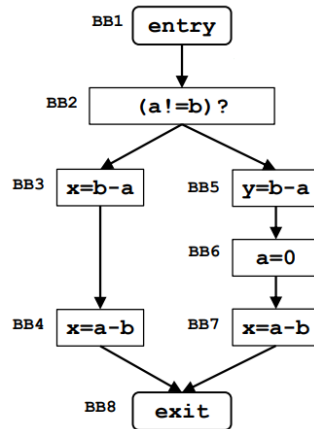


Assignment per il 19/04/2023

Linguaggi e Compilatori, gruppo 12.

Primo esercizio: Very Busy Expressions



- Derivare una formalizzazione per il framework di Dataflow Analysis, riempiendo lo specchio coi parametri adeguati.

	Dataflow Problem Very Busy Expressions
Domain	Sets of expressions
Direction	Backward $\text{in}[b] = f_b(\text{out}[b])$ $\text{out}[b] = \wedge \text{in}[\text{succ}(b)]$
Transfer function	$f_b(x) = \text{GEN}_b \cup (x - \text{Kill}_b)$
Meet Operation (\wedge)	\cap
Boundary Condition	$\text{in}[\text{exit}] = \emptyset$
Initial interior points	$\text{in}[b] = \mu$

GEN e Kill sono rispettivamente: USE_{expr} e la definizione di un operando già usato.

- Per il CFG di esempio fornito popolare una tabella con le iterazioni dell'algoritmo iterativo di soluzione del problema.

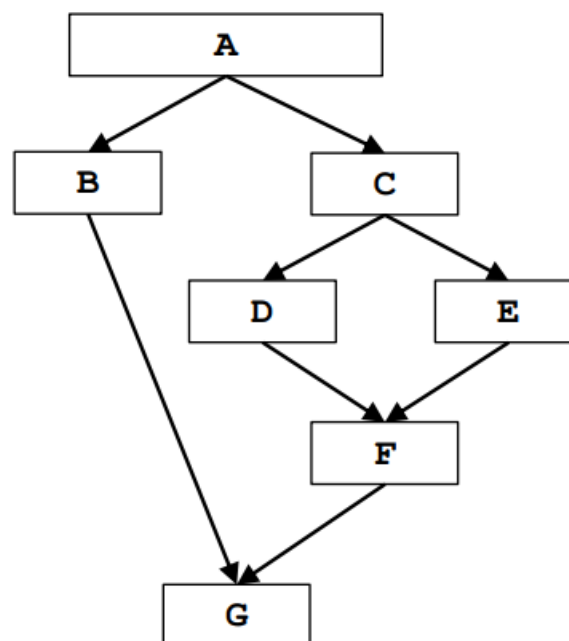
Bit Vector: $\langle b-a, a-b \rangle$

	ITERAZIONE 1		ITERAZIONE 2		ITERAZIONE 3	
	in	out	in	out	in	out
BB1	1,1	1,1	1,0	1,0	1,0	1,0
BB2	1,1	1,1	1,0	1,0	1,0	1,0
BB3	1,1	1,1	1,1	0,1	1,1	0,1

BB4	1,1	\emptyset	0,1	\emptyset	0,1	\emptyset
BB5	1,1	1,1	1,0	0,0	1,0	0,0
BB6	1,1	1,1	0,0	0,1	0,0	0,1
BB7	1,1	\emptyset	0,1	\emptyset	0,1	\emptyset
BB8	\emptyset	0,0	\emptyset	\emptyset	\emptyset	\emptyset

Non serve fare una quarta iterazione perché si è verificata la stabilità in due iterazioni successive, quindi l'algoritmo si fermerà.

Secondo esercizio: Dominator analysis



1. Derivare una formalizzazione per il framework di Dataflow Analysis, riempiendo lo specchio coi parametri adeguati.

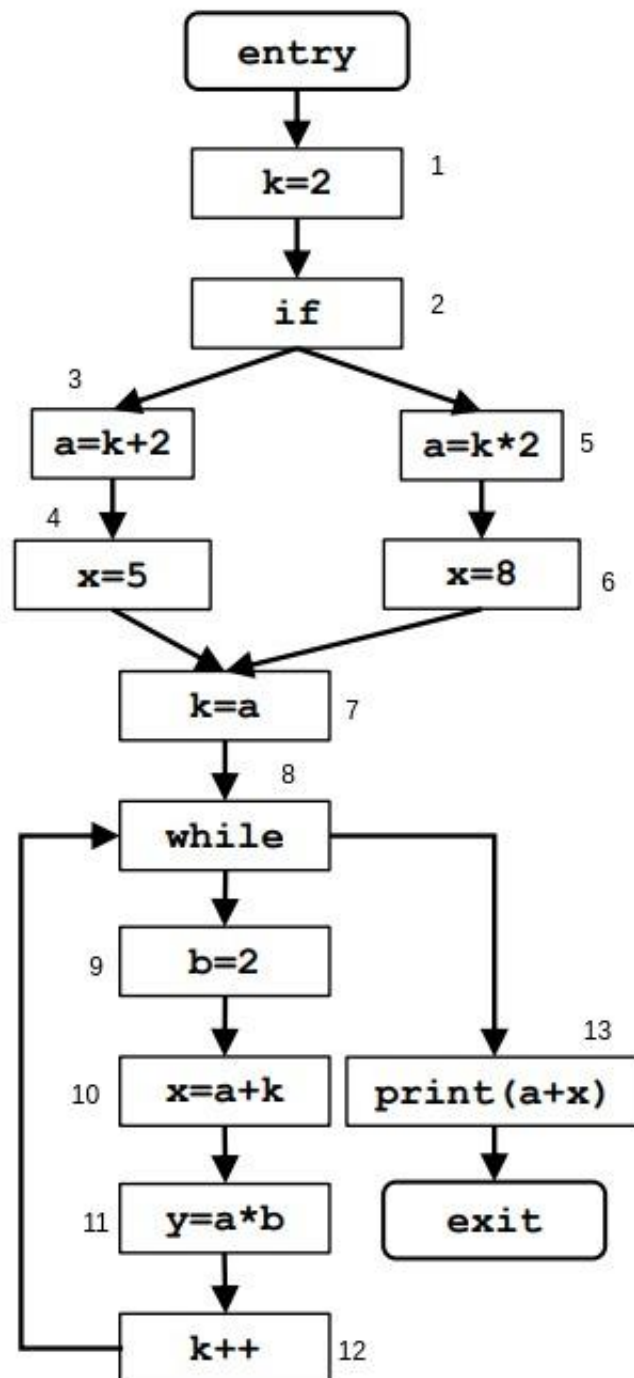
	Dataflow Problem Dominator Analysis
Domain	Sets of Basic Blocks
Direction	Forward $out[b] = f_b(in[b])$ $in[b] = \wedge out[prec(b)]$
Transfer function	$f_b(x) = B \cup x$
Meet Operation (\wedge)	\cap
Boundary Condition	$out[entry] = entry$
Initial interior points	$out[b_i] = \mu$

