Flow2Vec: Value-Flow-Based Precise Code Embedding

Yulei Sui[1], Xiao Cheng[2], Guanqin Zhang[1], Haoyu Wang[2]

[1] University of Technology Sydney, Australia
[2] Beijing University of Posts and Telecommunications, China

November 17 2020

Contribution

A new code embedding approach which marries **precise static analysis** and recent advances in **high-order proximity embedding**, by preserving

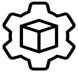
- **interprocedural alias-aware program dependence**, and
- **context-free language reachability**,

to better support subsequent code analysis tasks, such as code summarization and semantic labeling.

Code Embedding

Source Code | **Model** | **Code Property Prediction**

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```



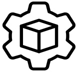
Code Embedding

Source Code

Model

Semantic Label

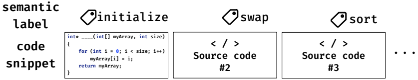
```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```



initialize

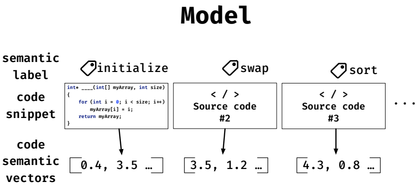
Code Embedding

Model



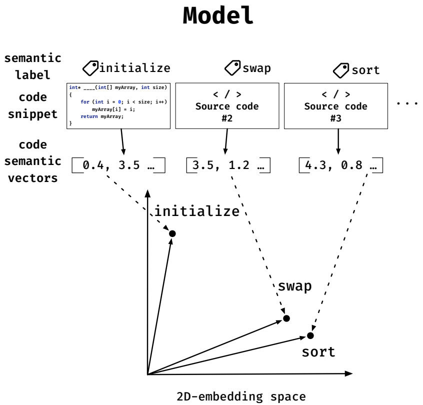
Code Embedding

Code semantic vector in geometric space



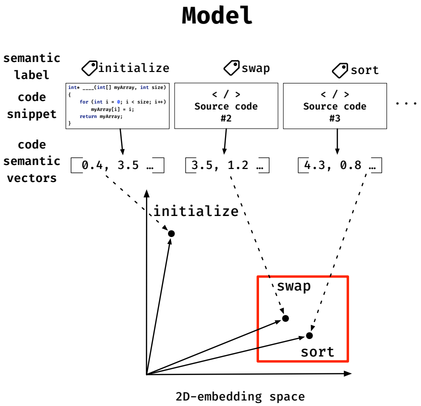
Code Embedding

Code semantic vector in geometric space



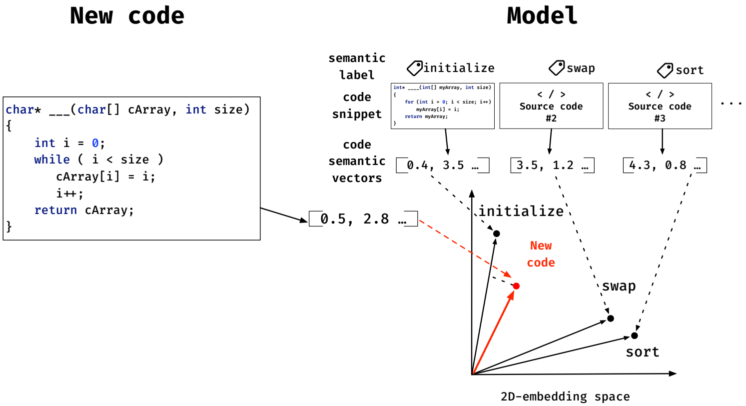
Code Embedding

Code semantic vector in geometric space



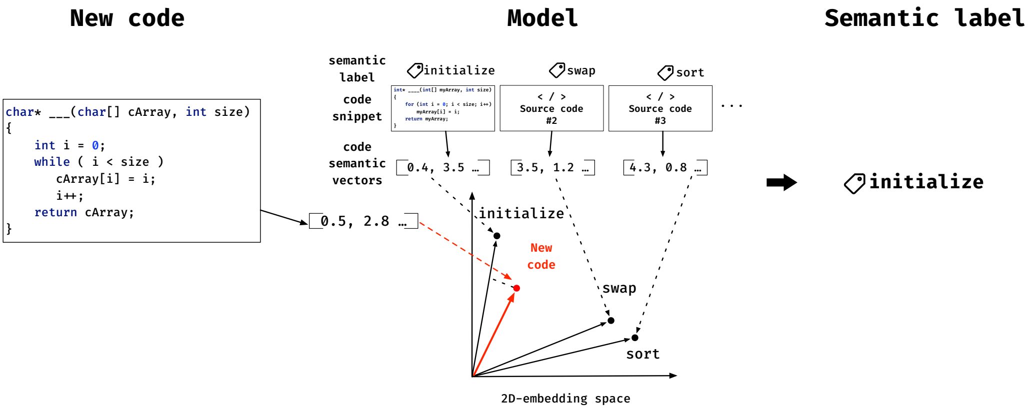
Code Embedding

Code semantic vector in geometric space



Code Embedding

Code semantic vector in geometric space



Existing Embedding Approaches

Structure-oblivious embedding

Source code ➡ **A bag of 'sentences'**[1,2]

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

[1] Distributed representations of words and phrases and their compositionality. In NeurIPS '13

[2] Distributed representations of sentences and documents. In ICML '14

Existing Embedding Approaches

Structure-oblivious embedding

Source code

A bag of ‘sentences’ [1,2]

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

➡

```
int*____(int[]myArray,int size)
```

[1] Distributed representations of words and phrases and their compositionality. In NeurIPS ’13

[2] Distributed representations of sentences and documents. In ICML ‘14

Existing Embedding Approaches

Structure-oblivious embedding

Source code

A bag of 'sentences' [1,2]

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

→

int	*	____	(	int	[]	myArray	,	int	size	)

[1] Distributed representations of words and phrases and their compositionality. In NeurIPS '13

[2] Distributed representations of sentences and documents. In ICML '14

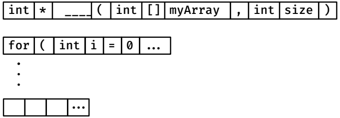
Existing Embedding Approaches

Structure-oblivious embedding

Source code

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

A bag of 'sentences' [1,2]



[1] Distributed representations of words and phrases and their compositionality. In NeurIPS '13

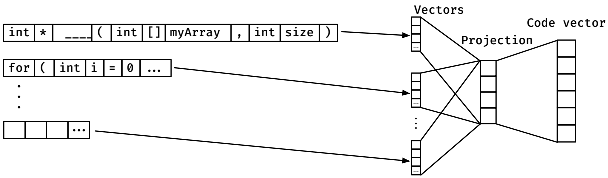
[2] Distributed representations of sentences and documents. In ICML '14

Existing Embedding Approaches

Structure-oblivious embedding

Source code **A bag of 'sentences'** [1,2] **Embedding**

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```



[1] Distributed representations of words and phrases and their compositionality. In NeurIPS '13

[2] Distributed representations of sentences and documents. In ICML '14

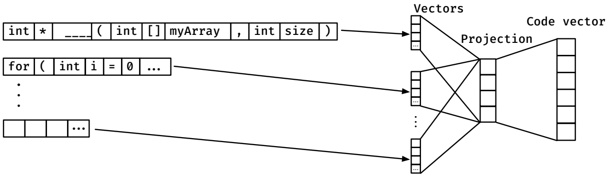
Existing Embedding Approaches

Structure-oblivious embedding

Source code **A bag of ‘sentences’** [1,2] **Embedding**

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}

char* ___(char[] cArray, int size)
{
    int i = 0;
    while ( i < size )
       cArray[i] = i;
       i++;
    return cArray;
}
```



