Discovering Properties about Arrays in Simple Programs

Nicolas Halbwachs and Mathias Péron

Grenoble – France

Considering Arrays in Static Analysis

A lot of work done

- array bound checking
  but . . .
- array dependence/dataflow analysis
  for automatic parallelization
  for optimizations

```
i ← A[j] ;
A[i] ← x
```

A lot of work to be done

- array contents!
  decision procedure
  - synthesis of properties
    - Which properties?
    - How many dimensions?
    - Dynamic memory? pointers?

```
for i = 2 to n do
    s ← 0 ;
    for j = 1 to i-1 do
        s ← s + A[j]
    A[i] ← s
```

Considering Arrays in Static Analysis

A lot of work done

- array bound checking
  but ...
- array dependence/dataflow analysis
  for automatic parallelization
  for optimizations

```
i ← A[j] ;
A[i] ← x
```

A lot of work to be done

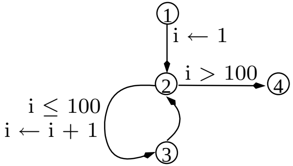
- array contents!
  decision procedure
  - synthesis of properties
    - Which properties?
    - How many dimensions?
    - Dynamic memory? pointers?

```
for i = 1 to n do
  S[i] ← 0
for i = 1 to n do
  A[i] ← A[i] + S[i]
  for j = i+1 to n do
    S[j] ← S[j] + A[i]
```

Static Analysis thanks to Abstract Interpretation

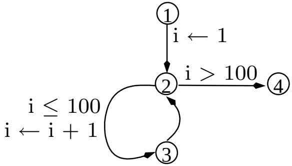
```
i ← 1 ;
while i ≤ 100
do
  i ← i + 1 ;
```

Static Analysis thanks to Abstract Interpretation



Static Analysis thanks to Abstract Interpretation

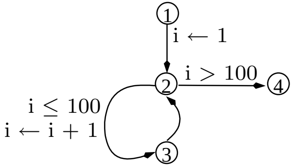
$R1 = i \in [-\infty, +\infty]$
$R2 = (R1\ [i \leftarrow 1]) \sqcup R3$
$R3 = (R2 \sqcap (i \leq 100))\ [i \leftarrow i + 1]$
$R4 = R3 \sqcap (i > 100)$



	1st	*2nd*		
R1	$i \in [-\infty, +\infty]$			
R2	$\perp$			
R3	$\perp$			
R4	$\perp$			

Static Analysis thanks to Abstract Interpretation

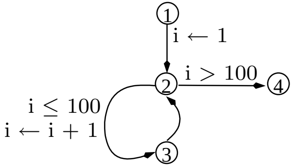
$R1 = i \in [-\infty, +\infty]$
$R2 = (R1\ [i \leftarrow 1]) \sqcup R3$
$R3 = (R2 \sqcap (i \leq 100))\ [i \leftarrow i + 1]$
$R4 = R3 \sqcap (i > 100)$



	1st	*2nd*	*3th*	
R1	$i \in [-\infty, +\infty]$	$\text{i} \in [-\infty, +\infty]$		
R2	$\perp$	$i \in [1, 1]$		
R3	$\perp$	$i \in [2, 2]$		
R4	$\perp$	$\perp$		

Static Analysis thanks to Abstract Interpretation

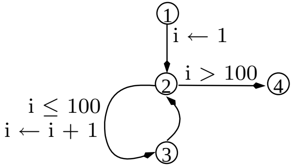
R1 $= i \in [-\infty, +\infty]$
R2 $= (\text{R1 } [i \leftarrow 1]) \sqcup \text{R3}$
R3 $= (\text{R2 } \sqcap (i \leq 100)) \ [i \leftarrow i+1]$
R4 $= \text{R3 } \sqcap (i > 100)$



	1st	*2nd*	*3th*	
R1	$i \in [-\infty, +\infty]$	$\text{i} \in [-\infty, +\infty]$	$\text{i} \in [-\infty, +\infty]$	
R2	$\perp$	$i \in [1, 1]$	$i \in [1, 2]$	
R3	$\perp$	$i \in [2, 2]$	$i \in [2, 3]$	
R4	$\perp$	$\perp$	$\perp$	

