



UNIMORE
UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

Dipartimento di Scienze Fisiche,
Informatiche e Matematiche

8. Secondo Assignment - Dataflow analysis

Compilatori – Middle end [I215-014]

Corso di Laurea in INFORMATICA
(D.M.270/04) [16-215]
Anno accademico 2024/2025

Prof. Andrea Marongiu
andrea.marongiu@unimore.it

Dataflow Analysis Assignment

Per ciascuno dei seguenti tre problemi di analisi

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

	Dataflow Problem X
Domain	Set of Expression
Direction	Backward $in[b] = f_b(out[b])$ $out[b] = 1 in[succ[b]]$
Transfer function	$in[b] = Gen_b \cup (out[b] \text{ kill } b)$
Meet Operation (\wedge)	\cap
Boundary Condition	$in[exit] = \emptyset$
Initial interior points	$in[b] = \cup$ (universal set)

espressioni generate dalle istruzioni nel BB

ogni assegnamento alla espressione in cui la variabile è un operando

Dataflow Analysis Assignment

Per ciascuno dei seguenti tre problemi di analisi

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[B]	OUT[B]	IN[B]	OUT[B]	IN[B]	OUT[B]
BB1	< ... >	< ... >				
BB2						
BB3						

Vettore di bit con le seguenti posizioni: e_1 e_2

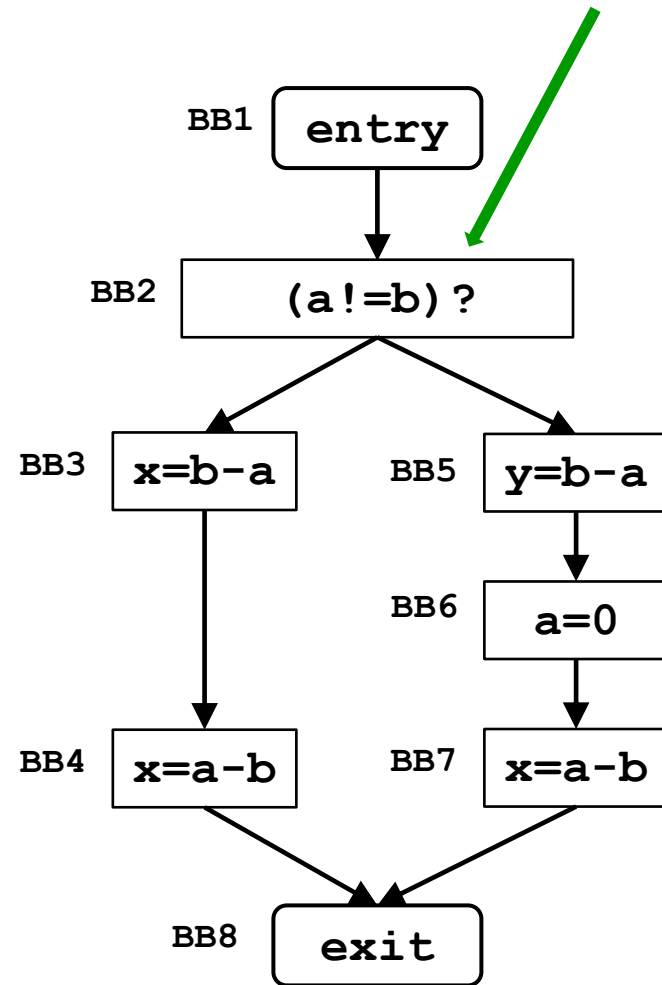
	INCBJ	OUTCBJ
DB1(ENTRY)	/	10
BB2	10	10
BB3	11	01
BB4	01	Ø
BB5	10	00
SB6	00	01
BB7	01	Ø
BB8(EXIT)	Ø	/

BB	Gen	util
1	Ø	Ø
2	Ø	Ø
3	e_1	Ø
4	e_2	Ø
5	e_1	Ø
6	Ø	e_1, e_2
7	e_2	Ø
8	Ø	Ø

1) Very Busy Expressions

Quali espressioni sono **very busy** in questo punto?

