Slicing

Mohammad Mousavi

Department of Informatics, King's College London

Software Testing and Measurement,
November 23, 2021

Outline

1. Sequential Slicing

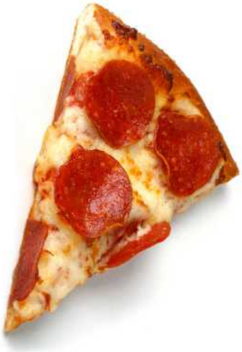
2. Structured Slicing

Slices

An executable subset of the program

- capturing possible (indirect) dependencies
- among all definitions and uses
- influencing the value of a set of variables.

Also called: cone of influence reduction



Annotated Flow Graphs

Defining nodes

$DEF(n, v)$ holds (for a var. v and a node n), when n defines v.
Examples:

- $input(v)$, or
- $v := exp$

$$DEF(n) \; = \; \{v \mid DEF(n, v)\}$$

Annotated Flow Graphs

Using nodes

$USE(n, v)$ holds (for a var. v and a node n), when n uses the values of v.
Examples:

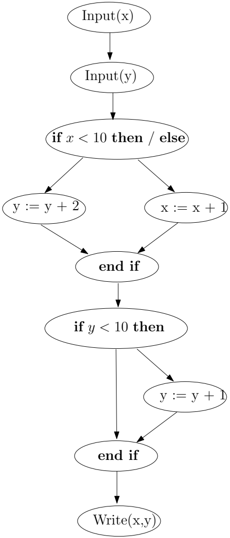
- $output(v)$,
- $x := exp(v)$,
- $if\ cond(v)\ then$, or
- $while\ cond(v)\ do$, ...

$$USE(n) \;=\; \{v \mid USE(n, v)\}$$

Also $REF(n, v)$ in the literature

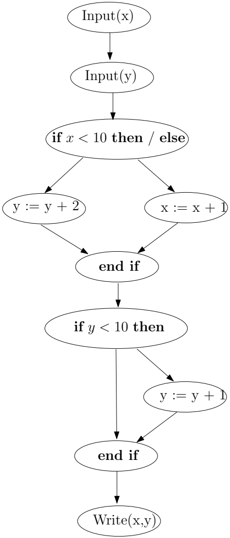
Definitions and Uses: An Example

```
Input(x)
Input(y)
if x < 10 then
    y := y + 2
else
    x:= x+1
end if
if y > 20 then
    y := y +1;
end if
Write(x,y)
end
```



Definitions and Uses: An Example

```
1:  Input(x) {DEF(1) = {x}}
2:  Input(y) {DEF(2) = {y}}
3:  if x < 10 then
4:      y := y + 2 {DEF(4) = USE(4) = {y}}
5:  else
6:      x:= x+1
7:  end if
8:  if y > 20 then
9:      y := y +1;
10: end if
11: Write(x,y) {USE(11) = {x,y}}
12: end
```



Slicing: An Example

```
1:  Input(x)
2:  Input(y)
3:  total := 0
4:  sum := 0
5:  if x ≤ 1 then
6:      sum := y
7:  else
8:      Input(z)
9:      total := x * y
10: end if
11: Write(total, sum)
```

Slice on $\{total\}$ at 11?

Slicing: An Example

```
1:  Input(x)
2:  Input(y)
3:  total := 0
4:  sum := 0
5:  if  x ≤ 1  then
6:      sum := y
7:  else
8:      Input(z)
9:      total := x * y
10: endif
11: Write(total, sum)
```

Slice on $\{total\}$ at 11?

Slicing: An Example

Slice on $\{total\}$ at 11:

```
1: Input(x)
2: Input(y)
3: total := 0
4: if x ≤ 1 then
5:
6: else
7:     total = x * y
8: end if
```

Slicing: An Example

```
1: Input(b)
2: c := 1
3: d := 3
4: a := d
5: d := b + d
6: b := b + 1
7: a := b + c
8: Write(a)
```

Slice on $\{d, c\}$ at 6?

