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

Copyright note

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	?
Direction	? ? ?
Transfer function	?
Meet Operation ($\wedge$)	?
Boundary Condition	?
Initial interior points	?

Dataflow Analysis Assignment

Per ciascuno dei seguenti tre problemi di analisi

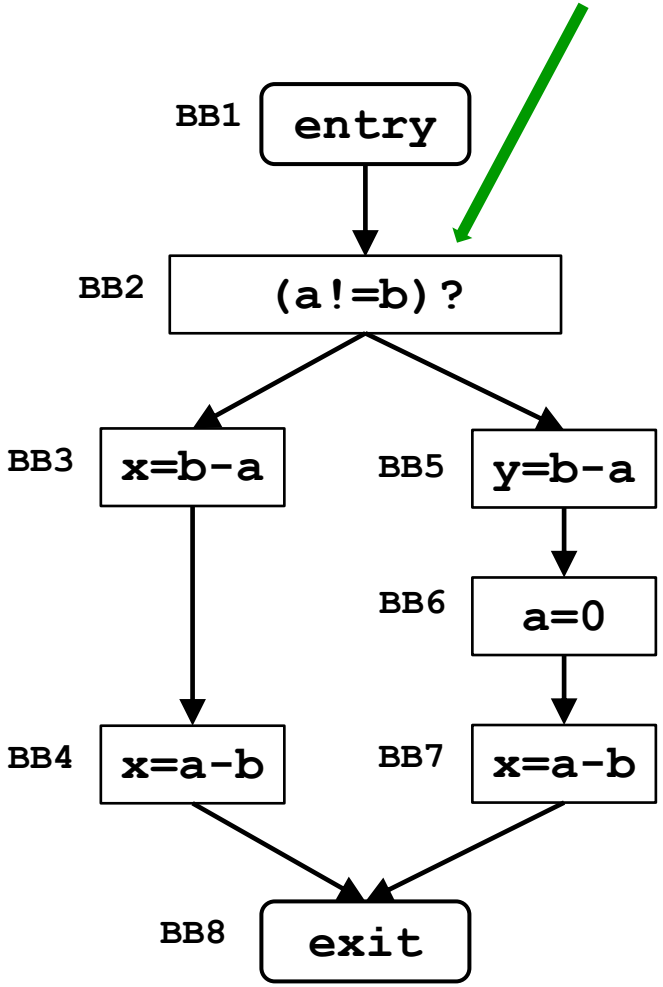
2. Per il CFG di esempio fornito popolare una tabella con le iterazioni dell'algoritmo iterativo di soluzione del problema

	Iterazione 1		**Iterazione 2**		**Iterazione 3**	
	IN[B]	OUT[B]	IN[B]	OUT[B]	IN[B]	OUT[B]
BB1	< … >	< … >				
BB2						
BB3						

1) Very Busy Expressions

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

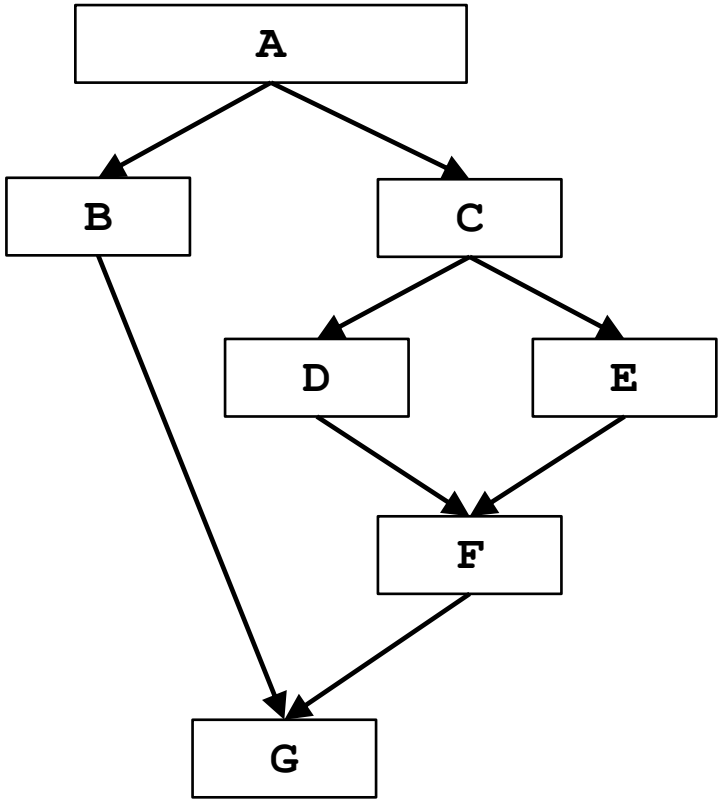
- Un'espressione è ***very busy*** in un punto *p* se, indipendentemente dal percorso preso da *p*, l'espressione viene usata prima che uno dei suoi operandi venga definito.
- Un'espressione *a+b* è **very busy** in un punto *p* se *a+b* è valutata in tutti i percorsi da *p* a EXIT e non c'è una definizione 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

