

Very Busy Expression

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.

Ogni blocco genera delle espressioni, e se ne definisce uno degli operandi, la uccide. Allo stesso modo, ogni blocco riceve delle espressioni definite in precedenza e potrebbe ucciderle definendone gli operandi.

	Very Busy Expression
Domain	Sets of Expressions
Direction	Backward:: $IN[B] = f_B(IN[B])$ $OUT[B] = \bigwedge IN[succ(B)]$
Transfer Function	$f_B(x) = Use_B \cup (x - Def_B)$ Use_B tutte le espressioni e che appaiono in B Def_B se B definisce t , si uccidono tutte le espressioni che contengono t
Meet Operation	\cap
Boundary Condition	$OUT[entry] = \emptyset$
Initial interior points	$OUT[B] = U$

Basic Block	Iter1	
	IN[BB]	OUT[BB]
BB1	a-b, b-a	a-b, b-a
BB2	a-b, b-a	a-b, b-a
BB3	a-b, b-a	a-b
BB4	a-b	\emptyset
BB5	b-a	\emptyset
BB6	\emptyset	a-b
BB7	a-b	\emptyset
BB8	\emptyset	\emptyset

Dominator Analysis

Ipotizzo che il nodo A sia entry, e il nodo G sia exit

	Dominator Analysis
Domain	Sets of Blocks
Direction	Forward $OUT[B] = f_B(IN[B])$ $IN[B] = \bigwedge OUT[pred(B)]$
Transfer Function	$f_B(x) = Gen[B] \cup (IN[B] - Kill[B])$ Gen(B) sono i blocchi dominatori Kill(B) sono i
Meet Operation	\cap
Boundary Condition	$IN[entry] = \emptyset$
Initial interior points	$IN[B] = \cup$

Basic Block	Iter1	
	IN[BB]	OUT[BB]
A	\emptyset	A
B	A	A, B
C	A	A, C
D	A, C	D, C, A
E	A, C	E, C, A
F	$OUT[D] \cap$ $OUT[E] = \{A, C\}$	F, C, A
G	$OUT[B] \cap$ $OUT[F] = \{A\}$	A

Constant Propagation

	Constant Propagation
Domain	Sets of variables
Direction	Forward: $IN[B] = \bigwedge OUT[pred(B)]$ $OUT[B] = f_B(IN[B])$
Transfer Function	$f_B(x) = Gen_B \cup (x - Kill_B)$ Gen_B è l'insieme di variabili dal valore costante definite nel blocco $Kill_B$ è l'insieme di variabili che vengono definite come non costanti nel blocco
Meet Operation	\cap
Boundary Condition	$IN[entry] = \emptyset$
Initial interior points	$IN[B] =$