Static Analysis thanks to Abstract Interpretation

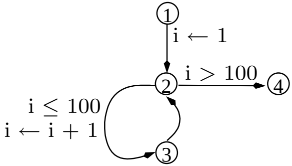
$R1 = i \in [-\infty, +\infty]$
$R2 = (R1\ [i \leftarrow 1]) \sqcup R3$
$R3 = (R2 \sqcap (i \leq 100))\ [i \leftarrow i + 1]$
$R4 = R3 \sqcap (i > 100)$



	1st	*2nd*	*4th*	
R1	$i \in [-\infty, +\infty]$	$\mathrm{i} \in [-\infty, +\infty]$	$\mathrm{i} \in [-\infty, +\infty]$	
R2	$\perp$	$i \in [1, 1]$	$i \in [1, 3]$	
R3	$\perp$	$i \in [2, 2]$	$i \in [2, 4]$	
R4	$\perp$	$\perp$	$\perp$	

Static Analysis thanks to Abstract Interpretation

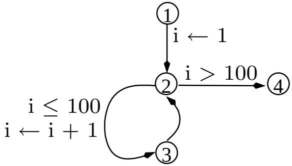
R1 $= i \in [-\infty, +\infty]$
R2 $=$ (R1 $[i \leftarrow 1]$) $\sqcup$ R3
R3 $=$ (R2 $\sqcap (i \leq 100)$) $[i \leftarrow i+1]$
R4 $=$ R3 $\sqcap (i > 100)$



	1st	*2nd*	*102th is FP*	
R1	$i \in [-\infty, +\infty]$	$\mathrm{i} \in [-\infty, +\infty]$	$\mathrm{i} \in [-\infty, +\infty]$	
R2	$\perp$	$i \in [1, 1]$	$i \in [1, 101]$	
R3	$\perp$	$i \in [2, 2]$	$i \in [2, 101]$	
R4	$\perp$	$\perp$	$i \in [101, 101]$	

Static Analysis thanks to Abstract Interpretation

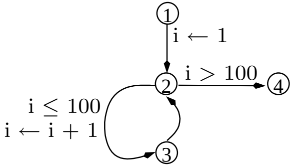
$R1 = i \in [-\infty, +\infty]$
$R2 = R2 \nabla ((R1\ [i \leftarrow 1]) \sqcup R3)$
$R3 = (R2 \sqcap (i \leq 100))\ [i \leftarrow i + 1]$
$R4 = R3 \sqcap (i > 100)$



	1st	*2nd*	*3th*	
R1	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$	
R2	$\perp$	$i \in [1, 1]$	$i \in [1, 2]$	
R3	$\perp$	$i \in [2, 2]$		
R4	$\perp$	$\perp$		

Static Analysis thanks to Abstract Interpretation

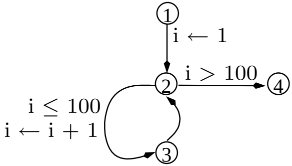
$R1 = i \in [-\infty, +\infty]$
$R2 = R2 \nabla ((R1\ [i \leftarrow 1]) \sqcup R3)$
$R3 = (R2 \sqcap (i \leq 100))\ [i \leftarrow i + 1]$
$R4 = R3 \sqcap (i > 100)$



	1st	*2nd*	*3th is FP*	
R1	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$	
R2	$\perp$	$i \in [1, 1]$	$i \in [1, +\infty]$	
R3	$\perp$	$i \in [2, 2]$	$i \in [2, 101]$	
R4	$\perp$	$\perp$	$i \in [101, +\infty]$	

Static Analysis thanks to Abstract Interpretation

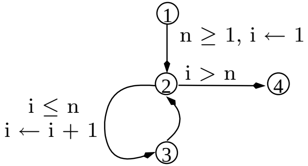
$$\begin{aligned} R1 &= i \in [-\infty, +\infty] \\ R2 &= R2 \nabla ((R1\ [i \leftarrow 1]) \sqcup R3) \\ R3 &= (R2 \sqcap (i \leq 100))\ [i \leftarrow i+1] \\ R4 &= R3 \sqcap (i > 100) \end{aligned}$$