Slicing: An Example

Slice on $\{d, c\}$ at 6:

```
1: Input(b)
2: c := 1
3: d := 3
4: d := b + d
```

$(6, \{d, c\})$ (in general (n, V)): the slicing criterion

Outline of the algorithm

Slice criterion (n, V)

- Statements in the slice: those define the relevant variables.
- At n, $v \in V$: relevant.
- A relevant $v \in DEF(m)$: v is no more relevant above m,
- but then all variables in $USE(m)$ become relevant above m.

Relevant Variables

Given a slicing criterion (n, V), $Relevant_0(m) =$

$$\begin{cases} V & \text{if } m = n+1 \\ \{v \mid \exists_{m \to m'}(v \in relevant(m') \setminus DEF(m) \vee \\ \quad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \text{otherwise} \end{cases}$$

Relevant Variables

Given a slicing criterion (n, V), $Relevant_0(m) =$

$$\begin{cases} 1)\ V & \text{if } m = n+1 \\ 2a)\{v \mid \exists_{m \to m'}(v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b)\qquad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

1) base case: all variables in V are initially relevant

2a) v remains relevant: has been relevant below and not defined at m

2b) v becomes relevant: defines relevant variables

Slicing: An Example

Slicing criterion: $(6,\ \{d, c\})$?

$$Relevant_0(m) = \begin{cases} 1)\ \ V & \text{if } m = n+1 \\ \\ 2a)\{v \mid \exists_{m \rightarrow m'}(v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b)\qquad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

m	$\text{Relevant}_0(\text{m})$
1 Input(b)	
2 c := 1	
3 d := 3	
4 a := d	
5 d := b + d	
6 b := b + 1	
	$\{d, c\}$

Slicing: An Example

Slicing criterion: $(6, \ \{d, c\})$?

$Relevant_0(m) =$

$$\begin{cases} 1) \ \ V & \text{if } m = n+1 \\ & \\ 2a) \{v \mid \exists_{m \to m'} (v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b) \qquad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

m	Relevant$_0$(m)
1 Input(b)	
2 c := 1	
3 d := 3	
4 a := d	
5 d := b + d	
6 b := b + 1	$\{d, c\}$
	$\{d, c\}$

Slicing: An Example

Slicing criterion: $(6,\ \{d, c\})$?

$$Relevant_0(m) = \begin{cases} 1)\ \ V & \text{if } m = n+1 \\ & \\ 2a)\{v \mid \exists_{m \to m'}(v \in relevant(m') \setminus DEF(m) \lor & \text{otherwise} \\ 2b)\qquad (DEF(m) \cap relevant(m') \neq \emptyset \land v \in USE(m)))\} & \end{cases}$$

m	Relevant$_0$(m)
1 Input(b)	
2 c := 1	
3 d := 3	
4 a := d	
5 d := b + d	$\{c, b, d\}$
6 b := b + 1	$\{d, c\}$
	$\{d, c\}$

Slicing: An Example

Slicing criterion: $(6,\ \{d, c\})$?

$Relevant_0(m) =$

$$\begin{cases} 1)\ \ V & \text{if } m = n+1 \\ & \\ 2a)\{v \mid \exists_{m \to m'}(v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b)\qquad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

m	Relevant$_0$(m)
1 Input(b)	
2 c := 1	
3 d := 3	
4 a := d	$\{c, b, d\}$
5 d := b + d	$\{c, b, d\}$
6 b := b + 1	$\{d, c\}$
	$\{d, c\}$

Slicing: An Example

Slicing criterion: $(6, \ \{d, c\})$?

$Relevant_0(m) =$

$$\begin{cases} 1)\ \ V & \text{if } m = n+1 \\ & \\ 2a)\{v \mid \exists_{m \to m'}(v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b) \qquad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

m	Relevant$_0$(m)
1 Input(b)	
2 c := 1	
3 d := 3	$\{c, b\}$
4 a := d	$\{c, b, d\}$
5 d := b + d	$\{c, b, d\}$
6 b := b + 1	$\{d, c\}$
	$\{d, c\}$

Slicing: An Example

Slicing criterion: $(6,\ \{d, c\})$?