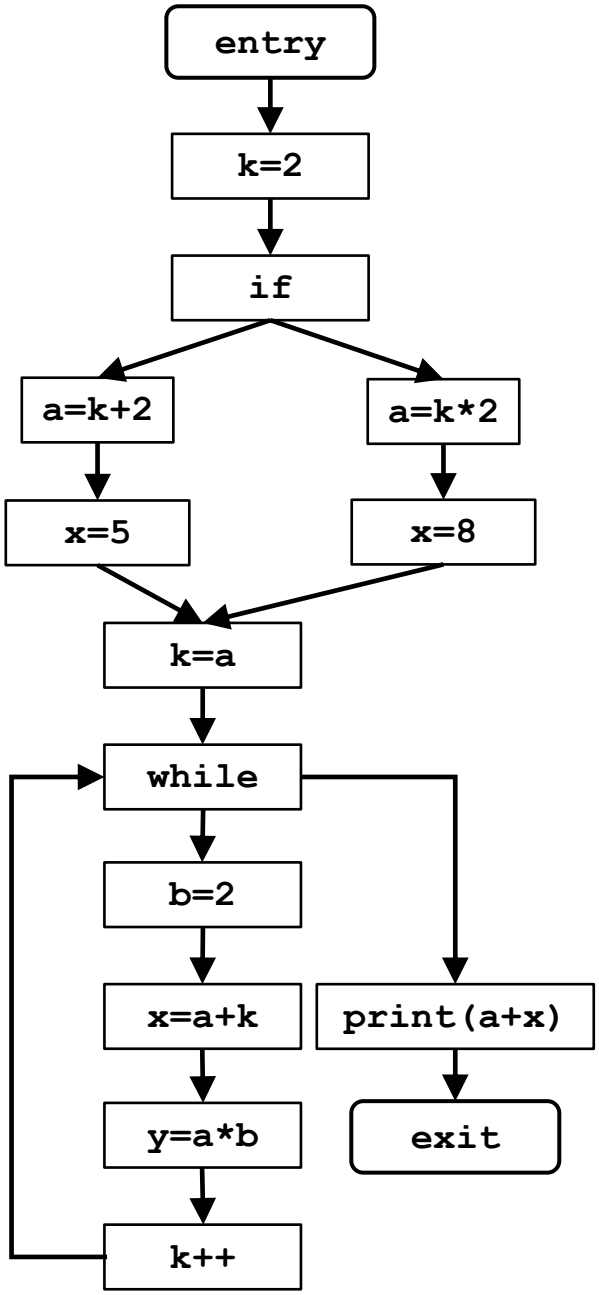
- In un CFG diciamo che un nodo *X* **domina** un altro nodo *Y* se il nodo *X* appare in ogni percorso del grafo che porta dal blocco ENTRY al blocco *Y*
- Annotiamo ogni *basic block* Bi con un insieme DOM[Bi]
  - Bi $\in$ DOM[Bj] se e solo se Bi domina Bj
- Per definizione un nodo domina sé stesso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



`DOM[F] = {A,C,F}`

3) Constant Propagation

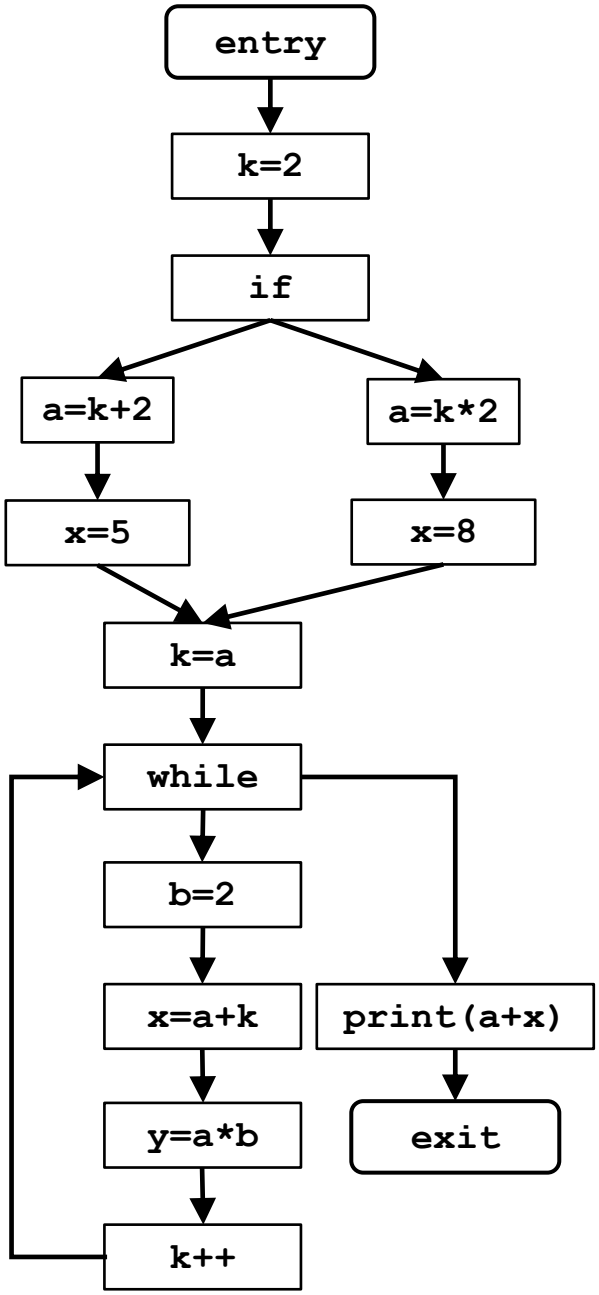
- L'obiettivo della *constant propagation* è quello di determinare in quali punti del programma le variabili hanno un valore costante.
- L'informazione da calcolare per ogni nodo *n* del CFG è un insieme di ***coppie*** del tipo *<variabile, valore costante>*.
- Se abbiamo la coppia *<x, c>* al nodo *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` → **`y = 10`**
  - `z = w + x` → **`z = 17`**

- Tenere conto di questo aspetto nel determinare le equazioni



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