	1st	*2nd*	*3th is FP*	*desc. is FP*
R1	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$	$i \in [-\infty, +\infty]$
R2	$\perp$	$i \in [1, 1]$	$i \in [1, +\infty]$	$i \in [1, 101]$
R3	$\perp$	$i \in [2, 2]$	$i \in [2, 101]$	$i \in [2, 101]$
R4	$\perp$	$\perp$	$i \in [101, +\infty]$	$i \in [101, 101]$

Static Analysis thanks to Abstract Interpretation

$R1 = \top$
$R2 = R2 \nabla ((R1\ [i \leftarrow 1]) \sqcup R3)$
$R3 = (R2 \sqcap (i \leq n))\ [i \leftarrow i+1]$
$R4 = R3 \sqcap (i > n)$

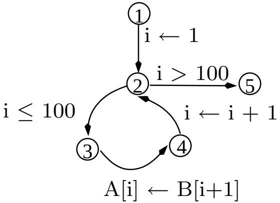


	1st	*2nd*	*3th is FP*	*desc. is FP*
R1	$\top$	$\top$	$\top$	$\top$
R2	$\perp$	$n \geq i = 1$	$n \geq 1, i \geq 1$	$n \geq i-1, i \geq 1$
R3	$\perp$	$n \geq i-1, n \geq 1, i = 2$	$n \geq i-1 \geq 1$	$n \geq i-1 \geq 1$
R4	$\perp$	$\perp$	$i > n \geq 1$	$i = n+1 \geq 2$

Array Summarization

```
i ← 1 ;
while i ≤ 100 do
    A[i] ← B[i+1] ;
    i ← i + 1 ;
```

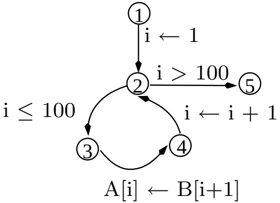
Array Summarization



Array Summarization

- Abstract each array A by a single variable a
- Interpretation
  $\psi(a) \;\Leftrightarrow\; \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoondown exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:

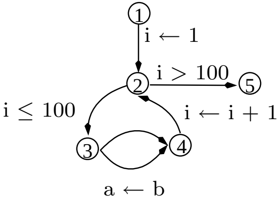
  $\text{R4} = \text{R3} \sqcup (\text{R3}\ [a \leftarrow b])$



Array Summarization

- Abstract each array A by a single variable a
- Interpretation $\psi(a) \Leftrightarrow \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoondown exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:

  $R4 = R3 \sqcup (R3\ [a \leftarrow b])$



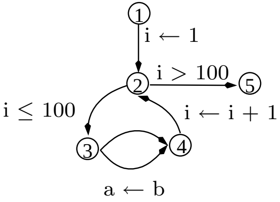
Issues

- weak assignment can only lose information
- information about the initial content of arrays must be obtained by other means

Array Summarization

- Abstract each array A by a single variable a
- Interpretation
  $\psi(a) \Leftrightarrow \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoonup exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:

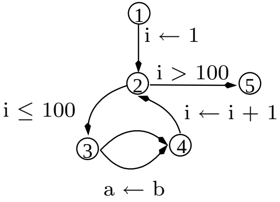
$$R4 = R3 \sqcup (R3\ [a \leftarrow b])$$



	1st	
R1	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$	
R2	$\perp$	
R3	$\perp$	
R4	$\perp$	
R5	$\perp$	

Array Summarization

- Abstract each array A by a single variable a
- Interpretation $\psi(a) \;\Leftrightarrow\; \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoondown exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:

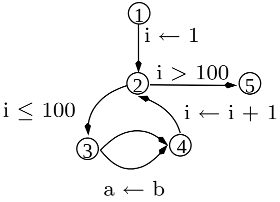


$$\text{R4} = \text{R3} \sqcup (\text{R3}\ [a \leftarrow b])$$

	1st	*2nd*
R1	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$
R2	$\perp$	$a = i - 1 = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10,\ 4 \leq b - i \leq 9$
R3	$\perp$	*idem*
R4	$\perp$	
R5	$\perp$	

Array Summarization

- Abstract each array A by a single variable a
- Interpretation
  $\psi(a) \iff \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoondown exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:

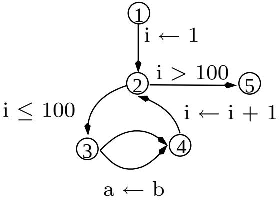


$$\text{R4} = \text{R3} \sqcup (\text{R3}\ [a \leftarrow b])$$

	1st	*2nd*
R1	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$
R2	$\perp$	$a = i - 1 = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10,\ 4 \leq b - i \leq 9$
R3	$\perp$	*idem*
R4	$\perp$	$i = 1, 5 \leq b \leq 10$ $4 \leq b - i \leq 9$
R5	$\perp$	

Array Summarization

- Abstract each array A by a single variable a
- Interpretation
  $\psi(a) \Leftrightarrow \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoondown exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:

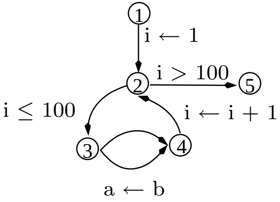


R4 $=$ R3 $\sqcup$ (R3 $[a \leftarrow b]$)

	1st	*2nd*
R1	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$
R2	$\perp$	$a = i - 1 = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10,\ 4 \leq b - i \leq 9$
R3	$\perp$	*idem*
R4	$\perp$	$i = 1, 5 \leq a = b \leq 10$ $4 \leq b - i \leq 9,\ 4 \leq a - i \leq 9$
R5	$\perp$	

Array Summarization

- Abstract each array A by a single variable a
- Interpretation
  $\psi(a) \Leftrightarrow \forall \ell = 1..n, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is weak assignment to variable a ($a \leftharpoondown exp$). *i.e.* indeterministic choice between $a \leftarrow exp$ and *leave unchanged*:



R4 $=$ R3 $\sqcup$ (R3 $[a \leftarrow b]$)

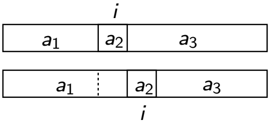
	1st	*2nd*
R1	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$	$a = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10$
R2	$\perp$	$a = i - 1 = 0, 5 \leq b \leq 10$ $5 \leq b - a \leq 10,\ 4 \leq b - i \leq 9$
R3	$\perp$	*idem*
R4	$\perp$	$i = 1, 0 \leq a \leq 10, 5 \leq b \leq 10$ $0 \leq b - a \leq 10,\ 4 \leq b - i \leq 9, -1 \leq a - i \leq 9$
R5	$\perp$	

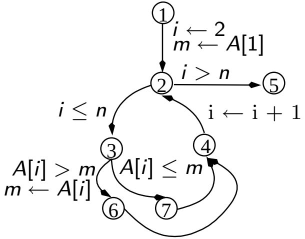
Symbolic Partitioning & Summarization

[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i{-}1], A_2 = A[i], A_3 = A[i{+}1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \;\Leftrightarrow\; \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i + 1$ is
  $a_1 \leftharpoonup a_2; a_2 \leftarrow a_3$

i $\leftarrow i + 1$



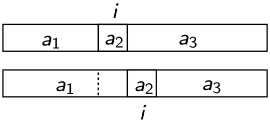


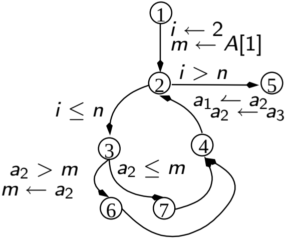
Symbolic Partitioning & Summarization

[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i{-}1], A_2 = A[i], A_3 = A[i{+}1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \;\Leftrightarrow\; \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$

i $\leftarrow i+1$

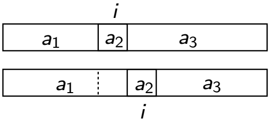




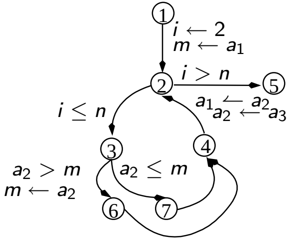
Symbolic Partitioning & Summarization
[Gopan, Reps, Sagiv - POPL'05]