2. Per il CFG di esempio fornito popolare una tabella con le iterazioni dell'algoritmo iterativo di soluzione del problema.

	ITERAZIONE 1		ITERAZIONE 2		ITERAZIONE 3	
	in	out	in	out	in	out
A	A	A	A	A	A	A
B	A	B	A	AB	A	AB
C	AC	C	A	AC	A	AC
D	AC	D	AC	ACD	AC	ACD
E	AC	E	AC	ACE	AC	ACE
F	∅	F	AC	ACF	AC	ACF
G	∅	G	A	AG	A	AG

Non serve fare una quarta iterazione perché si è verificata la stabilità in due iterazioni successive, quindi l'algoritmo si fermerà.

Terzo esercizio: Constant Propagation



1. Derivare una formalizzazione per il framework di Dataflow Analysis, riempiendo lo specchietto coi parametri adeguati.

	Dataflow Problem Dominator Analysis
Domain	Sets of <variable, constant value>
Direction	Forward $out[b] = f_b(in[b])$ $in[b] = \wedge out[prec(b)]$

Transfer function	$f_b(x) = \text{GEN}_b \cup \{x - \text{KILL}_b\}$
Meet Operation (^)	\cap
Boundary Condition	$\text{out}[\text{entry}] = \mu$
Initial interior points	$\text{out}[b_i] = \mu$

L'insieme μ è un artefatto per indicare un'assenza di conflitto nella definizione del valore delle variabili. Figura come lo stato 'ignoto' dove si trova il sistema all'avvio.

GEN_b è il LHS di un'espressione dove nessuno degli operandi è la variabile stessa, e dove questi hanno valori costanti noti al momento del calcolo.

KILL_b è la definizione di una variabile fino ad allora avente un valore noto.

2. Per il CFG di esempio fornito popolare una tabella con le iterazioni dell'algoritmo iterativo di soluzione del problema.

NOTA: Non useremo vector/liste per semplificare la notazione. Le tuple sono considerate nel loro insieme per l'operazione di intersezione.

Non serve fare una quinta iterazione perché si è verificata la stabilità in due iterazioni successive, quindi l'algoritmo si fermerà.

	ITERAZIONE 1		ITERAZIONE 2		ITERAZIONE 3		ITERAZIONE 4	
	in	out	in	out	in	out	in	out
Entry	-	μ	-	-	-	-	-	-
1	μ	μ	-	(k,2)	-	(k,2)	-	(k,2)
2	μ	μ	(k,2)	(k,2)	(k,2)	(k,2)	(k,2)	(k,2)
3	μ	μ	(k,2)	(k,2) (a,4)	(k,2)	(k,2) (a,4)	(k,2)	(k,2) (a,4)
4	μ	μ	(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)
5	μ	μ	(k,2)	(k,2) (a,4)	(k,2)	(k,2) (a,4)	(k,2)	(k,2) (a,4)
6	μ	μ	(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)
7	μ	μ	(k,2) (a,4)	(k,4) (a,4)	(k,2) (a,4)	(k,4) (a,4)	(k,2) (a,4)	(k,4) (a,4)
8	μ	μ	(k,4) (a,4)	(k,4) (a,4)	(a,4)	(a,4)	(a,4)	(a,4)
9	μ	μ	(k,4) (a,4)	(b,2) (k,4) (a,4)	(a,4)	(a,4) (b,2)	(a,4)	(a,4) (b,2)
10	μ	μ	(b,2) (k,4) (a,4)	(b,2) (k,4) (a,4) (x,8)	(a,4) (b,2)	(a,4) (b,2)	(a,4) (b,2)	(a,4) (b,2)
11	μ	μ	(b,2) (k,4) (a,4) (x,8)	(b,2) (k,4) (a,4) (x,8) (y,8)	(a,4) (b,2)	(a,4) (b,2) (y,8)	(a,4) (b,2)	(a,4) (b,2) (y,8)
12	μ	μ	(b,2) (k,4) (a,4) (x,8) (y,8)	(b,2) (a,4) (x,8) (y,8) (k,5)	(a,4) (b,2) (y,8)	(a,4) (b,2) (y,8)	(a,4) (b,2) (y,8)	(a,4) (b,2) (y,8)
13	μ	μ	(k,4) (a,4)	(k,4) (a,4)	(a,4)	(a,4)	(a,4)	(a,4)
Exit	μ	μ	(k,4) (a,4)	(k,4) (a,4)	(a,4)	(a,4)	(a,4)	(a,4)