

# 1 Alcançabilidade de Definições

*Gen* ...

*Kill* ...

*IN* ...

*OUT* ...

	Gen	Kill	IN	OUT
$B_1$	11000000000	11101011000	00000000000	11000000000
$B_2$	00110000000	01110101011	11000000000	10110000000
$B_3$	00000000000	00000000000	10110000000	10110000000
$B_4$	00001000000	10001010000	11000000000	01001000000
$B_5$	00000111100	11111111111	11111011110	00000111100
$B_6$	00000000010	00010100011	00000111100	00000011110
$B_7$	00000000001	00010100011	10110011110	10100011101
$B_8$	00000000000	00000000000	10110000000	10110000000
$B_9$	00000000000	00000000000	01001000000	01001000000
$B_{10}$	00000000000	00000000000	00000011110	00000011110
$B_{11}$	00000000000	00000000000	10110000000	10110000000
$B_{12}$	00000000000	00000000000	00000011110	00000011110

Table 1: Alcançabilidade de Definições —  $(a := 4, b := 5, b := 5, c := a + b, a := 4, c := a + b, a := 5, b := c, d := b, c := a + b, c := a + b)$

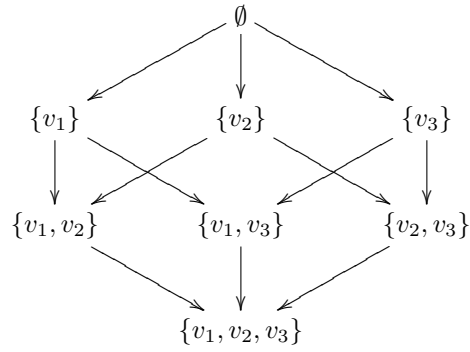
## 2 Vivacidade de Variáveis

*Gen* ...

*Kill* ...

*IN* ...

*OUT* ...



	Gen	Kill	IN	OUT
$B_1$	0001	1100	0001	1101
$B_2$	1001	0110	1001	1100
$B_3$	0000	0000	1100	1100
$B_4$	0000	1000	0100	1100
$B_5$	1100	1111	1100	1101
$B_6$	1101	0010	1101	1100
$B_7$	1100	0010	1100	0000
$B_8$	0000	0000	1100	1100
$B_9$	0000	0000	1100	1100
$B_{10}$	0000	0000	1100	1100
$B_{11}$	0000	0000	1100	1100
$B_{12}$	0000	0000	1100	1100

Table 2: Vivacidade de Variáveis —  $(a, b, c, d)$

### 3 Disponibilidade de Expressões

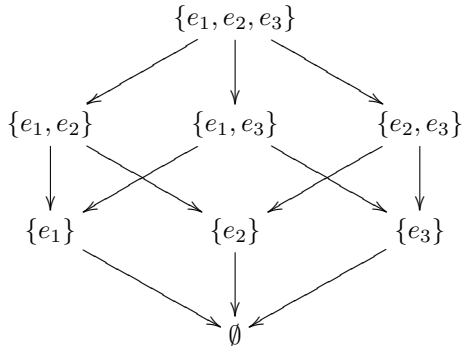
A análise é para frente (*forward*) e sua intenção é determinar em cada ponto do código, quais expressões estão disponíveis, isto é, foram seguramente executadas e, caso fossem executadas novamente (naquele ponto) produziriam o mesmo resultado.

*Gen* Indica quais expressões foram geradas dentro do bloco e que não foram “mortas” por redefinições de seus operandos dentro do mesmo bloco. **É igual à entrada das expressões antecipáveis.**

*Kill* Indica quais expressões (considerando o universo inteiro) foram mortas por redefinições (posteriores <sup>1</sup>) de seus operandos que ocorrem dentro do bloco.

*IN* Indica quais expressões estão disponíveis na entrada do bloco. É uma interseção das saídas dos blocos predecessores.

*OUT* Indica quais expressões estão disponíveis na saída do bloco. **É igual ao último *Gen*.**



	Gen	Kill	IN	OUT
$B_1$	0	1	0	0
$B_2$	1	1	0	1
$B_3$	0	0	1	1
$B_4$	0	1	0	0
$B_5$	0	1	0	0
$B_6$	1	0	0	1
$B_7$	1	0	1	1
$B_8$	0	0	1	1
$B_9$	0	0	0	0
$B_{10}$	0	0	1	1
$B_{11}$	0	0	1	1
$B_{12}$	0	0	1	1

Table 3: Disponibilidade de Expressões —  $((+, a, b))$

<sup>1</sup>Só faz sentido em análises internas ao bloco.

## 4 Disponibilidade de Expressões Antecipáveis

A análise é para trás (**backward**) e sua intenção é determinar em cada ponto do código, quais expressões podem ser movidas para o início do bloco (ou para blocos antecedentes).

*Gen* Indica quais expressões podem ser movidas para o início do bloco (ou para blocos antecedentes).

*Kill* Indica quais expressões (considerando o universo inteiro) foram mortas por redefinições (anteriores <sup>2</sup>) de seus operandos que ocorrem dentro do bloco.