$Relevant_0(m) =$

$$\begin{cases} 1)\ \ V & \text{if } m = n+1 \\ \\ 2a)\{v \mid \exists_{m \to m'} (v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b)\qquad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

m	Relevant$_0$(m)
1 Input(b)	
2 c := 1	$\{b\}$
3 d := 3	$\{c, b\}$
4 a := d	$\{c, b, d\}$
5 d := b + d	$\{c, b, d\}$
6 b := b + 1	$\{d, c\}$
	$\{d, c\}$

Slicing: An Example

Slicing criterion: $(6, \; \{d, c\})$?

$Relevant_0(m) =$

$$\begin{cases} 1) \; V & \text{if } m = n+1 \\ \\ 2a) \{v \mid \exists_{m \to m'} (v \in relevant(m') \setminus DEF(m) \vee & \text{otherwise} \\ 2b) \quad (DEF(m) \cap relevant(m') \neq \emptyset \wedge v \in USE(m)))\} & \end{cases}$$

m	Relevant$_0$(m)
1 Input(b)	$\emptyset$
2 c := 1	$\{b\}$
3 d := 3	$\{c, b\}$
4 a := d	$\{c, b, d\}$
5 d := b + d	$\{c, b, d\}$
6 b := b + 1	$\{d, c\}$
	$\{d, c\}$

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \to \ldots \to n$ and
   there exists an m' such that $m \to m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \to \ldots \to n$ and
   there exists an m' such that $m \to m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

m	$\mathrm{Relevant_0(m)}$	**DEF(m)**	$\in \mathrm{Slice_0}(6, \{d, c\})$
1 Input(b)	$\emptyset$		
2 c := 1	$\{b\}$		
3 d := 3	$\{c, b\}$		
4 a := d	$\{c, b, d\}$		
5 d := b + d	$\{c, b, d\}$		
6 b := b + 1	$\{d, c\}$		
	$\{d, c\}$		

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \to \ldots \to n$ and
   there exists an m' such that $m \to m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

m	$\mathrm{Relevant}_0(\mathrm{m})$	**DEF(m)**	$\in \mathrm{Slice}_0(6, \{\mathrm{d}, \mathrm{c}\})$
1 Input(b)	$\emptyset$		
2 c := 1	$\{b\}$		
3 d := 3	$\{c, b\}$		
4 a := d	$\{c, b, d\}$		
5 d := b + d	$\{c, b, d\}$		
6 b := b + 1	$\{d, c\}$	$\{b\}$	$\times$
	$\{d, c\}$		

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \rightarrow \ldots \rightarrow n$ and
   there exists an m' such that $m \rightarrow m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

m	$\mathrm{Relevant_0(m)}$	**DEF(m)**	$\in \mathrm{Slice_0}(6, \{d, c\})$
1 Input(b)	$\emptyset$		
2 c := 1	$\{b\}$		
3 d := 3	$\{c, b\}$		
4 a := d	$\{c, b, d\}$		
5 d := b + d	$\{c, b, d\}$	$\{d\}$	$\surd$
6 b := b + 1	$\{d, c\}$	$\{b\}$	$\times$
	$\{d, c\}$		

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \to \ldots \to n$ and
   there exists an m' such that $m \to m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

m	$\mathrm{Relevant_0(m)}$	**DEF(m)**	$\in \mathrm{Slice_0}(6, \{d, c\})$
1 Input(b)	$\emptyset$		
2 c := 1	$\{b\}$		
3 d := 3	$\{c, b\}$		
4 a := d	$\{c, b, d\}$	$\{a\}$	$\times$
5 d := b + d	$\{c, b, d\}$	$\{d\}$	$\surd$
6 b := b + 1	$\{d, c\}$	$\{b\}$	$\times$
	$\{d, c\}$		

