

Assignment - Dominator Analysis

In un *CFG* si definisce *dominante* un nodo X su un altro nodo Y se il nodo X appare in ogni percorso che porta dal blocco *EXIT* al blocco Y ; verificata questa condizione, è possibile scrivere $X \text{ dom } Y$.

La **Dominator Analysis** identifica quali istruzioni di un programma *dominano* altre istruzioni; data una sequenza di istruzioni, determina quali istruzioni devono essere eseguite prima affinché le altre possano essere eseguite.

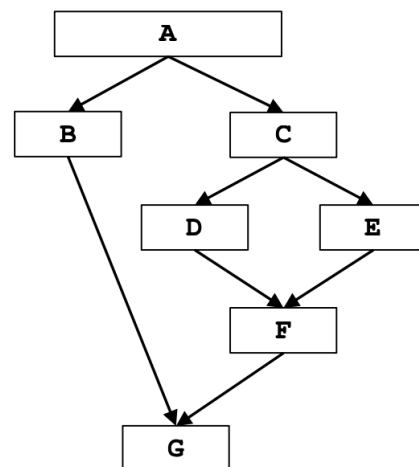
È importante ricordare che, per definizione, un nodo domina sé stesso: $B_i \in \text{DOM}[B_i]$.

Risulta utile, inoltre, rappresentare le relazioni di dominanza mediante un albero, il **dominance tree**.

L'esistenza di un **dominance tree** discende dalla una proprietà dei nodi dominanti: ogni nodo dominante n ha un unico dominante immediato m , cioè l'ultimo dominante di n lungo qualsiasi percorso dal nodo iniziale a n .

Si dice che m domina immediatamente n se $m \text{ dom } n$ AND $m \neq n$.

Le caratteristiche del DFA sono riassunte nella tabella sottostante:



Dominator Analysis

	<i>Dominator Analysis framework</i>
Domain	Basic Blocks
Direction	Forward
Transfer Function	$f_b(x) = Gen_b \cup (x - Kill_b)$
Meet Operation	\cap
Boundary Conditions	$in[ENTRY] = ENTRY$
Initial Interior Points	$out[b] = U$
Equations	$in[b] = \cap out[pred[b]]$ $out[b] = f_b(in[b])$

	<i>IN</i>	<i>OUT</i>
A	0000000	1000000
B	1000000	1100000
C	1000000	1010000
D	1010000	1011000
E	1010000	1010100
F	1010000	1010010
G	1000000	1000001

	<i>IN</i>	<i>OUT</i>
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	$\{A\}$	$\{A, G\}$

Assumendo un **bit vector** di dimensione pari al numero di **basic block** nel *CFG*, e considerando che ogni nodo è dominante di sé stesso, è possibile capire quali nodi sono dominanti per altri. I risultati dell'algoritmo sono dapprima mostrati in rappresentazione **bit vector** poi mediante gli insiemi, in formato più semplice.