*IN* Indica quais expressões podem ser movidas para blocos antecedentes.

*OUT* Indica quais expressões de blocos subsequentes podem ser movidas para o final do bloco atual – estas expressões podem ou não serem antecipadas pelo bloco atual.

	Gen	Kill	IN	OUT
$B_1$	0	1	0	0
$B_2$	0	1	0	1
$B_3$	0	0	1	1
$B_4$	0	1	0	1
$B_5$	1	1	1	1
$B_6$	1	0	1	1
$B_7$	1	0	1	0
$B_8$	0	0	1	1
$B_9$	0	0	1	1
$B_{10}$	0	0	1	1
$B_{11}$	0	0	1	1
$B_{12}$	0	0	1	1

Table 4: Disponibilidade de Expressões Antecipáveis —  $((+, a, b))$

---

<sup>2</sup>Só faz sentido em análises internas ao bloco.

## 5 Disponibilidade Parcial de Expressões

*Gen* ...

*Kill* ...

*IN* ...

*OUT* ...

	Gen	Kill	IN	OUT
$B_1$	0	1	0	0
$B_2$	1	1	0	1
$B_3$	0	0	1	1
$B_4$	0	1	0	0
$B_5$	0	1	1	0
$B_6$	1	0	0	1
$B_7$	1	0	1	1
$B_8$	0	0	1	1
$B_9$	0	0	0	0
$B_{10}$	0	0	1	1
$B_{11}$	0	0	1	1
$B_{12}$	0	0	1	1

Table 5: Disponibilidade Parcial de Expressões —  $((+, a, b))$

## 6 Mortalidade de Variáveis

*Gen ...*

*Kill ...*

*In ...*

*In ...*

	Gen	Kill	IN	OUT
$B_1$	1100	0001	1110	0110
$B_2$	0110	1001	0110	0011
$B_3$	0000	0000	0011	0011
$B_4$	1000	0000	1111	1111
$B_5$	1111	1100	1111	0010
$B_6$	0010	1101	0010	0011
$B_7$	0010	1100	0011	1111
$B_8$	0000	0000	1111	1111
$B_9$	0000	0000	1111	1111
$B_{10}$	0000	0000	1111	1111
$B_{11}$	0000	0000	0011	0011
$B_{12}$	0000	0000	0011	0011

Table 6: Mortalidade de Variáveis —  $(a, b, c, d)$

## 7 Alcançabilidade de Definições para Propagação de Cópias

*Gen ...*

*Kill ...*

*In ...*

*In ...*

	Gen	Kill	IN	OUT
$B_1$	11000000000	11100000000	00000000000	11000000000
$B_2$	00100000000	01100000000	11000000000	10100000000
$B_3$	00000000000	00000000000	10100000000	10100000000
$B_4$	00000000000	10000000000	11000000000	01000000000
$B_5$	00000000000	11100000000	00000000000	00000000000
$B_6$	00000000000	00000000000	00000000000	00000000000
$B_7$	00000000000	00000000000	00000000000	00000000000
$B_8$	00000000000	00000000000	10100000000	10100000000
$B_9$	00000000000	00000000000	01000000000	01000000000
$B_{10}$	00000000000	00000000000	00000000000	00000000000
$B_{11}$	00000000000	00000000000	10100000000	10100000000
$B_{12}$	00000000000	00000000000	00000000000	00000000000

Table 7: Alcançabilidade de Definições para Propagação de Cópias — ( $a := 4, b := 5, b := 5, c := a + b, a := 4, c := a + b, a := 5, b := c, d := b, c := a + b, c := a + b$ )

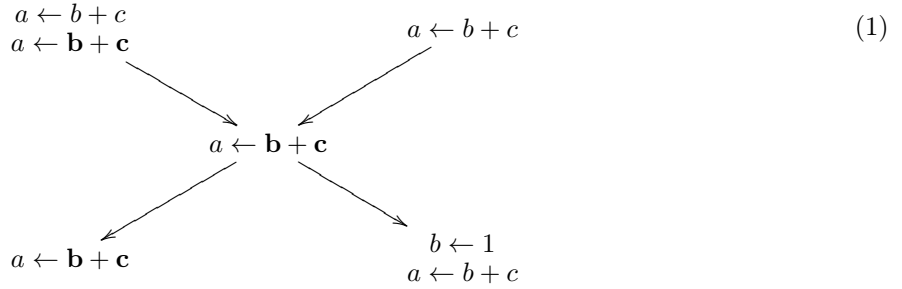
## 8 Eliminação de Redundâncias Parciais

### 8.1 Expressão Redundante

Uma expressão é *redundante* no ponto  $p$  se em cada caminho até  $p$ :

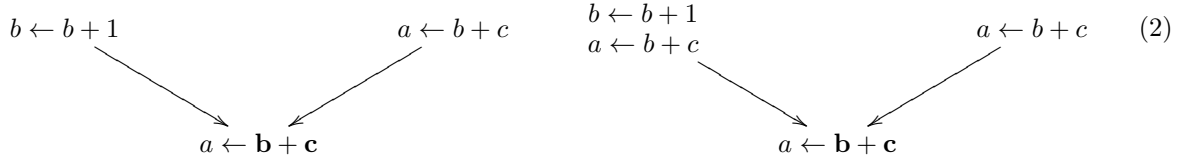
1. Ela é avaliada antes de alcançar  $p$ , e
2. Nenhum de seus operandos constituintes é redefinido antes de  $p$ .