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \rightarrow \ldots \rightarrow n$ and
   there exists an m' such that $m \rightarrow m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

m	$\mathrm{Relevant}_0(\mathrm{m})$	**DEF(m)**	$\in \mathrm{Slice}_0(6, \{d, c\})$
1 Input(b)	$\emptyset$		
2 c := 1	$\{b\}$		
3 d := 3	$\{c, b\}$	$\{d\}$	$\surd$
4 a := d	$\{c, b, d\}$	$\{a\}$	$\times$
5 d := b + d	$\{c, b, d\}$	$\{d\}$	$\surd$
6 b := b + 1	$\{d, c\}$	$\{b\}$	$\times$
	$\{d, c\}$		

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \to \ldots \to n$ and
   there exists an m' such that $m \to m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(6, \{d, c\})$
1 Input(b)	$\emptyset$		
2 c := 1	$\{b\}$	$\{c\}$	$\surd$
3 d := 3	$\{c, b\}$	$\{d\}$	$\surd$
4 a := d	$\{c, b, d\}$	$\{a\}$	$\times$
5 d := b + d	$\{c, b, d\}$	$\{d\}$	$\surd$
6 b := b + 1	$\{d, c\}$	$\{b\}$	$\times$
	$\{d, c\}$		

Slicing Sequential Programs

$m \in Slice_0(n, V)$ when

1. $n = m$ and $DEF(m) \cap V \neq \emptyset$, or
2. $m \to \ldots \to n$ and
   there exists an m' such that $m \to m'$ and
   $DEF(m) \cap Relevant_0(m') \neq \emptyset$

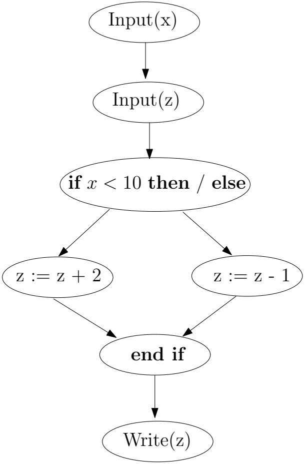
m	$\mathrm{Relevant}_0(\mathrm{m})$	**DEF(m)**	$\in \mathrm{Slice}_0(6, \{d, c\})$
1 Input(b)	$\emptyset$	$\{b\}$	$\surd$
2 c := 1	$\{b\}$	$\{c\}$	$\surd$
3 d := 3	$\{c, b\}$	$\{d\}$	$\surd$
4 a := d	$\{c, b, d\}$	$\{a\}$	$\times$
5 d := b + d	$\{c, b, d\}$	$\{d\}$	$\surd$
6 b := b + 1	$\{d, c\}$	$\{b\}$	$\times$
	$\{d, c\}$		

Outline

Slicing Programs with Conditions

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z := z + 2;
5: else
6:     z := z - 1;
7: end if
8: Write(z)
```

Slice wrt. the criterion $(3, \{x\})$?



Slicing Programs with Conditions

Slice wrt. the criterion $(3, \{x\})$?

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(3, \{x\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
	$\{x\}$		

Slicing Programs with Conditions

Slice wrt. the criterion $(3, \{x\})$?

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(3, \{x\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**	$\{x\}$ $\{x\}$	$\emptyset$	$\times$

Slicing Programs with Conditions

Slice wrt. the criterion $(3, \{x\})$?

m	$\mathrm{Relevant}_0(\mathrm{m})$	**DEF(m)**	$\in \mathrm{Slice}_0(3, \{\mathrm{x}\})$
1 Input(x)			
2 Input(z)	$\{x\}$	$\{z\}$	$\times$
3,5 **if** $x < 10$ **then** / **else**	$\{x\}$	$\emptyset$	$\times$
	$\{x\}$		

Slicing Programs with Conditions

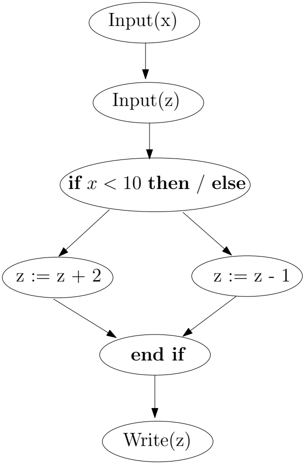
Slice wrt. the criterion $(3, \{x\})$?

m	$\mathrm{Relevant}_0(\mathrm{m})$	**DEF(m)**	$\in \mathrm{Slice}_0(3, \{\mathrm{x}\})$
1 Input(x)	$\emptyset$	$\{x\}$	$\surd$
2 Input(z)	$\{x\}$	$\{z\}$	$\times$
3,5 **if** $x < 10$ **then** / **else**	$\{x\}$	$\emptyset$	$\times$
	$\{x\}$		

Slicing Programs with Conditions

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z := z + 2;
5: else
6:     z := z - 1;
7: end if
8: Write(z)
```