Textually different but semantically equivalent

[1] Distributed representations of words and phrases and their compositionality. In NeurIPS ’13

[2] Distributed representations of sentences and documents. In ICML ‘14

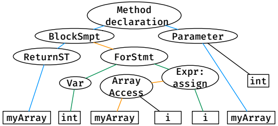
Existing Embedding Approaches

Structure-preserving embedding

Source code

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

Abstract Syntax Tree [3]



[3] code2vec: Learning distributed representations of code. POPL .2019

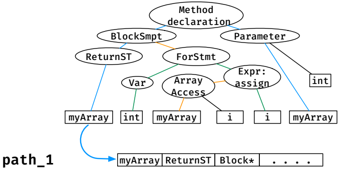
Existing Embedding Approaches

Structure-preserving embedding

Source code

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

A bag of 'paths' on AST [3]



[3] code2vec: Learning distributed representations of code. POPL .2019

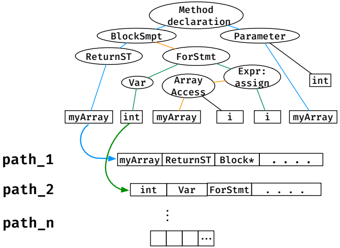
Existing Embedding Approaches

Structure-preserving embedding

Source code

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

A bag of ‘paths’ on AST [3]



[3] code2vec: Learning distributed representations of code. POPL .2019

Existing Embedding Approaches

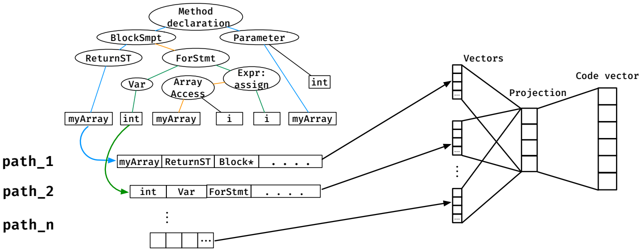
Structure-preserving embedding

Source code

```
int* ____(int[] myArray, int size)
{
    for (int i = 0; i < size; i++)
        myArray[i] = i;
    return myArray;
}
```

A bag of 'paths' on AST [3]

Embedding



[3] code2vec: Learning distributed representations of code. POPL .2019

Problems and Limitations

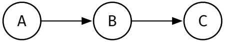
(a) Fail to capture asymmetric transitivity

(b) Alias-unaware

(c) Intraprocedural/context-insensitivity

Problems and Limitations

(a) Fail to capture asymmetric transitivity



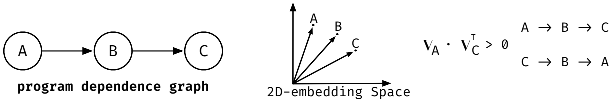
program dependence graph

(b) Alias-unaware

(c) Intraprocedural/context-insensitivity

Problems and Limitations

(a) Fail to capture asymmetric transitivity

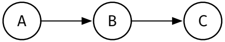


(b) Alias-unaware

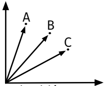
(c) Intraprocedural/context-insensitivity

Problems and Limitations

(a) Fail to capture asymmetric transitivity



program dependence graph



2D-embedding Space

$V_A \cdot V_C^T > 0$

A → B → C ✓ **Real reachability and correctly preserved**

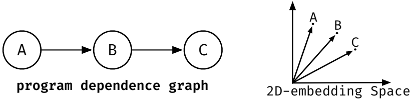
C → B → A ✗ **Spurious reachability but imprecisely preserved**

(b) Alias-unaware

(c) Intraprocedural/context-insensitivity

Problems and Limitations

(a) Fail to capture asymmetric transitivity

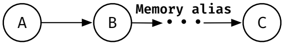


$V_A \cdot V_C^T > 0$

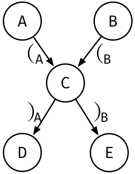
A → B → C ✓ Real reachability and correctly preserved

C → B → A ✗ Spurious reachability but imprecisely preserved

(b) Alias-unaware

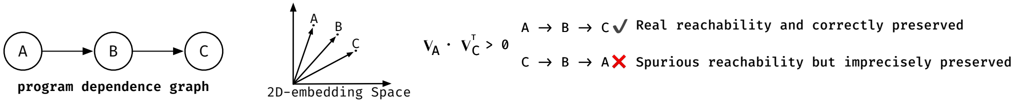


(c) Intraprocedural/context-insensitivity

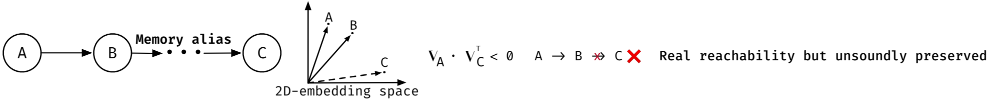


Problems and Limitations

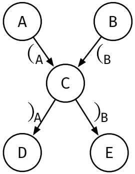
(a) Fail to capture asymmetric transitivity



(b) Alias-unaware

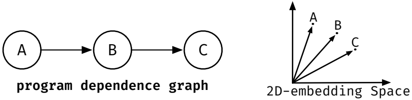


(c) Intraprocedural/context-insensitivity



Problems and Limitations

(a) Fail to capture asymmetric transitivity

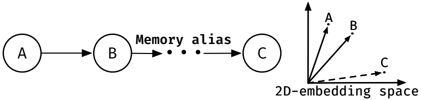


$V_A \cdot V_C^T > 0$

A → B → C ✔ Real reachability and correctly preserved

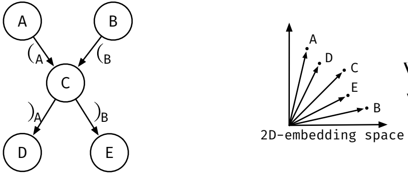
C → B → A ✘ Spurious reachability but imprecisely preserved