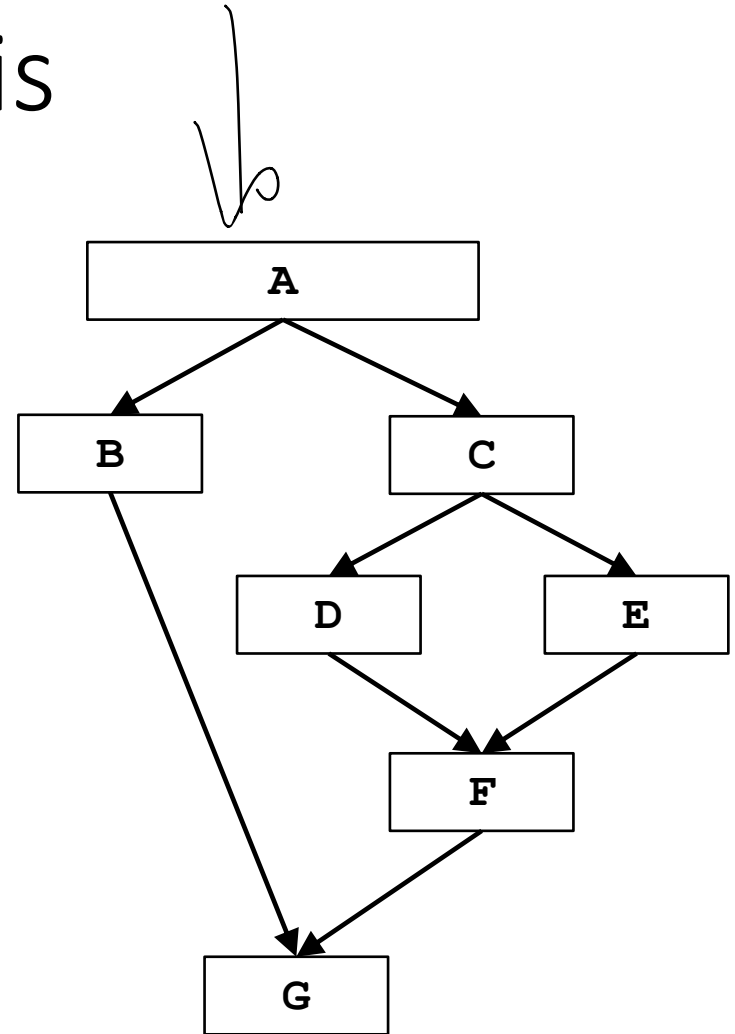
- Un'espressione è **very busy** in un punto p se, indipendentemente dal percorso preso da p , l'espressione viene usata prima che uno dei suoi operandi venga definito.
- Un'espressione $a+b$ è **very busy** in un punto p se $a+b$ è valutata in tutti i percorsi da p a EXIT e non c'è una definizione di a o b lungo tali percorsi
 - Ci interessa l'insieme di espressioni disponibili (available) all'inizio del blocco B
 - L'insieme dipende dai percorsi che cominciano al punto p prima di B



ENABLES CODE HOISTING

2) Dominator Analysis

- In un CFG diciamo che un nodo X **domina** un altro nodo Y se il nodo X appare in ogni percorso del grafo che porta dal blocco ENTRY al blocco Y
- Annotiamo ogni *basic block* B_i con un insieme $DOM[B_i]$
 - $B_i \in DOM[B_j]$ se e solo se B_i domina B_j
- Per definizione un nodo domina sé stesso
 - $B_i \in DOM[B_i]$



$DOM[F] = \{A, C, F\}$

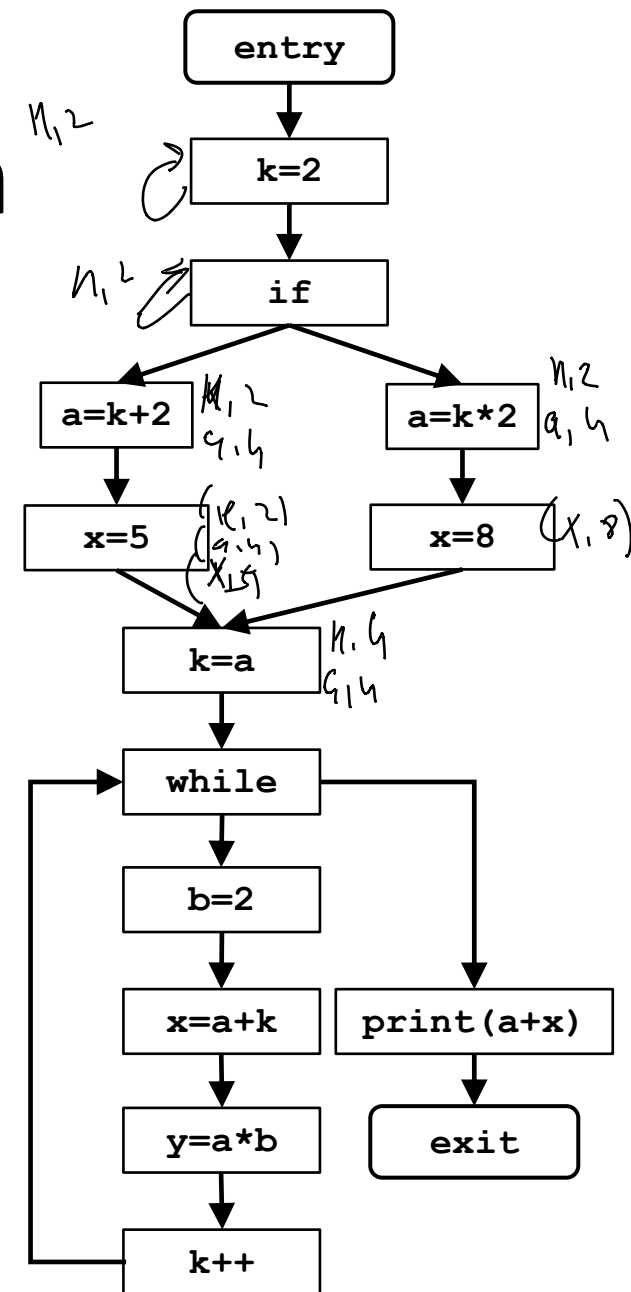
	Dataflow Problem X
Domain	Sets of BB
Direction	Forward $out(b) = f_b(in(b))$ $in(b) = \bigwedge out(pred(b))$
Transfer function	$out(b) = B \cup in(b)$
Meet Operation (\wedge)	\cap
Boundary Condition	$out(entry) = ENTRY$
Initial interior points	$out(b) = \text{universal set}$

