1. VERY BUSY EXPRESSION

	Very Busy Expression
Domain	Insieme delle espressioni
Direction	backward: in[b] = fb (out[b]) out[b] = ∧ in[succ(b)]
Transfer function	fb = Genb U (out[b] - kill[b])
Meet Operation (^)	Ω
Boundary Condition	IN[exit] = Ø
Initial interior points	IN[b] = Ø

	1° Iterazione	
	IN[B]	OUT[B]
BB8	Ø	Ø
BB7	{a-b}	Ø
BB6	Ø	{a-b}
BB5	{b-a}	Ø
BB4	{a-b}	Ø
BB3	{b-a},{a-b}	{a-b}
BB2	{b-a}	{b-a}
BB1	{b-a}	{b-a}

Aggiungo anche una tabella con i Kill e le Gen dei vari basic block

	Gen[]	Kill[]
BB1	Ø	Ø
BB2	Ø	Ø
BB3	{b-a}	Ø
BB4	{a-b}	Ø
BB5	{b-a}	Ø
BB6	Ø	{a-b}
BB7	{a-b}	Ø
BB8	Ø	Ø

2. DOMINATOR ANALYSIS

	Dominator Analysis
Domain	Insieme di basic block
Direction	Forward: out[b] = fb (in[b]) in[b] = \(\) out[pred(b)]
Transfer function	fb = Genb U (out[b])
Meet Operation (^)	Ω
Boundary Condition	OUT(entry) = entry
Initial interior points	OUT[b] = b

	1° Iterazione	
	IN[B]	OUT[B]
BBA	Ø	А
BBB	А	A,B
BBC	А	A,C
BBD	A,C	A,C,D
BBE	A,C	A,C,E
BBF	A,C,D,E	A,C,F
BBG	A,C,F,B	A,G

3. Constant Propagation

	Constant Propagation	
Domain	Insieme di coppie [variabile, valore], dove variabile = Z U {N} N(null) = valore indeterminabile	
Direction	Forward: out[b] = fb (in[b]) in[b] = ∧ out[pred(b)]	
Transfer function	Per ogni BB, la fb aggiorna lo stato s [variabile, valore]: 1) Assegnamento diretto: x = c -> s' = s [x,c] 2) Assegnamento con expr: Se tutti gli operandi di expr sono costanti in s, calcola expr ed aggiorna s, altrimenti s[x, N] 3) Ridefinizioni non costanti: k++, k allora s [x,N]	
Meet Operation (^)	Interesezione punto a punto (∩): Se due valori differiscono, la variabili diventa	
Boundary Condition	IN[entry] = Ø	

Initial interior points	OUT[b] = T (Tutte le variabili iniziali non costanti)
-------------------------	-------------------------------------------------------

	1° Iterazione	
	IN[B]	OUT[B]
BB1	Ø	Ø
BB2	Ø	<k,2></k,2>
BB3	<k,2></k,2>	<k,2></k,2>
BB4	<k,2></k,2>	<k,2>,<a,4></a,4></k,2>
BB5	<k,2>,<a,4></a,4></k,2>	<k,2>,<a,4>, <x,5></x,5></a,4></k,2>
BB6	<k,2></k,2>	<k,2>,<a,4></a,4></k,2>
BB7	<k,2>,<a,4></a,4></k,2>	<k,2>,<a,4>,<x,8></x,8></a,4></k,2>
BB8	<k,2>,<a,4>,<x,8> ,<x,5></x,5></x,8></a,4></k,2>	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>
BB9	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>
BB10	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>
BB11	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>

	2° Iterazione	
	IN[B]	OUT[B]
BB1	Ø	Ø
BB2	Ø	<k,2></k,2>
BB3	<k,2></k,2>	<k,2></k,2>
BB4	<k,2></k,2>	<k,2>,<a,4></a,4></k,2>
BB5	<k,2>,<a,4></a,4></k,2>	<k,2>,<a,4>, <x,5></x,5></a,4></k,2>
BB6	<k,2></k,2>	<k,2>,<a,4></a,4></k,2>
BB7	<k,2>,<a,4></a,4></k,2>	<k,2>,<a,4>,<x,8></x,8></a,4></k,2>
BB8	<k,2>,<a,4>,<x,8> ,<x,5></x,5></x,8></a,4></k,2>	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>
BB9	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>
BB10	<k,4>,<a,4>,<x,n></x,n></a,4></k,4>	<k,4>,<a,4>,<x,n>, <b,2></b,2></x,n></a,4></k,4>
BB11	<k,4>,<a,4>,<x,n>, <b,2></b,2></x,n></a,4></k,4>	<k,4>,<a,4>,<x,8>, <b,2></b,2></x,8></a,4></k,4>
BB12	<k,4>,<a,4>,<x,8>,<b,2></b,2></x,8></a,4></k,4>	<k,4>,<a,4>,<x,8>, <b,2>,<y,8></y,8></b,2></x,8></a,4></k,4>
BB13	<k,4>,<a,4>,<x,8>, <b,2>,<y,8></y,8></b,2></x,8></a,4></k,4>	<k,n>,<a,4>,<x,8>, <b,2>,<y,8></y,8></b,2></x,8></a,4></k,n>