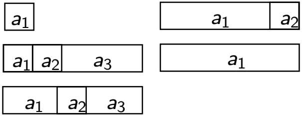
- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \;\Leftrightarrow\; \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$

i $\leftarrow i+1$



- An abstract value is a set of configurations. A lattice element is associated to each of them

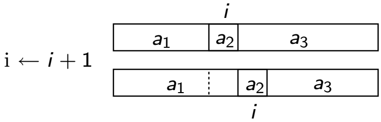




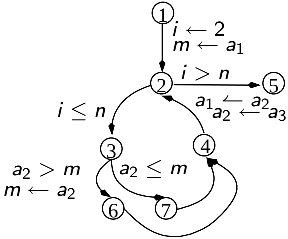
Symbolic Partitioning & Summarization

[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i + 1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$



- An abstract value is a set of configurations. A lattice element is associated to each of them



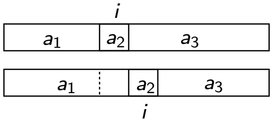
	2nd	3th
R2	$m = a_1$	
R6		
R7		
R4		
R5		

Symbolic Partitioning & Summarization

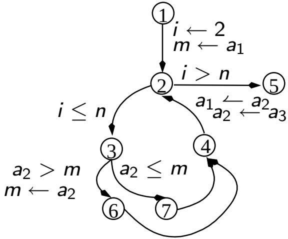
[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$

i $\leftarrow i+1$



- An abstract value is a set of configurations. A lattice element is associated to each of them



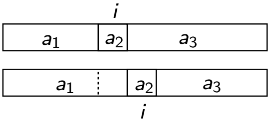
	2nd	3th
R2	$m = a_1$	
R6	$a_2 > a_1 = m$	
R7		
R4		
R5		

Symbolic Partitioning & Summarization

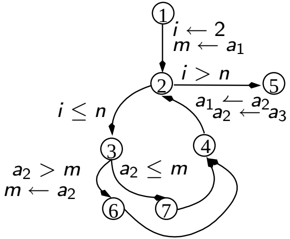
[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$

i $\leftarrow i+1$



- An abstract value is a set of configurations. A lattice element is associated to each of them



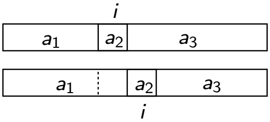
	2nd	3th
R2	$m = a_1$	
R6	$m = a_2 > a_1$	
R7		
R4		
R5		

Symbolic Partitioning & Summarization

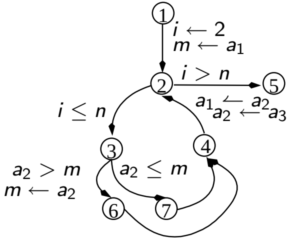
[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoonup a_2; a_2 \leftarrow a_3$

i $\leftarrow i+1$



- An abstract value is a set of configurations. A lattice element is associated to each of them

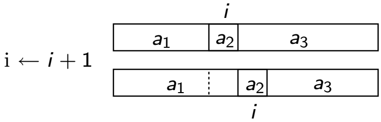


	2nd	3th
R2	$m = a_1$	
R6	$m = a_2 > a_1$	
R7	$a_1 = m \geq a_2$	
R4		
R5		

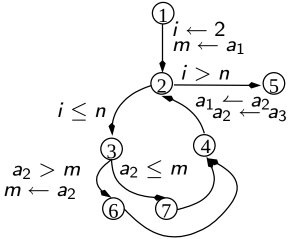
Symbolic Partitioning & Summarization

[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoonup a_2; a_2 \leftarrow a_3$



- An abstract value is a set of configurations. A lattice element is associated to each of them

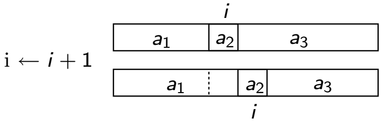


	2nd	3th
R2	$m = a_1$	
R6	$m = a_2 > a_1$	
R7	$a_1 = m \geq a_2$	
R4	$m \geq a_1, m \geq a_2$	
R5		