BB	IN	OUT
A	\emptyset	$\{A\}$
B	$\{A\}$	$\{A, B\}$
C	$\{A\}$	$\{A, C\}$
D	$\{A, C\}$	$\{A, C, D\}$
E	$\{A, C\}$	$\{A, C, E\}$
F	$\{A, C\}$	$\{A, C, F\}$
G	$\{A\}$	$\{A, G\}$

Dobbiamo aggiungere Entry e Exit che è stato se non

3) Constant Propagation

- L'obiettivo della *constant propagation* è quello di determinare in quali punti del programma le variabili hanno un valore costante.
- L'informazione da calcolare per ogni nodo n del CFG è un insieme di **coppie** del tipo *<variabile, valore costante>*.
- Se abbiamo la coppia *<x, c>* al nodo n , significa che x è garantito avere il valore c ogni volta che n viene raggiunto durante l'esecuzione del programma.

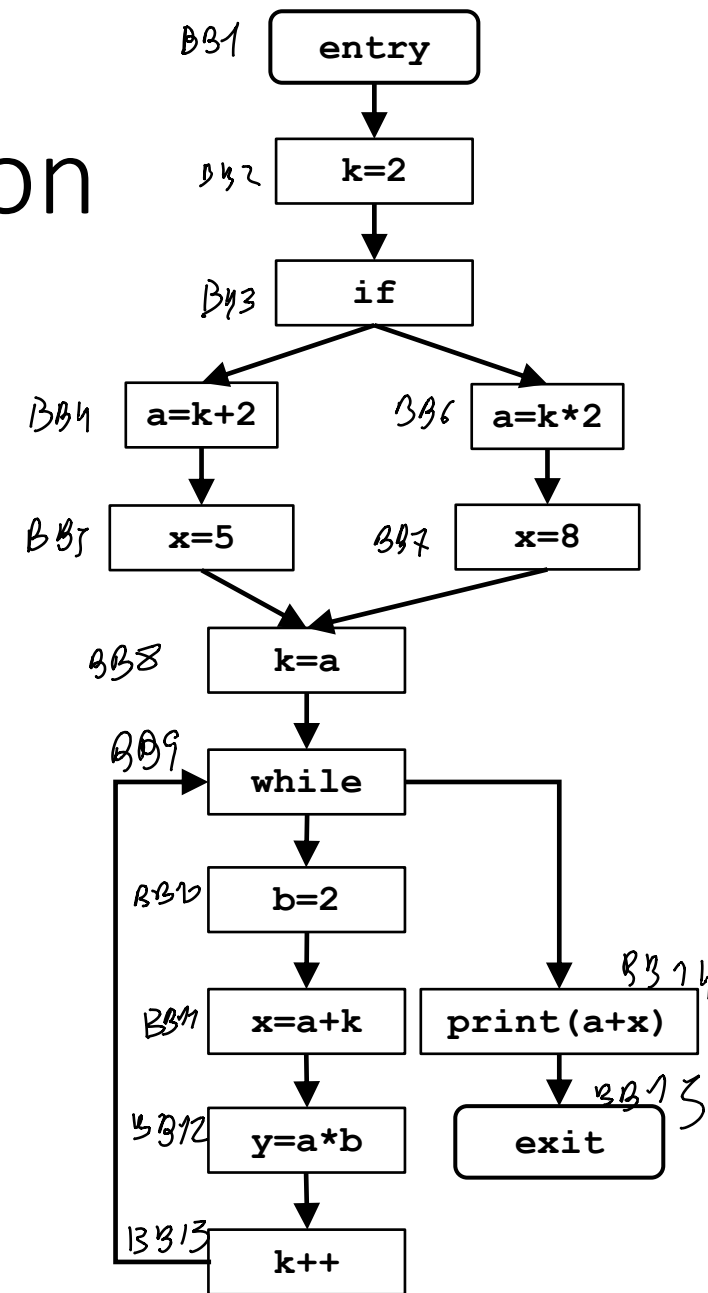


3) Constant Propagation

- **NOTA:** L'analisi di CP riesce a determinare il valore costante di espressioni binarie in cui uno o entrambi gli operandi siano delle variabili il cui valore costante sia noto:

- $w = 5$
- $x = 12$
- $y = x - 2 \rightarrow y = 10$
- $z = w + x \rightarrow z = 17$

- Tenere conto di questo aspetto nel determinare le equazioni



	Dataflow Problem X
Domain	$(var, valore esterne)$
Direction	Forward $out(b) = f6(in(b))$ $in(b) \subseteq 1 \text{ outpred}(b)$
Transfer function	$out(b) = Gen_b \cup (in(b) - kill_b)$
Meet Operation (\wedge)	\cap
Boundary Condition	$out(Entry) = \emptyset$
Initial interior points	$out(b) = \text{universal set}$

NO BIT VECTOR
 ABBONDIANTISSIMO

quelli sono veri

e il solo se eliminano i - in pr una costante

BB	GEN	UILL
1	\emptyset	\emptyset
2	$\{k, 2\}$	$\{u, c\}$ con $c \neq 2$
3	\emptyset	\emptyset
4	$\{a, k, 2\}$	$\{a, c\}$ con $c \neq k+2$
5	$\{x, 5\}$	$\{x, c\}$ con $c \neq 5$
6	$\{a, k+2\}$	$\{a, c\}$ con $c \neq k+2$
7	$\{k, 2\}$	$\{x, c\}$ con $c \neq 8$
8	$\{k, a\}$	$\{k, c\}$ con $c \neq a$
9	\emptyset	\emptyset
10	$\{b, 2\}$	$\{b, c\}$ con $c \neq 2$
11	$\{k, a+k\}$	$\{x, c\}$ con $c \neq a+k$
12	$\{y, a+b\}$	$\{y, c\}$ con $c \neq a+b$
13	$\{u, u+1\}$	$\{x, c\}$ con $c \neq k+1$
14	\emptyset	\emptyset
15	\emptyset	\emptyset

interpol come
 tutte le variabili
 con gli valori costanti

II ITERAZIONE

BB	IN	OUT
1	/	\emptyset
2	\emptyset	$\{c, 2\}$
3	$\{u, 2\}$	$\{u, 2\}$
4	$\{u, 2\}$	$\{a, u\}, \{k, 2\}$
5	$\{a, u\}, \{u, 2\}$	$\{a, u\}, \{k, 2\}, \{x, 5\}$
6	$\{k, 2\}$	$\{a, u\}, \{u, 2\}$
7	$\{a, u\}, \{k, 2\}$	$\{a, u\}, \{u, 2\}, \{x, 8\}$
8	$\{a, u\}, \{k, 2\}$	$\{a, u\}, \{k, u\}$
9	$\{a, u\}, \{u, u\}$	$\{a, u\}, \{u, u\}$
10	$\{a, u\}, \{u, u\}$	$\{a, u\}, \{k, u\}, \{b, 2\}$
11	$\{a, u\}, \{u, u\}, \{b, 2\}$	$\{k, 8\}, \{a, u\}, \{k, u\}, \{b, 2\}$
12	$out(11)$	$\{y, 8\}, \{x, 8\}, \{a, u\}, \{k, u\}, \{b, 2\}$
13	$out(12)$	$\{k, 5\}, \{y, 8\}, \{x, 8\}, \{a, u\}, \{b, 2\}$
14	$out(13)$	$out(14)$
15	$out(14)$	$out(15)$

II

III IT.

VBUTZ R → CONNESSIONE

IN	OUT
9 $out(10) \cap out(8) = \{a, u\}$	$\{a, u\}$
10 $\{a, u\}$	$\{b, 2\}, \{a, u\}$
11 $\{b, 2\}, \{a, u\}$	$\{b, 2\}, \{a, u\}$
12 $\{b, 2\}, \{a, u\}$	$\{b, 2\}, \{a, u\}, \{y, 8\}$
13 $out(12)$	$in(13)$
14 $out(13)$	$out(14)$
15 $out(14)$	$out(15)$

Deadline per la consegna

- La deadline per la consegna del secondo assignment è martedì 15 aprile 2025
- Usate preferibilmente lo stesso link già comunicato per il primo assignment, organizzando il vostro repository in cartelle strutturate per assignment