Slice wrt. the criterion $(8, \{z\})$?



Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(8, \{\text{z}\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1			
7 **end if**			
8 Write(z)			
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1			
7 **end if**			
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\mathrm{Relevant}_0(\mathrm{m})$	**DEF(m)**	$\in \mathrm{Slice}_0(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1			
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(8, \{\text{z}\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**	$\{z\}$	$\emptyset$	$\times$
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	**DEF(m)**	$\in \text{Slice}_0(8, \{\text{z}\})$
1 Input(x)			
2 Input(z)	$\emptyset$	$\{z\}$	$\surd$
3,5 **if** $x < 10$ **then** / **else**	$\{z\}$	$\emptyset$	$\times$
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_0(\text{m})$	DEF(m)	$\in \text{Slice}_0(8, \{z\})$
1 Input(x)	$\emptyset$	$\{x\}$	$\times$
2 Input(z)	$\emptyset$	$\{z\}$	$\surd$
3,5 **if** $x < 10$ **then** / **else**	$\{z\}$	$\emptyset$	$\times$
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Structured Programs: Informal Idea

1. Start with sequential slicing algorithm: $Slice_0(n, v)$
2. Find all conditionals $Cond_{k+1}(n, V)$ influencing $m \in Slice_k(n, v)$
3. Add the following node to $Slice_k(n, V)$, the result: $Slice_{k+1}(n, V)$
    1. the conditional in $c \in Cond_k n, V$ and
    2. those statement influencing the conditions of c
4. repeat 2 until a fixed-point

(Inverse) Denominators

$m \in IDen(n)$ (m inversely denominates n)
when m appears in all paths $n \to \ldots \to n_t$.

$m = NIDen(n)$(the nearest inverse denominator of n) when
$m \in IDen(n)$ and
for all $m' \in IDen(n)$ either $m = m'$ or there is a simple path
$m \to \ldots \to m'$.

$m \in Infl(n)$(m is influenced by n) when
m appears in a path from n to $NIDen(n)$
($m \neq n$, $m \neq NIDen(n)$, $NIDen(n)$ may not appear in the path).

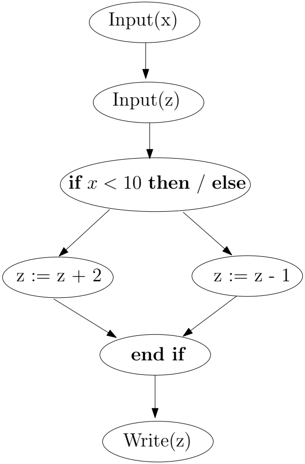
Slicing Programs with Conditions

1: Input(x)
2: Input(z)
3: **if** x $<$ 10 **then**
4:     z $=$ z $+$ 2;
5: **else**
6:     z $=$ z - 1;
7: **end if**
8: Write(z)

$NIDen(1)$? $Infl(1)$?

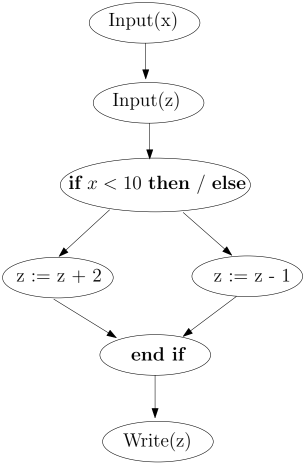
$NIDen(2)$? $Infl(2)$?