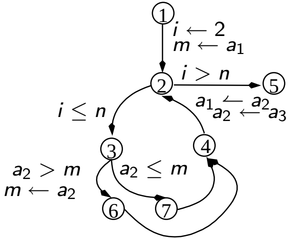
Symbolic Partitioning & Summarization

[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \;\Leftrightarrow\; \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i + 1$ is
  $a_1 \leftharpoonup a_2; a_2 \leftarrow a_3$



- An abstract value is a set of configurations. A lattice element is associated to each of them



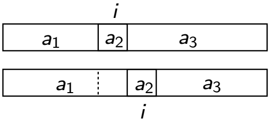
	2nd	3th
R2	$m = a_1$	
R6	$m = a_2 > a_1$	
R7	$a_1 = m \geq a_2$	
R4	$m \geq a_1, m \geq a_2$	
R5	$i = 2, n = 1$ $m = a_1$	

Symbolic Partitioning & Summarization

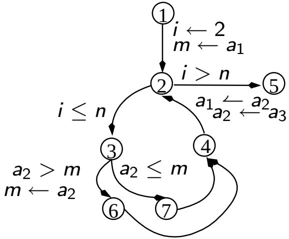
[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i + 1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$

i $\leftarrow i + 1$



- An abstract value is a set of configurations. A lattice element is associated to each of them

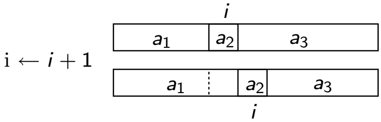


	2nd	3th
R2	$m = a_1$	$m \geq a_1$
R6	$m = a_2 > a_1$	
R7	$a_1 = m \geq a_2$	
R4	$m \geq a_1, m \geq a_2$	
R5	$i = 2, n = 1$ $m = a_1$	

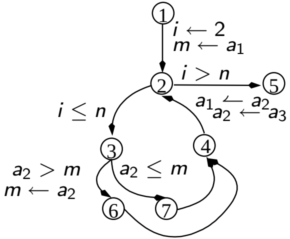
Symbolic Partitioning & Summarization

[Gopan, Reps, Sagiv - POPL'05]

- Partition each array into symbolic slices
  $A_1 = A[1..i-1], A_2 = A[i], A_3 = A[i+1..n]$
- Abstract slice A_p by a single variable a_p
- Interpretation
  $\psi(a_p) \Leftrightarrow \forall \ell \in I_p, \psi(A[\ell])$
- Assignment $A[i] \leftarrow exp$ is $a_2 \leftarrow exp$.
- Incrementation $i \leftarrow i+1$ is
  $a_1 \leftharpoondown a_2; a_2 \leftarrow a_3$



- An abstract value is a set of configurations. A lattice element is associated to each of them



	2nd	3th
R2	$m = a_1$	$m \geq a_1$
R6	$m = a_2 > a_1$	...
R7	$a_1 = m \geq a_2$	...
R4	$m \geq a_1, m \geq a_2$	...
R5	$i = 2, n = 1$ $m = a_1$	$n = i + 1$ $m \geq a_1$

Conclusions

- able to discover unary properties about array elements
- unable to discover relations between array elements
- able to check (with PVLA) such relations, provided by the user. e.g. $\forall \ell = 1..n, A[\ell] = B[\ell]$

This Work

- Generalization to discover relations with shifts
  $\forall \ell \in I, \psi\,(A1[\ell + k_1], \ldots, Am[\ell + k_m])$
- Clear element-wise relations : only between shifts of a same array slice (LUSTRE-V4)
  ~~$A[1..i] = A[i]$~~, $A[i] = A[i-1], A[1..i-1] < A[2..i], A[1..i] \leq 5^i$
- Symbolic slices as formulas for better manipulation
- Lost information in weak assignement reduced
- Contents are not always numerics!