(b) Alias-unaware



$V_A \cdot V_C^T < 0$ A → B ↛ C ✘ Real reachability but unsoundly preserved

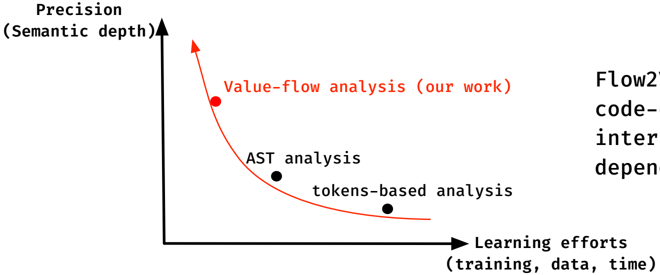
(c) Intraprocedural/context-insensitivity



$V_A \cdot V_D^T > 0$ A → C → D ✔ Real reachability and correctly preserved

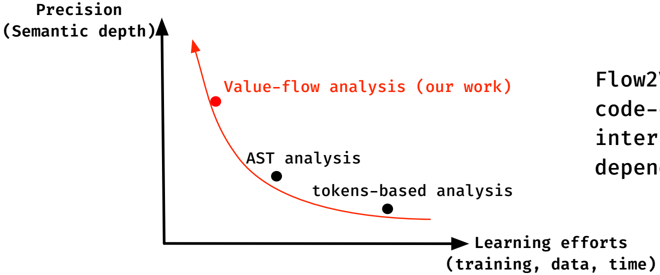
$V_A \cdot V_E^T > 0$ A → C → E ✘ Spurious reachability but imprecisely preserved

The Aim of This Work

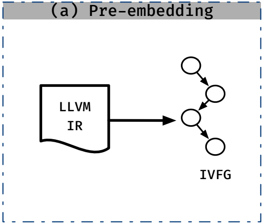


Flow2Vec: a high-order proximity code-embedding approach by preserving interprocedural alias-aware program dependence

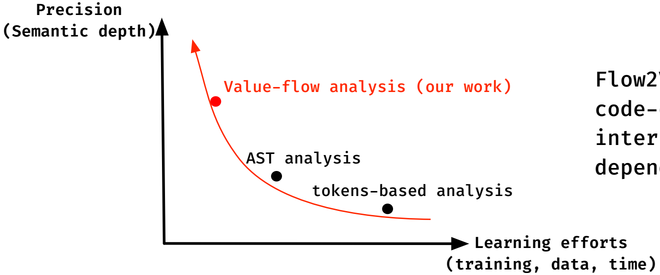
The Aim of This Work



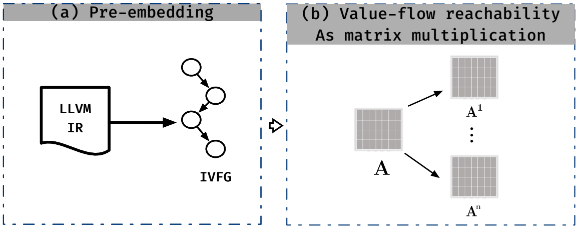
Flow2Vec: a high-order proximity code-embedding approach by preserving interprocedural alias-aware program dependence



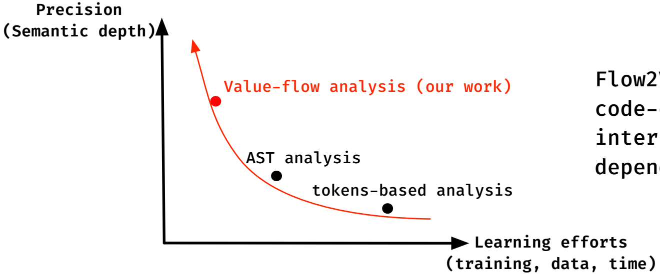
The Aim of This Work



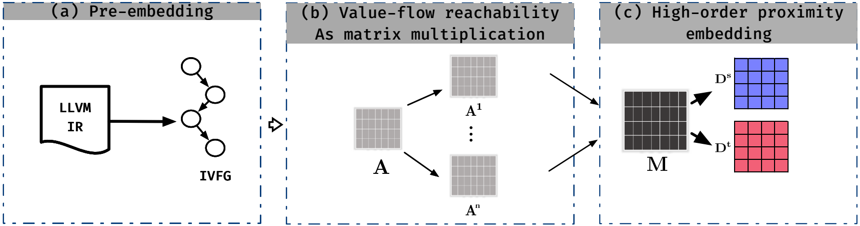
Flow2Vec: a high-order proximity code-embedding approach by preserving interprocedural alias-aware program dependence



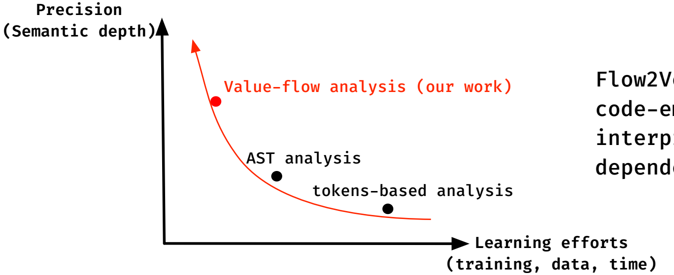
The Aim of This Work



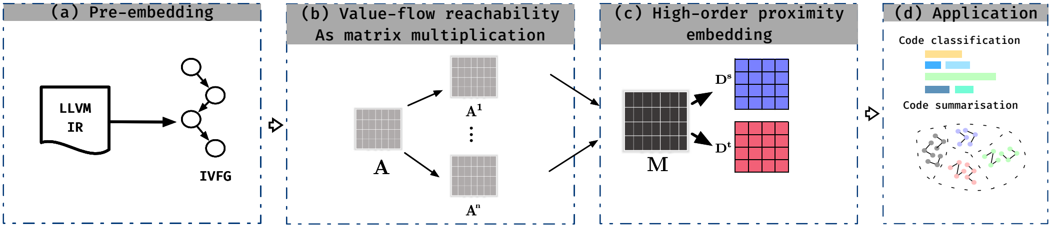
Flow2Vec: a high-order proximity code-embedding approach by preserving interprocedural alias-aware program dependence



The Aim of This Work



Flow2Vec: a high-order proximity code-embedding approach by preserving interprocedural alias-aware program dependence



A Motivating Example

Phase (a) Pre-embedding

```
    foo(){
l1:   stack = malloc(...);
l2:   queue = malloc(...);
l3:   p = initialize(stack); // cs1
l4:   q = initialize(queue); // cs2


      ……
    }
l5: initialize(x){
      // initialization for
      // objects that 'x' points to
      ……
l6:   return x;
    }
```