Por exemplo, na Equação 1, as ocorrências de expressões em **negrito** são redundantes.



Uma expressão é *parcialmente redundante* no ponto  $p$  se ela é redundante ao longo de alguns caminhos, mas não todos, até  $p$ .

Por exemplo, na Equação 2, a expressão  $b + c$  em negrito no diagrama da esquerda é parcialmente redundante. A inserção de uma cópia de  $b + c$  depois da definição de  $b$  pode tornar uma expressão parcialmente redundante em uma totalmente redundante como mostra o diagrama da direita.



	ENTRY	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$	$B_8$	$B_9$	$B_{10}$	$B_{11}$	$B_{12}$	$B_{13}$	EXIT
e_gen	{0}	{0}	{0}	{1}	{0}	{0}	{1}	{1}	{1}	{0}	{0}	{0}	{0}	{0}	{0}
e_kill	{0}	{1}	{1}	{0}	{0}	{1}	{1}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}
anticipated_out	{0}	{0}	{1}	{1}	{1}	{1}	{1}	{1}	{0}	{1}	{1}	{1}	{1}	{1}	{0}
anticipated_in	{0}	{0}	{0}	{1}	{1}	{0}	{1}	{1}	{1}	{1}	{1}	{1}	{1}	{1}	{0}
available_in	{0}	{0}	{0}	{0}	{1}	{0}	{1}	{0}	{1}	{1}	{0}	{1}	{1}	{1}	{1}
available_out	{0}	{0}	{0}	{1}	{1}	{0}	{0}	{1}	{1}	{1}	{1}	{1}	{1}	{1}	{1}
earliest	{0}	{0}	{0}	{1}	{0}	{0}	{0}	{1}	{0}	{0}	{1}	{0}	{0}	{0}	{0}
postponable_in	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}
postponable_out	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{0}	{1}	{0}	{0}	{0}	{0}
latest	{0}	{0}	{0}	{1}	{0}	{0}	{0}	{1}	{0}	{0}	{1}	{0}	{0}	{0}	{0}
used_out	{0}	{0}	{0}	{1}	{1}	{0}	{0}	{1}	{0}	{1}	{1}	{1}	{1}	{1}	{0}
used_in	{0}	{0}	{0}	{0}	{1}	{0}	{1}	{0}	{1}	{1}	{0}	{1}	{1}	{1}	{0}
cond_1	{0}	{0}	{0}	{1}	{0}	{0}	{0}	{1}	{0}	{0}	{1}	{0}	{0}	{0}	{0}
cond_2	{0}	{0}	{0}	{1}	{0}	{0}	{1}	{1}	{1}	{0}	{0}	{0}	{0}	{0}	{0}

Table 8: Eliminação de Redundâncias Parciais —  $((+, a, b))$



## 9 Propagação de Constantes

*Gen ...*

*Kill ...*

*In ...*

*In ...*

	IN	OUT
$B_1$	$(\top, \top, \top, \top, \top)$	$(4, 5, \top, \top, \top)$
$B_2$	$(4, 5, \top, \top, \top)$	$(4, 5, \top, \top, \top)$
$B_3$	$(4, 5, \top, \top, \top)$	$(4, 5, 9, \top, 9)$
$B_4$	$(4, 5, 9, \top, 9)$	$(4, 5, 9, \top, 9)$
$B_5$	$(4, 5, \top, \top, \top)$	$(4, 5, \top, \top, \top)$
$B_6$	$(\perp, \perp, \perp, \perp, \perp)$	$(5, \perp, \perp, \perp, \perp)$
$B_7$	$(5, \perp, \perp, \perp, \perp)$	$(5, \perp, \perp, \perp, \perp)$
$B_8$	$(\perp, \perp, \perp, \perp, \perp)$	$(\perp, \perp, \perp, \perp, \perp)$
$B_9$	$(4, 5, 9, \top, 9)$	$(4, 5, 9, \top, 9)$
$B_{10}$	$(4, 5, \top, \top, \top)$	$(4, 5, \top, \top, 9)$
$B_{11}$	$(5, \perp, \perp, \perp, \perp)$	$(5, \perp, \perp, \perp, \perp)$
$B_{12}$	$(4, 5, 9, \top, 9)$	$(4, 5, 9, \top, 9)$
$B_{13}$	$(5, \perp, \perp, \perp, \perp)$	$(5, \perp, \perp, \perp, \perp)$

Table 9: Propagação de Constantes —  $(a, b, c, d, t_5)$