This Work Is on Simple Programs

- one-dimensional arrays
- simple traversal: $i \leftarrow exp$; while(cond){$\ldots$; $i \leftarrow i \pm 1$}
- simple array access: A[i] := exp(B[i+k])

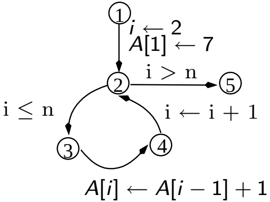
```
A[1] := 7 ;
for i := 2 to n do
    A[i] := A[i-1]+1
```

```
for i := 2 to n do
    x := A[i]; j := i - 1 ;
    while j ≥ 1 and A[j] > x
    do
        A[j + 1] := A[j] ;
        j := j - 1
    A[j + 1] := x
```

```
x := A[1] ; i := 1 ; j := n ;
while i ≤ j do
    if A[i] ≤ x then
        A[i - 1] := A[i] ;
        i := i + 1
    else
        while j ≥ i and
        A[j] ≥ x do
            j := j - 1
        if j > i then
            A[i - 1] := A[j];
            A[j] := A[i] ;
            i := i + 1 ;
            j := j - 1
A[i - 1] := x ;
```

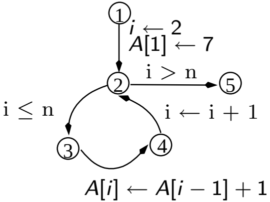
Abstract Values

- We keep a formula ($\in L_N$) over indices
- Symbolic slices are formulas ($\in L_N$) over indices $\mathcal{I}$ more a quantified symbol $\mathcal{I} \cup \{\ell\}$
  $\varphi_1 = (1 \leq \ell < i), \varphi_2 = (1 \leq \ell = i)$,
  $\varphi_3 = (1 \leq i < \ell \leq n)$
- Attached to each slice p, a formula ψ_p ($\in L_C$) over slice variables.
- Slice variable a^z in φ_p represents array slice $A[\ell + z], \varphi_p(\ell)$, x represents scalar expansion to array $x^{|\varphi_p|}$
- If $\varphi \Rightarrow \neg(\exists \ell \varphi_p)$, ψ_p is whatever. False!
  $\forall \ell,\ \ell \in \emptyset\ \Rightarrow\ \text{False}(\ell)$
- Interpretation, on P, $\Psi = (\varphi, (\psi_p)_{p \in P})$
  $\varphi(\mathcal{I}) \wedge$
  $\forall p \in P, \forall \ell,$
  $\varphi_p(\mathcal{I} \cup \{\ell\})\ \Rightarrow \psi_p[A[\ell + z]/a_p^z]$



Abstract Values

- We keep a formula ($\in L_N$) over indices
- Symbolic slices are formulas ($\in L_N$) over indices $\mathcal{I}$ more a quantified symbol $\mathcal{I} \cup \{\ell\}$
  $\varphi_1 = (1 \leq \ell < i), \varphi_2 = (1 \leq \ell = i),$
  $\varphi_3 = (1 \leq i < \ell \leq n)$
- Attached to each slice p, a formula ψ_p ($\in L_C$) over slice variables.
- Slice variable a^z in φ_p represents array slice $A[\ell + z], \varphi_p(\ell)$, x represents scalar expansion to array $x^{|\varphi_p|}$
- If $\varphi \Rightarrow \neg(\exists \ell \varphi_p)$, ψ_p is whatever. False!
  $\forall \ell,\ \ell \in \emptyset\ \Rightarrow\ \text{False}(\ell)$
- Interpretation, on P, $\Psi = (\varphi, (\psi_p)_{p \in P})$
  $\varphi(\mathcal{I}) \wedge$
  $\forall p \in P, \forall \ell,$
  $\varphi_p(\mathcal{I} \cup \{\ell\})\ \Rightarrow \psi_p[A[\ell + z]/a_p^z]$



$(i = n + 1),$
$\psi_1 = (a_1^0 = b_1^0), \psi_2 = \psi_3 = \bot_C$

$\varphi_1 = (\ell = 1), \varphi_2 = (2 \leq \ell < i)$
$\varphi_3 = (\ell = i), \varphi_4 = (i < \ell \leq n)$

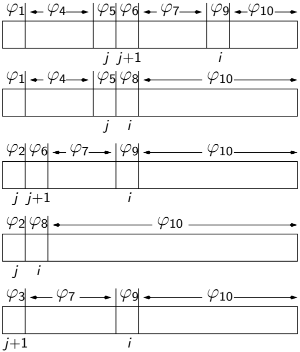
$(2 \leq i \leq n),$
$\psi_1 = (a_1^0 \leq a_1^1), \psi_2 = (a_2^0 \geq a_2^{-1})$
$\psi_3 = (a_3^0 = x), \psi_4 = \top_C$