A Motivating Example

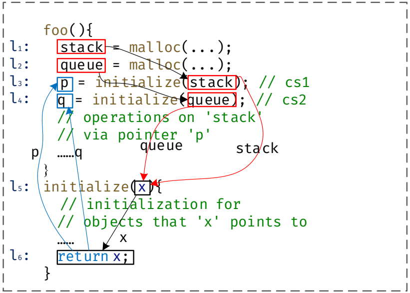
Phase (a) Pre-embedding

```
    foo(){
l1:   stack = malloc(...);
l2:   queue = malloc(...);
l3:   p = initialize(stack); // cs1
l4:   q = initialize(queue); // cs2
      // operations on 'stack'
      // via pointer 'p'
      ......
    }
l5: initialize(x){
      // initialization for
      // objects that 'x' points to
      ......
l6:   return x;
    }
```

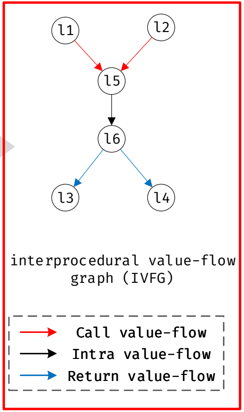
Call site 1	p refers to stack
Call site 2	q refers to queue

A Motivating Example

Phase (a) Pre-embedding

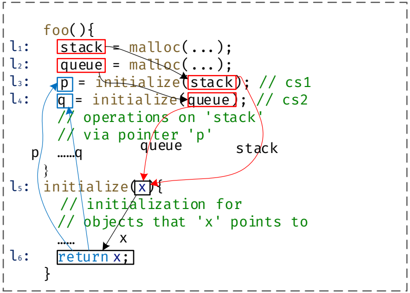


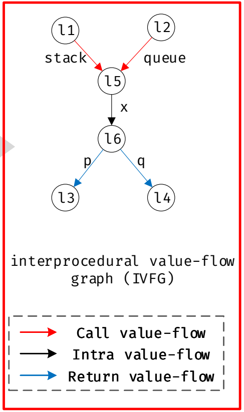
Call site 1	p refers to stack
Call site 2	q refers to queue



A Motivating Example

Phase (a) Pre-embedding

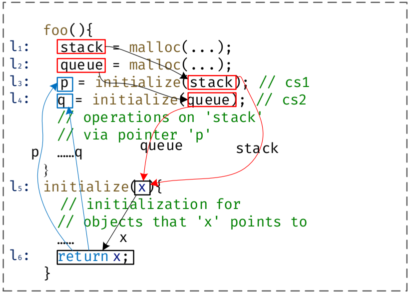




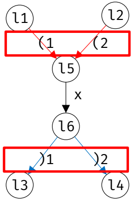
Call site 1	p refer to stack
Call site 2	q refer to queue

A Motivating Example

Phase (a) Pre-embedding



Call site 1	p refers to stack
Call site 2	q refers to queue

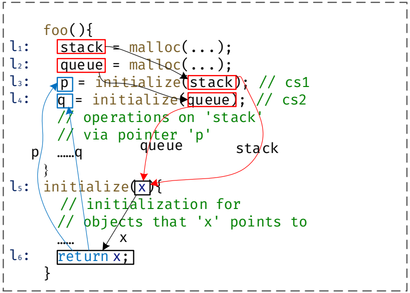


interprocedural value-flow graph (IVFG)

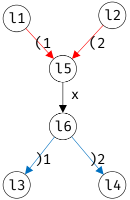
- Call value-flow
- Intra value-flow
- Return value-flow

A Motivating Example

Phase (a) Pre-embedding



Call site 1	p refers to stack
Call site 2	q refers to queue



interprocedural value-flow graph (IVFG)

→	Call value-flow
→	Intra value-flow
→	Return value-flow

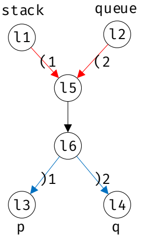
$$
\begin{array}{c} \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\left[\begin{array}{cccccc}
0 & 0 & 0 & 0 & (_1 & 0 \\
0 & 0 & 0 & 0 & (_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 &)_1 &)_2 & 0 & 0
\end{array}\right]
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

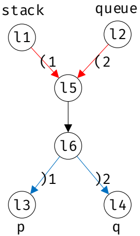
$$
\begin{array}{c}
 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & (_1 & 0 \\
0 & 0 & 0 & 0 & (_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 &)_1 &)_2 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

$$\begin{array}{c|c} & \begin{matrix} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{matrix} \\ \begin{matrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & (_1 & 0 \\ 0 & 0 & 0 & 0 & (_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 &)_1 &)_2 & 0 & 0 \end{bmatrix} \end{array}$$

$\mathbf{A}$

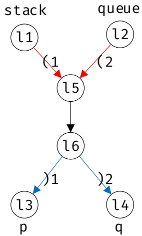
Symbolic Adjacency matrix

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

$$\begin{array}{c|c} & \begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\ \begin{array}{c} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & (_1 & 0 \\ 0 & 0 & 0 & 0 & (_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 &)_1 &)_2 & 0 & 0 \end{bmatrix} \end{array}$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$\begin{array}{c|c} & \begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\ \begin{array}{c} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & (_1 * 1 \\ 0 & 0 & 0 & 0 & 0 & (_2 * 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 *)_1 & 1 *)_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$

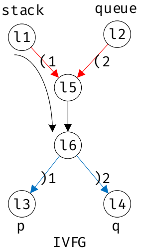
Matrix multiplication

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



$$
\begin{array}{c}
 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & (_1 & 0 \\
0 & 0 & 0 & 0 & (_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 &)_1 &)_2 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c}
 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & \boxed{(_1 * 1} \\
0 & 0 & 0 & 0 & 0 & (_2 * 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 *)_1 & 1 *)_2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

$$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$$

Matrix multiplication

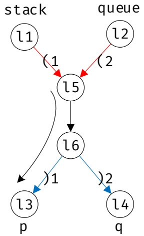
$$\boxed{\mathbf{A}^2_{1,6} = (_1 * 1 \quad l_1 \xrightarrow{(_1} l_5 \rightarrow l_6}$$

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

$$
\begin{array}{c|c}
 & \begin{matrix} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{matrix} \\
\begin{matrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{matrix} &
\begin{bmatrix}
0 & 0 & 0 & 0 & (_1 & 0 \\
0 & 0 & 0 & 0 & (_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 &)_1 &)_2 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c|c}
 & \begin{matrix} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{matrix} \\
\begin{matrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{matrix} &
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & (_1{*}1 \\
0 & 0 & 0 & 0 & 0 & (_2{*}1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \boxed{1{*})_1} & 1{*})_2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$

Matrix multiplication

$\mathbf{A}^2_{1,6} = (_1{*}1 \quad l_1 \xrightarrow{(_1} l_5 \rightarrow l_6$

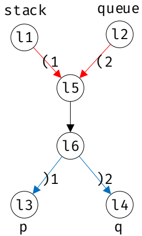
$\boxed{\mathbf{A}^2_{5,3} = 1{*})_1 \quad l_5 \rightarrow l_6 \xrightarrow{)_1} l_3}$

