$ $IN[B] = \cap OUT[pred[b]]$
Transfer Function	$f_B(x) = GEN[B] \cup (OUT[B] - KILL[B])$
Meet Operation(\wedge)	\cap
Boundary Condition	$OUT[entry] = \emptyset$
Initial interior points	$DOMIN[B] = u$

In questo caso non è stato utilizzato un bit vector ma un insieme di tuple(<<costante>,<valore>>) per ogni punto.

	Iterazione 1		Iterazione 2		Iterazione 3	
	IN[B]	OUT[B]	IN[B]	OUT[B]	IN[B]	OUT[B]
Entry	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
BB1	\emptyset	<<k,2>>	\emptyset	<<k,2>>	\emptyset	<<k,2>>
BB2	<<k,2>>	<<k,2>>	<<k,2>>	<<k,2>>	<<k,2>>	<<k,2>>
BB3	<<k,2>>	<<k,2>, <a,4>>	<<k,2>>	<<k,2>, <a,4>>	<<k,2>>	<<k,2>, <a,4>>
BB4	<<k,2>, <a,4>>	<<k,2>, <a,4>, <x,5>>	<<k,2>, <a,4>>	<<k,2>, <a,4>, <x,5>>	<<k,2>, <a,4>>	<<k,2>, <a,4>, <x,5>>
BB5	<<k,2>>	<<k,2>, <a,4>>	<<k,2>>	<<k,2>, <a,4>>	<<k,2>>	<<k,2>, <a,4>>
BB6	<<k,2>, <a,4>>	<<k,2>, <a,4>, <x,8>>	<<k,2>, <a,4>>	<<k,2>, <a,4>, <x,8>>	<<k,2>, <a,4>>	<<k,2>, <a,4>, <x,8>>
BB7	<<k,2>, <a,4>>	<<k,4>, <a,4>>	<<k,2>, <a,4>>	<<k,4>, <a,4>>	<<k,2>, <a,4>>	<<k,4>, <a,4>>
BB8	<<k,4>, <a,4>>	<<k,4>, <a,4>>	<<a,4>>	<<a,4>>	<<a,4>>	<<a,4>>
BB9	<<k,4>, <a,4>>	<<k,4>, <a,4>, <b,2>>	<<a,4>>	<<a,4>, <b,2>>	<<a,4>>	<<a,4>, <b,2>>

BB10	<<k,4>, <a,4>, <b,2>>	<<k,4>, <a,4>, <b,2>, <x,8>>	<<a,4>, <b,2>>	<<a,4>, <b,2>>	<<a,4>, <b,2>>	<<a,4>, <b,2>>
BB11	<<k,4>, <a,4>, <b,2>, <x,8>>	<<k,4>, <a,4>, <b,2>, <x,8>, <y,8>>	<<a,4>, <b,2>>	<<a,4>, <b,2>, <y,8>>	<<a,4>, <b,2>>	<<a,4>, <b,2>, <y,8>>
BB12	<<k,4>, <a,4>, <b,2>, <x,8>, <y,8>>	<<k,5>, <a,4>, <b,2>, <x,8>, <y,8>>	<<a,4>, <b,2>, <y,8>>	<<a,4>, <b,2>, <y,8>>	<<a,4>, <b,2>, <y,8>>	<<a,4>, <b,2>, <y,8>>
BB13	<<k,4>, <a,4>>	<<k,4>, <a,4>>	<<a,4>>	<<a,4>>	<<a,4>>	<<a,4>>
Exit	<<k,4>, <a,4>>	<<k,4>, <a,4>>	<<a,4>>	<<a,4>>	<<a,4>>	<<a,4>>