$NIDen(3)$? $Infl(3)$?



Slicing Programs with Conditions

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z = z + 2;
5: else
6:     z = z - 1;
7: end if
8: Write(z)
```



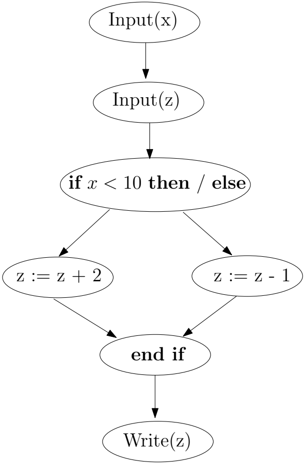
NIDen(1)? 2. *Infl*(1)? $\emptyset$.

NIDen(2)? *Infl*(2)?

NIDen(3)? *Infl*(3)?

Slicing Programs with Conditions

1: Input(x)
2: Input(z)
3: **if** x $<$ 10 **then**
4:     z $=$ z $+$ 2;
5: **else**
6:     z $=$ z - 1;
7: **end if**
8: Write(z)



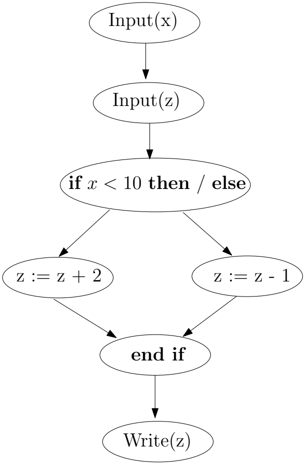
NIDen(1)? 2. *Infl*(1)? $\emptyset$.

NIDen(2)? 3. *Infl*(2)? $\emptyset$.
Observation, for sequential nodes *Infl*(n) $= \emptyset$.

NIDen(3)? *Infl*(3)?

Slicing Programs with Conditions

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z = z + 2;
5: else
6:     z = z - 1;
7: end if
8: Write(z)
```



NIDen(1)? 2. *Infl*(1)? $\emptyset$.

NIDen(2)? 3. *Infl*(2)? $\emptyset$.
Observation, for sequential nodes $\mathit{Infl}(n) = \emptyset$.

NIDen(3)? 7. *Infl*(3)? $\{4, 6\}$.

Slicing Structured Programs

Given a slicing criterion (n, V):
$m \in Cond_{k+1}(n, V)$ (conditions influencing $Slice_k(n, V)$) when there exists $m' \in Slice_k(n, V)$ and $m' \in Infl(m)$.

Slicing Structured Programs

Given a slicing criterion (n, V):
$m \in Cond_{k+1}(n, V)$ (conditions influencing $Slice_k(n, V)$) when
there exists $m' \in Slice_k(n, V)$ and $m' \in Infl(m)$.

$v \in Relevant_{k+1}(m)$ when
$v \in Relevant_k(m)$ or
there exists an $m' \in Cond_{k+1}(n, V)$ and
$v \in Relevant_0(m)$ w.r.t. the slicing criterion $(m', USE(m'))$.

Slicing Structured Programs

Given a slicing criterion (n, V):
$m \in Cond_{k+1}(n, V)$ (conditions influencing $Slice_k(n, V)$) when
there exists $m' \in Slice_k(n, V)$ and $m' \in Infl(m)$.

$v \in Relevant_{k+1}(m)$ when
$v \in Relevant_k(m)$ or
there exists an $m' \in Cond_{k+1}(n, V)$ and
$v \in Relevant_0(m)$ w.r.t. the slicing criterion $(m', USE(m'))$.

$m \in Slice_{k+1}(n, V)$ when
$m \in Cond_{k+1}(n, V)$ or
there exists an m' such that $m \to m'$ and
$DEF(m) \cap Relevant_{k+1}(m') \neq \emptyset$.

Slicing Programs with Conditions

Slice wrt. $(8, \{z\})$

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z = z + 2;
5: else
6:     z = z - 1;
7: end if
8: Write(z)
```

Slicing Programs with Conditions

Slice wrt. $(8, \{z\})$

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z = z + 2;
5: else
6:     z = z - 1;
7: end if
8: Write(z)
```

$Slice_0(8, \{z\}) = \{2, 4, 6\}$.