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\
l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\
l_6 & 0 & 0 &)_1 &)_2 & 0 & 0
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & (_1{*}1{*})_1 & (_1{*}1{*})_2 & 0 & 0 \\
l_2 & 0 & 0 & (_2{*}1{*})_1 & (_2{*}1{*})_2 & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$

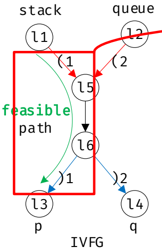
Matrix multiplication

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



$$
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{array}{c} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{array}
\begin{bmatrix}
0 & 0 & 0 & 0 & (_1 & 0 \\
0 & 0 & 0 & 0 & (_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 &)_1 &)_2 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{array}{c} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{array}
\begin{bmatrix}
0 & 0 & \boxed{(_1 {*} 1 {*})_1} & (_1 {*} 1 {*})_2 & 0 & 0 \\
0 & 0 & (_2 {*} 1 {*})_1 & (_2 {*} 1 {*})_2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$ Matrix multiplication

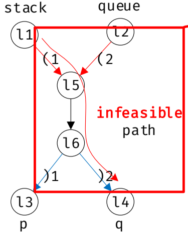
$\mathbf{A}^3_{1,3} = (_1 {*} 1 {*})_1$ $\quad l_1 \xrightarrow{(_1} l_5 \rightarrow l_6 \xrightarrow{)_1} l_3$ feasible path of length 3

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\
l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\
l_6 & 0 & 0 &)_1 &)_2 & 0 & 0
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & (_1{*}1{*})_1 & \boxed{(_1{*}1{*})_2} & 0 & 0 \\
l_2 & 0 & 0 & (_2{*}1{*})_1 & (_2{*}1{*})_2 & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$ Matrix multiplication

$\mathbf{A}^3_{1,3} = (_1{*}1{*})_1 \quad l1 \xrightarrow{(1} l5 \rightarrow l6 \xrightarrow{)1} l3$

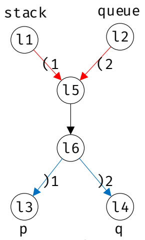
$\mathbf{A}^3_{1,4} = (_1{*}1{*})_2 \quad l1 \xrightarrow{(1} l5 \rightarrow l6 \xrightarrow{)2} l4$ infeasible path of length 3

$$\mathbf{A}^h = \underbrace{\mathbf{A} \cdot \ldots \cdot \mathbf{A}}_{\times h}$$

The power of matrix
h-th order reachability

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

$$
\begin{array}{c}
 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & (_1 & 0 \\
0 & 0 & 0 & 0 & (_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 &)_1 &)_2 & 0 & 0
\end{bmatrix}
\end{array}
$$

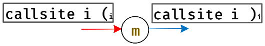
$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & \boxed{(_1 1^* 1^*)_1} & \cancel{(_1 1^* 1^*)_2} & 0 & 0 \\
l_2 & 0 & 0 & \cancel{(_2 1^* 1^*)_1} & \boxed{(_2 1^* 1^*)_2} & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$ Matrix multiplication

$L : C = CC \mid (_1\ C\ )_1 \mid \ldots \mid (_n\ C\ )_n \mid \epsilon$ context-free language

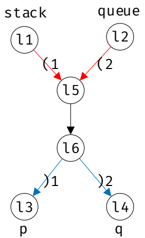


entering a method m from $\boxed{\text{callsite i } (_i}$

and exits from m back to $\boxed{\text{callsite i })_i}$

A Motivating Example

Phase (b) Value-flow reachability as matrix multiplication



IVFG

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\
l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\
l_6 & 0 & 0 &)_1 &)_2 & 0 & 0
\end{array}
$$

$\mathbf{A}$

Symbolic Adjacency matrix

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & \boxed{1} & \boxed{0} & 0 & 0 \\
l_2 & 0 & 0 & \boxed{0} & \boxed{1} & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$$

Matrix multiplication

A Motivating Example

Phase (c) High-order proximity embedding

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\
l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\
l_6 & 0 & 0 &)_1 &)_2 & 0 & 0
\end{array}
$$

$\mathbf{A}$

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & 0 & (_1{*}1 \\
l_2 & 0 & 0 & 0 & 0 & 0 & (_2{*}1 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 1{*})_1 & 1{*})_2 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & (_1{*}1{*})_1 & (_1{*}1{*})_2 & 0 & 0 \\
l_2 & 0 & 0 & (_2{*}1{*})_1 & (_2{*}1{*})_2 & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$

Katz Index $\mathbf{M} = \sum_{h=1}^{H} (\beta \cdot \mathbf{A})^h = 0.8 \cdot \mathbf{A} + 0.8^2 \cdot \mathbf{A}^2 + 0.8^3 \cdot \mathbf{A}^3$

A Motivating Example

Phase (c) High-order proximity embedding

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\
l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\
l_6 & 0 & 0 &)_1 &)_2 & 0 & 0
\end{array}
$$

$\mathbf{A}$

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & 0 & (_1*1 \\
l_2 & 0 & 0 & 0 & 0 & 0 & (_2*1 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 1*)_1 & 1*)_2 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & (_1*1*)_1 & (_1*1*)_2 & 0 & 0 \\
l_2 & 0 & 0 & (_2*1*)_1 & (_2*1*)_2 & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$

Katz Index

$$\mathbf{M} = \sum_{h=1}^{H} (\beta \cdot \mathbf{A})^h = \boxed{0.8} \cdot \mathbf{A} + \boxed{0.8^2} \cdot \mathbf{A}^2 + \boxed{0.8^3} \cdot \mathbf{A}^3$$

Context-sensitive value-flow reachability matrix

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0.512*(_1*1*)_1 & \cancel{0.512*(_1*1*)_2} & 0.8*(_1 & 0.64*(_1*1 \\
l_2 & 0 & 0 & \cancel{0.512*(_2*1*)_1} & 0.512*(_2*1*)_2 & 0.8*(_2 & 0.64*(_2*1 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0.64*1*)_1 & 0.64*1*)_2 & 0 & 0.8 \\
l_6 & 0 & 0 & 0.8*)_1 & 0.8*)_2 & 0 & 0
\end{array}
$$

A Motivating Example

Phase (c) High-order proximity embedding

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\
l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\
l_6 & 0 & 0 &)_1 &)_2 & 0 & 0
\end{array}
$$

$\mathbf{A}$

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0 & 0 & 0 & (_1*1 \\
l_2 & 0 & 0 & 0 & 0 & 0 & (_2*1 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 1*)_1 & 1*)_2 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & (_1*1*)_1 & (_1*1*)_2 & 0 & 0 \\
l_2 & 0 & 0 & (_2*1*)_1 & (_2*1*)_2 & 0 & 0 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$