Example of analysis

Operators through the family L(P)

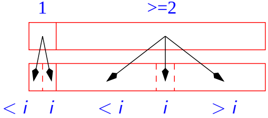
- a landmark : constant or index expression $i + k$ $(k \in \mathbb{Z})$ such that $A[i + k]$ appears either as the left-hand side of an assignment or in the condition of a test.

$$
\begin{aligned}
\varphi_1 &= (1 = \ell < j < i)\\
\varphi_2 &= (1 = j = \ell < i)\\
\varphi_3 &= (1 = j + 1 = \ell < i)\\
\varphi_4 &= (2 \leq \ell < j)\\
\varphi_5 &= (2 \leq j = \ell < i)\\
\varphi_6 &= (2 \leq j + 1 = \ell < i)\\
\varphi_7 &= (2 \leq j + 1 < \ell < i)\\
\varphi_8 &= (2 \leq \ell = j + 1 = i)\\
\varphi_9 &= (2 \leq j + 1 < \ell = i)\\
\varphi_{10} &= (2 \leq j + 1 \leq i < \ell)
\end{aligned}
$$

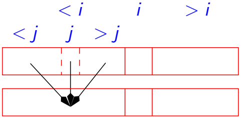


Operators through the family L(P)

- a landmark : constant or index expression $i + k$ $(k \in \mathbb{Z})$ such that $A[i + k]$ appears either as the left-hand side of an assignment or in the condition of a test.
  - Partitioning : when you reach the scope of a landmark

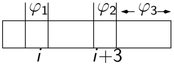


  - Merging (wrt an index, not a set of symbolic slices) : linked to the live status of the index



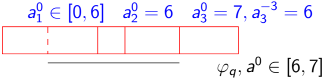
Operators into L(P)

- normalization: consistency on shifts



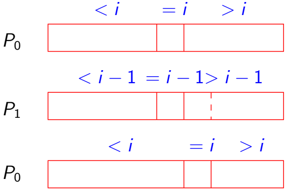
$$
\begin{array}{lcl}
\psi_1 & = & (a = x) \\
\psi_2 & = & (a = a^{-3}) \\
\psi_3 & = & (a > x, a^{-1} \geq x, a \geq a^{-1}) \\
\hline
\text{normalization} & & \\
\hline
\psi_1 & = & (a = x, a = a^3, a^3 = x) \\
\psi_2 & = & (a^{-3} = x, a = x, a = a^{-3}) \\
\psi_3 & = & (a > x, a^{-1} \geq x, a \geq a^{-1})
\end{array}
$$

- properties of a symbolic slice φ_p

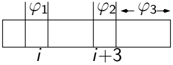


Operators into L(P)

- index change



- content assignement (aliasing avoided!)

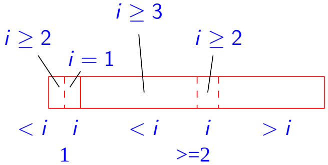


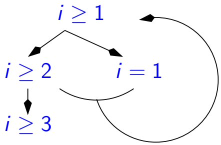
$$\begin{array}{lcl}\psi_1 & = & (a = x)\\ \psi_2 & = & (a > x)\\ \psi_3 & = & (a > x, a^{-1} > x, a \geq a^{-1})\end{array}$$

content assignment $A[i+3] := A[i]$

$$\begin{array}{lcl}\psi_1 & = & (a = x)\\ \psi_2 & = & (a = a^{-3})\\ \psi_3 & = & (a > x, a^{-1} \geq x, a \geq a^{-1})\end{array}$$

Contexts are Good for Non-Convex Analysis





Benchmarks and Future Work

	# *vert.* × # *edg.*	# φ_p	# *iter.*	*time (s)*
array copy	4 × 4	3	5	2
seq. init.	4 × 4	4	5	4
max. search	5 × 6	4	5	4
insert. sort	9 × 11	4-10	8	105
find	8 × 11	20	6	315

- improve the implementation
- more general programs ("for" loops with steps, recursivity...)
- more general properties (non convex slices)
- multi-dimensional arrays?
- generalization to function properties?
- properties about (multi-)sets of array values