Slicing Programs with Conditions

Slice wrt. $(8, \{z\})$

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z = z + 2;
5: else
6:     z = z - 1;
7: end if
8: Write(z)
```

$Slice_0(8, \{z\}) = \{2, 4, 6\}$.

$m \in Cond_{k+1}(n, V)$ (conditions influencing $Slice_k(n, V)$) when there exists $m' \in Slice_k(n, V)$ and $m' \in Infl(m)$.

Slicing Programs with Conditions

Slice wrt. $(8, \{z\})$

```
1: Input(x)
2: Input(z)
3: if x < 10 then
4:     z = z + 2;
5: else
6:     z = z - 1;
7: end if
8: Write(z)
```

$Slice_0(8, \{z\}) = \{2, 4, 6\}$.

$Cond_1(8, \{z\}) = \{3\}$

$Slice_1(8, \{z\})$?

Slicing Programs with Conditions

m	Relevant$_1$(m)	**DEF(m)**	$\in$ Slice$_1$(8, {z})
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1			
7 **end if**			
8 Write(z)			
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_1(m)$	**DEF(m)**	$\in \text{Slice}_1(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1			
7 **end if**			
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_1(\text{m})$	**DEF(m)**	$\in \text{Slice}_1(8, \{\text{z}\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1			
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	Relevant$_1$(m)	**DEF(m)**	$\in$ Slice$_1$(8, {z})
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2			
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_1(\text{m})$	**DEF(m)**	$\in \text{Slice}_1(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**			
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_1(\text{m})$	**DEF(m)**	$\in \text{Slice}_1(8, \{z\})$
1 Input(x)			
2 Input(z)			
3,5 **if** $x < 10$ **then** / **else**	$\{z, x\}$	$\emptyset$	$\surd$
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Slicing Programs with Conditions

m	$\text{Relevant}_1(\text{m})$	**DEF(m)**	$\in \text{Slice}_1(8, \{\text{z}\})$
1 Input(x)			
2 Input(z)	$\{x\}$	$\{z\}$	$\surd$
3,5 **if** $x < 10$ **then** / **else**	$\{z, x\}$	$\emptyset$	$\surd$
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

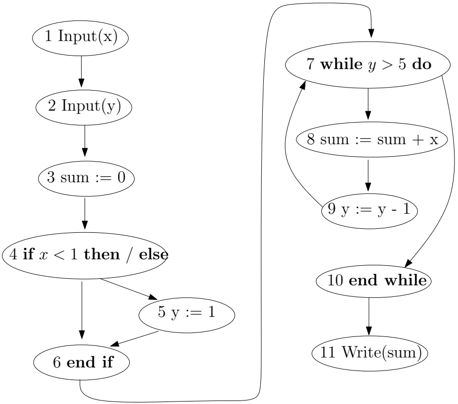
Slicing Programs with Conditions

m	Relevant$_1$(m)	**DEF(m)**	$\in$ Slice$_1$(8, {z})
1 Input(x)	$\emptyset$	$\{x\}$	$\surd$
2 Input(z)	$\{x\}$	$\{z\}$	$\surd$
3,5 **if** $x < 10$ **then** / **else**	$\{z, x\}$	$\emptyset$	$\surd$
4 z := z + 2	$\{z\}$	$\{z\}$	$\surd$
6 z := z - 1	$\{z\}$	$\{z\}$	$\surd$
7 **end if**	$\{z\}$	$\emptyset$	$\times$
8 Write(z)	$\{z\}$	$\emptyset$	$\times$
	$\{z\}$		

Another Example

Slice wrt. $(11, \{sum\})$?