Katz Index

$$\mathbf{M} = \sum_{h=1}^{H} (\beta \cdot \mathbf{A})^h = 0.8 \cdot \mathbf{A} + 0.8^2 \cdot \mathbf{A}^2 + 0.8^3 \cdot \mathbf{A}^3$$

Context-sensitive value-flow reachability matrix

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0.512 & 0 & 0.8 & 0.64 \\
l_2 & 0 & 0 & 0 & 0.512 & 0.8 & 0.64 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0.64 & 0.64 & 0 & 0.8 \\
l_6 & 0 & 0 & 0.8 & 0.8 & 0 & 0
\end{array}
$$

A Motivating Example

Phase (c) High-order proximity embedding

$$\begin{array}{c|cccccc} & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\ \hline l_1 & 0 & 0 & 0 & 0 & (_1 & 0 \\ l_2 & 0 & 0 & 0 & 0 & (_2 & 0 \\ l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_5 & 0 & 0 & 0 & 0 & 0 & 1 \\ l_6 & 0 & 0 &)_1 &)_2 & 0 & 0 \end{array}$$

$$\mathbf{A}$$

$$\begin{array}{c|cccccc} & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\ \hline l_1 & 0 & 0 & 0 & 0 & 0 & (_1 * 1 \\ l_2 & 0 & 0 & 0 & 0 & 0 & (_2 * 1 \\ l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_5 & 0 & 0 & 1 *)_1 & 1 *)_2 & 0 & 0 \\ l_6 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\mathbf{A}^2 = \mathbf{A} \cdot \mathbf{A}$$

$$\begin{array}{c|cccccc} & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\ \hline l_1 & 0 & 0 & (_1 * 1 *)_1 & (_1 * 1 *)_2 & 0 & 0 \\ l_2 & 0 & 0 & (_2 * 1 *)_1 & (_2 * 1 *)_2 & 0 & 0 \\ l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_5 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_6 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$\mathbf{A}^3 = \mathbf{A} \cdot \mathbf{A} \cdot \mathbf{A}$$

Katz Index

$$\mathbf{M} = \sum_{h=1}^{H} (\beta \cdot \mathbf{A})^h = 0.8 \cdot \mathbf{A} + 0.8^2 \cdot \mathbf{A}^2 + 0.8^3 \cdot \mathbf{A}^3$$

Context-sensitive value-flow reachability matrix

$$\begin{array}{c|cccccc} & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\ \hline l_1 & 0 & 0 & \boxed{0.512} & 0 & 0.8 & 0.64 \\ l_2 & 0 & 0 & 0 & 0.512 & 0.8 & 0.64 \\ l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_5 & 0 & 0 & 0.64 & 0.64 & 0 & 0.8 \\ l_6 & 0 & 0 & 0.8 & 0.8 & 0 & 0 \end{array}$$

Example: l_1 to l_3

$$l1 \xrightarrow{(_1} l5 \rightarrow l6 \xrightarrow{)_1} l3$$

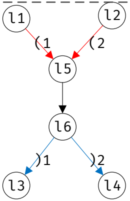
Reachability from l1 to l3:

$$\mathbf{M}_{1,3} = 0.8 \cdot \mathbf{A}_{1,3} + 0.8^2 \cdot \mathbf{A}^2_{1,3} + 0.8^3 \cdot \boxed{\mathbf{A}^3_{1,3}}$$

$$= 0.8 * 0 + 0.8^2 * 0 + 0.8^3 * \boxed{1} = 0.512$$

A Motivating Example

Phase (c) High-order proximity embedding



IVFG

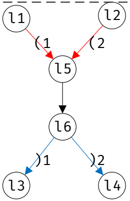
$$
\begin{array}{c|c}
 & \begin{matrix} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{matrix} \\
\begin{matrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \end{matrix} &
\begin{bmatrix}
0 & 0 & 0.512 & 0 & 0.8 & 0.64 \\
0 & 0 & 0 & 0.512 & 0.8 & 0.64 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.64 & 0.64 & 0 & 0.8 \\
0 & 0 & 0.8 & 0.8 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{M}$

High-order proximity Matrix

A Motivating Example

Phase (c) High-order proximity embedding



IVFG

$$
\begin{array}{c}
 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{bmatrix}
0 & 0 & 0.512 & 0 & 0.8 & 0.64 \\
0 & 0 & 0 & 0.512 & 0.8 & 0.64 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.64 & 0.64 & 0 & 0.8 \\
0 & 0 & 0.8 & 0.8 & 0 & 0
\end{bmatrix}
\end{array}
$$

$\mathbf{M}$

High-order proximity Matrix

Singular Value Decompose

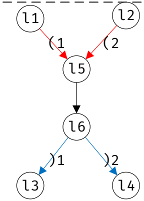
$$\mathbf{M} \approx \mathbf{U\Sigma V}^\top = \sum_{i=1}^{N} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \mathbf{D}^s \cdot \mathbf{D}^{t\top}$$

$$\mathbf{D}^s = \left[\sqrt{\sigma_1} \cdot \mathbf{u}_1, \cdots, \sqrt{\sigma_K} \cdot \mathbf{u}_K\right] = \left[\mathbf{d}_1^s, \cdots, \mathbf{d}_N^s\right]$$

$$\mathbf{D}^t = \left[\sqrt{\sigma_1} \cdot \mathbf{v}_1, \cdots, \sqrt{\sigma_K} \cdot \mathbf{v}_K\right] = \left[\mathbf{d}_1^t, \cdots, \mathbf{d}_N^t\right]$$

A Motivating Example

Phase (c) High-order proximity embedding



IVFG

$$
\begin{array}{c|cccccc}
 & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \\
\hline
l_1 & 0 & 0 & 0.512 & 0 & 0.8 & 0.64 \\
l_2 & 0 & 0 & 0 & 0.512 & 0.8 & 0.64 \\
l_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
l_5 & 0 & 0 & 0.64 & 0.64 & 0 & 0.8 \\
l_6 & 0 & 0 & 0.8 & 0.8 & 0 & 0
\end{array}
$$

$$\mathbf{M}$$

High-order proximity Matrix

Singular Value Decompose

$$\mathbf{M} \approx \mathbf{U\Sigma V}^\top = \sum_{i=1}^{N} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \mathbf{D}^s \cdot \mathbf{D}^{t\top}$$

$$\mathbf{D}^s = \left[\sqrt{\sigma_1}\cdot\mathbf{u}_1, \cdots, \sqrt{\sigma_K}\cdot\mathbf{u}_K\right] = \left[\mathbf{d}_1^s, \cdots, \mathbf{d}_N^s\right]$$