```
1:  Input(x)
2:  Input(y)
3:  sum := 0
4:  if x < 1 then
5:      y : = 1
6:  end if
7:  while y ≥ 1 do
8:      sum := sum + x
9:      y := y - 1
10: end while
11: Write(sum)
```



m	DEF(m)	$\text{Relevant}_0(m)$	Slice_0	Cond_1	Rel_1	Slice_1
1	$\{x\}$	$\emptyset$	$\surd,$	$\times$	$\emptyset$	$\surd$
2	$\{y\}$	$\{x\}$	$\times$	$\times$	$\{x\}$	$\surd$
3	$\{sum\}$	$\{x\}$	$\surd$	$\times$	$\{x, y\}$	$\surd$
4	$\emptyset$	$\{sum, x\}$	$\times$	$\times$	$\{sum, x, y\}$	$\times$
5	$\{y\}$	$\{sum, x\}$	$\times$	$\times$	$\{sum, x\}$	$\surd$
6	$\emptyset$	$\{sum, x\}$	$\times$	$\times$	$\{sum, x, y\}$	$\times$
7	$\emptyset$	$\{sum, x\}$	$\times$	$\surd$	$\{sum, x, y\}$	$\surd$
8	$\{sum\}$	$\{sum, x\}$	$\surd$	$\times$	$\{sum, x, y\}$	$\surd$
9	$\{y\}$	$\{sum, x\}$	$\times$	$\times$	$\{sum, x, y\}$	$\surd$
10	$\emptyset$	$\{sum\}$	$\times$	$\times$	$\{sum\}$	$\times$
11	$\emptyset$	$\{sum\}$	$\times$	$\times$	$\{sum\}$	$\times$
		$\{sum\}$			$\{sum\}$	

m	DEF(m)	$Cond_2$	Rel_2	$Slice_2$	$Slice^{(*)}$
1	$\{x\}$	$\times$	$\emptyset$	$\surd$	$\surd$
2	$\{y\}$	$\times$	$\{x\}$	$\surd$	$\surd$
3	$\{sum\}$	$\times$	$\{x, y\}$	$\surd$	$\surd$
4	$\emptyset$	$\surd$	$\{sum, x, y\}$	$\surd$	$\surd$
5	$\{y\}$	$\times$	$\{sum, x\}$	$\surd$	$\surd$
6	$\emptyset$	$\times$	$\{sum, x, y\}$	$\times$	$\surd$
7	$\emptyset$	$\surd$	$\{sum, x, y\}$	$\surd$	$\surd$
8	$\{sum\}$	$\times$	$\{sum, x, y\}$	$\surd$	$\surd$
9	$\{y\}$	$\times$	$\{sum, x, y\}$	$\surd$	$\surd$
10	$\emptyset$	$\times$	$\{sum\}$	$\times$	$\surd$
11	$\emptyset$	$\times$	$\{sum\}$	$\times$	$\times$

(*) Syntactic check after generating the slice:

if then (/**else**) $\in$ *Slice* $\Rightarrow$ (the corresponding) **end if** $\in$ *Slice*

while ... **do** $\in$ *Slice* $\Rightarrow$ (the corresponding) **end while** $\in$ *Slice*

...

The Ideal Slicing Algorithm?

Slice wrt. $(2, \{x\})$?

1: Input(x)

2: x := x

The Ideal Slicing Algorithm?

Slice wrt. $(2, \{x\})$?

```
1: Input(x)
2: x := x
```

Slice wrt. $(5, \{x\})$?

```
1: if true then
2:     x := 1
3: else
4:     x := 2
5: end if
```

The Ideal Slicing Algorithm?

Slice wrt. $(2, \{x\})$?

```
1: Input(x)
2: x := x
```

Slice wrt. $(5, \{x\})$?

```
1: if true then
2:     x := 1
3: else
4:     x := 2
5: end if
```

No algorithm for the smallest slice exists!
Reason: Undecidability of halting/termination.

Slicing: Applications

1. Test adequacy: for each output variable, all du-paths in its slice must be covered
2. Robustness testing: Add pseudo-variables that check dangerous situations,
   generate the slice and test
3. Regression testing: testing if a change influences a particular component
   (i.e., if the slice of the component interface contains the change)
4. Debugging:
   code review
   comparing a correct running program with a new faulty version