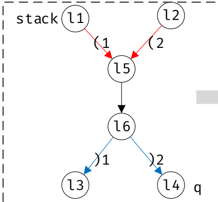
$$\mathbf{D}^t = \left[\sqrt{\sigma_1}\cdot\mathbf{v}_1, \cdots, \sqrt{\sigma_K}\cdot\mathbf{v}_K\right] = \left[\mathbf{d}_1^t, \cdots, \mathbf{d}_N^t\right]$$

$$
\left[\begin{array}{ccc|ccc}
-0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\
0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.33 & -0.79 & 0 & 0 & 0 \\
0 & 0.71 & -0.58 & 0 & 0 & 0
\end{array}\right]
\qquad
\left[\begin{array}{ccc|ccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\
0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\
0 & -0.71 & -0.58 & 0 & 0 & 0 \\
0 & -0.33 & -0.79 & 0 & 0 & 0
\end{array}\right]
$$

$$\mathbf{D}^s \qquad\qquad \mathbf{D}^t$$

A Motivating Example

Phase (c) High-order proximity embedding



IVFG

$$
\begin{array}{c}
 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6
\end{array}
\begin{array}{c}
\begin{array}{cccccc} l_1 & l_2 & l_3 & l_4 & l_5 & l_6 \end{array} \\
\begin{bmatrix}
0 & 0 & 0.512 & 0 & 0.8 & 0.64 \\
0 & 0 & 0 & 0.512 & 0.8 & 0.64 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.64 & 0.64 & 0 & 0.8 \\
0 & 0 & 0.8 & 0.8 & 0 & 0
\end{bmatrix}
\end{array}
$$

$$\mathbf{M}$$

High-order proximity Matrix

Singular Value Decompose

$$\mathbf{M} \approx \mathbf{U\Sigma V}^\top = \sum_{i=1}^{N} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \mathbf{D}^s \cdot \mathbf{D}^{t\top}$$

$$\mathbf{D}^s = \left[\sqrt{\sigma_1} \cdot \mathbf{u}_1, \cdots, \sqrt{\sigma_K} \cdot \mathbf{u}_K\right] = \left[\mathbf{d}_1^s, \cdots, \mathbf{d}_N^s\right]$$

$$\mathbf{D}^t = \left[\sqrt{\sigma_1} \cdot \mathbf{v}_1, \cdots, \sqrt{\sigma_K} \cdot \mathbf{v}_K\right] = \left[\mathbf{d}_1^t, \cdots, \mathbf{d}_N^t\right]$$

$$
\mathbf{D}^s = \left[\begin{array}{ccc:ccc}
-0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\
0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.33 & -0.79 & 0 & 0 & 0 \\
0 & 0.71 & -0.58 & 0 & 0 & 0
\end{array}\right]
\qquad
\mathbf{D}^t = \left[\begin{array}{ccc:ccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\
0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\
0 & -0.71 & -0.58 & 0 & 0 & 0 \\
0 & -0.33 & -0.79 & 0 & 0 & 0
\end{array}\right]
$$

Reachability between i and j

$$l_i \xrightarrow{\mathbf{d}_i^s \cdot \mathbf{d}_j^t} l_j$$

A Motivating Example

Phase (c) High-order proximity embedding

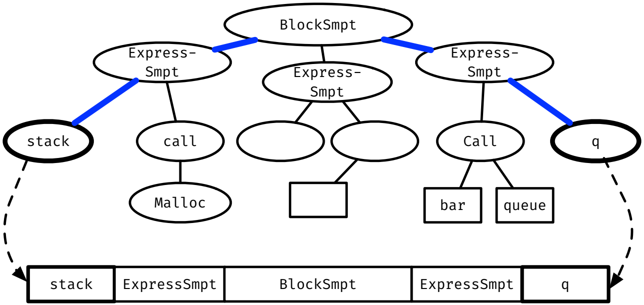
$$
\begin{array}{c}
\begin{array}{cc}
\mathbf{d}_1^s & \\
 & \\
 & \\
 & \\
 & \\
 &
\end{array}
\left[\begin{array}{ccc:ccc}
\boxed{-0.51 \quad -0.49 \quad -0.69} & & & 0 & 0 & 0 \\
0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.33 & -0.79 & 0 & 0 & 0 \\
0 & 0.71 & -0.58 & 0 & 0 & 0
\end{array}\right] \\
\mathbf{D}^s
\end{array}
$$

$$
\begin{array}{c}
\begin{array}{c}
 \\
 \\
\mathbf{d}_3^t \\
\mathbf{d}_4^t \\
\mathbf{d}_5^t \\
\mathbf{d}_6^t
\end{array}
\left[\begin{array}{ccc:ccc}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\
0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\
0 & -0.71 & -0.58 & 0 & 0 & 0 \\
0 & -0.33 & -0.79 & 0 & 0 & 0
\end{array}\right] \\
\mathbf{D}^t
\end{array}
$$

Reachability	Path length
$\mathbf{d}_1^s \cdot {\mathbf{d}_5^t}^\top = 0.75$	1
$\mathbf{d}_1^s \cdot {\mathbf{d}_6^t}^\top = 0.71$	2
$\mathbf{d}_1^s \cdot {\mathbf{d}_3^t}^\top = 0.5$	3
$\mathbf{d}_1^s \cdot {\mathbf{d}_4^t}^\top = -0.02$	infeasible

A Motivating Example

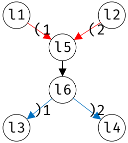
Phase (c) High-order proximity embedding



`infeasible dependence relation between `stack` to `q``

A Motivating Example

Phase (d) Value-Flow Vector and Applications



$$\left[\begin{array}{ccc|ccc} -0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\ 0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.33 & -0.79 & 0 & 0 & 0 \\ 0 & 0.71 & -0.58 & 0 & 0 & 0 \end{array}\right]$$

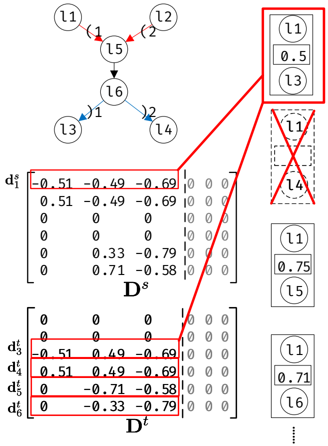
$$\mathbf{D}^s$$

$$\left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\ 0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\ 0 & -0.71 & -0.58 & 0 & 0 & 0 \\ 0 & -0.33 & -0.79 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{D}^t$$

A Motivating Example

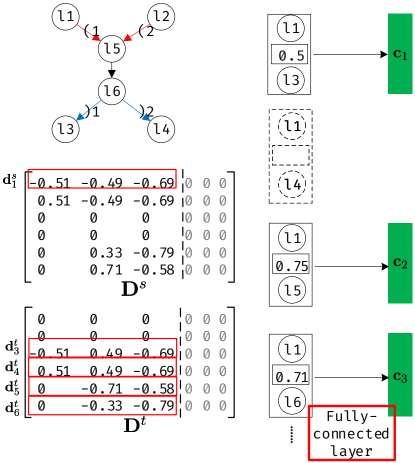
Phase (d) Value-flow Vectors and Applications



A Motivating Example

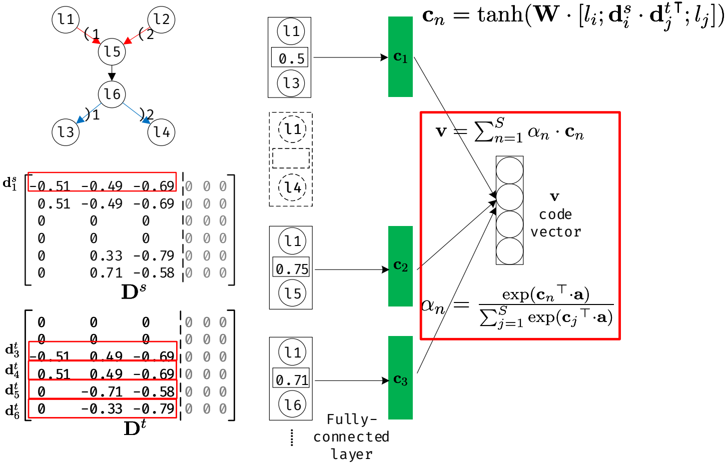
Phase (d) Value-flow Vectors and Applications

$$\mathbf{c}_n = \tanh(\mathbf{W} \cdot [l_i; \mathbf{d}_i^s \cdot \mathbf{d}_j^{t\top}; l_j])$$



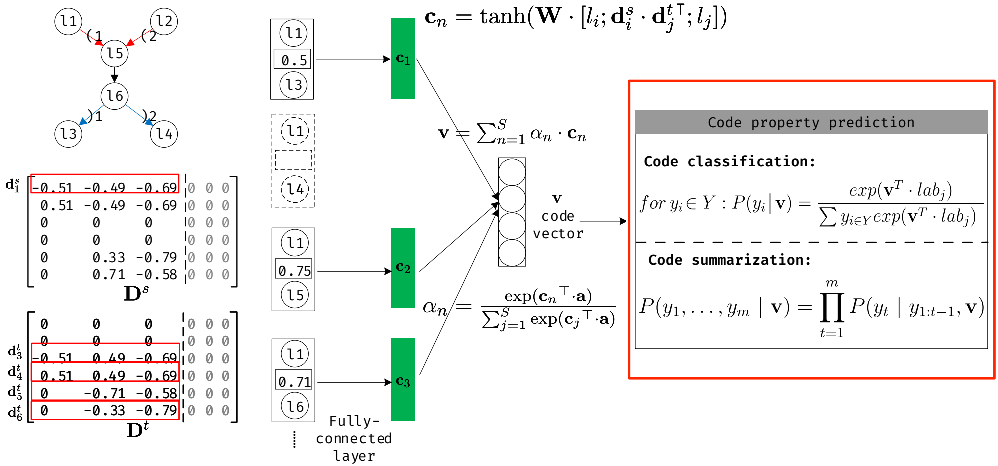
A Motivating Example

Phase (d) Value-flow Vectors and Applications



A Motivating Example

Phase (d) Value-flow Vectors and Applications



$$\mathbf{D}^s = \left[\begin{array}{ccc|ccc} -0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\ 0.51 & -0.49 & -0.69 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.33 & -0.79 & 0 & 0 & 0 \\ 0 & 0.71 & -0.58 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{D}^t = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\ 0.51 & 0.49 & -0.69 & 0 & 0 & 0 \\ 0 & -0.71 & -0.58 & 0 & 0 & 0 \\ 0 & -0.33 & -0.79 & 0 & 0 & 0 \end{array}\right]$$

Experimental Evaluation

Benchmarks

. . .

32 popular open-source C/C++ projects

Total Line of Instructions:4,922,162

Total Methods:17,529

Total Pointers: 2,913,748

Total Objects: 190,157

Total Number of Calls:536,033

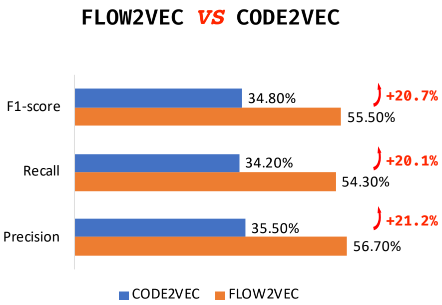
Total IVFGNodes: 4,637,301

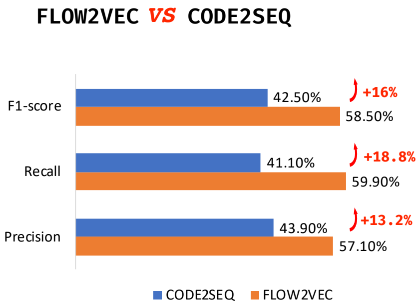
Total IVFGEdges: 6,531,578

*Conducted machine: Intel Xeon Gold 6132 @ 2.60GHz CPUs and 128GB of RAM (All finish analyzing in 272.5mins)

Experimental Evaluation

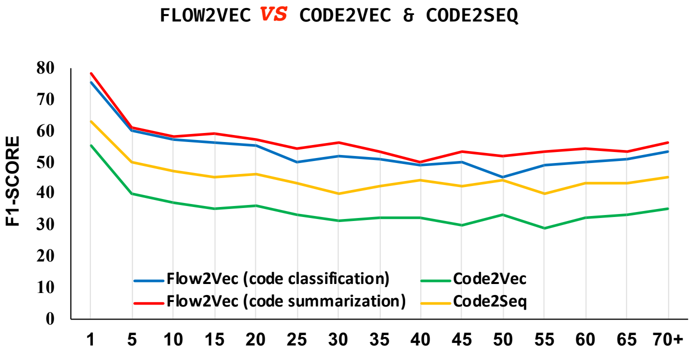
Comparison with baselines





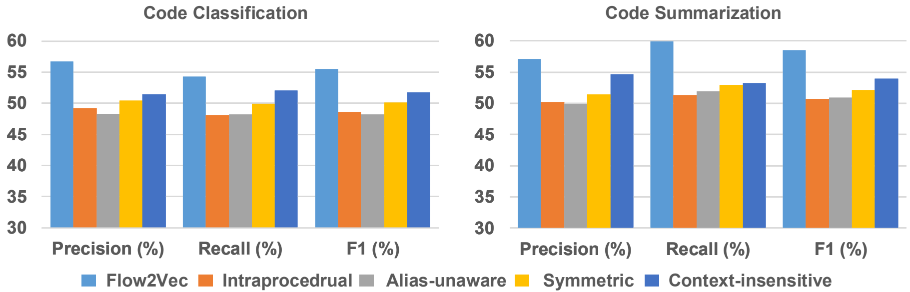
Experimental Evaluation

F1-score under different lengths of code



Experimental Evaluation

Ablation analysis



Thanks!

Q & A