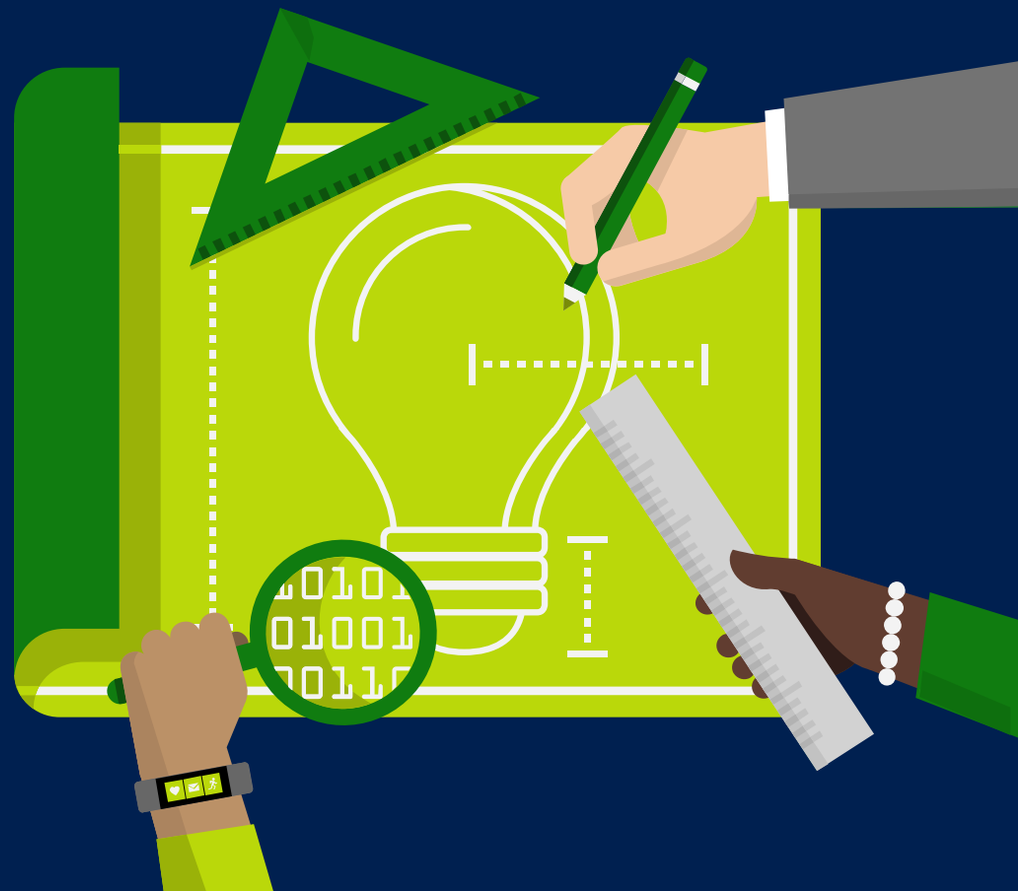
Microsoft

Detecting Variable Misuses with Graph Neural Networks

Miltos Allamanis

Joint work with Marc Brockschmidt and Mahmoud Khademi



https://arxiv.org/abs/1711.00740

Motivating Example

```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull( clazz );

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

Possible type-correct options: **`clazz, first`**

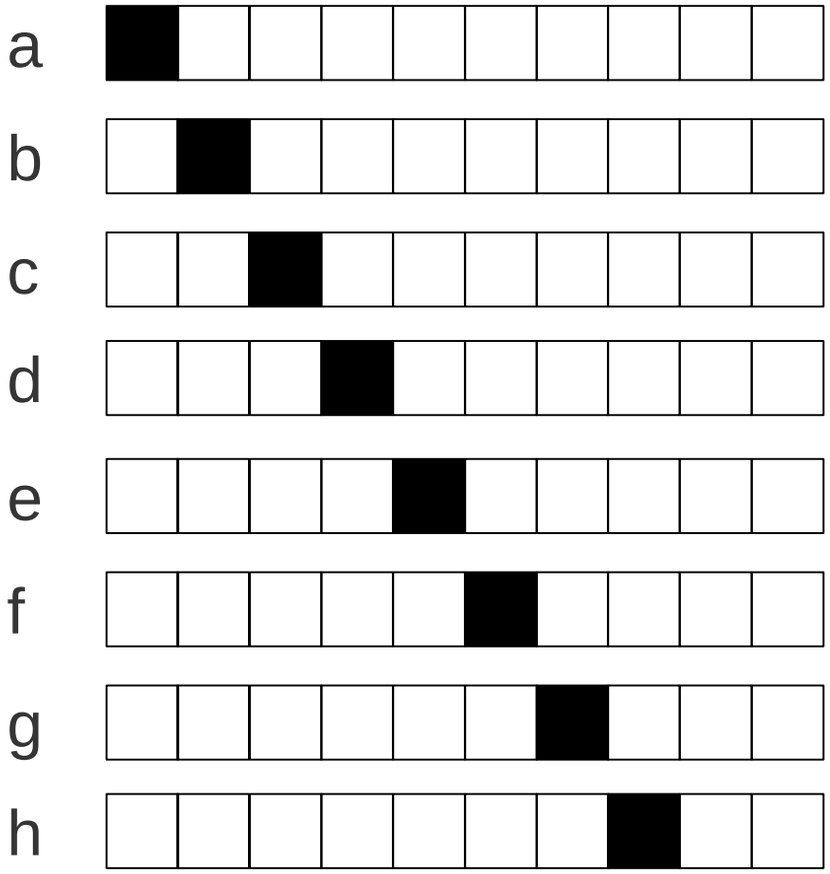
Microsoft

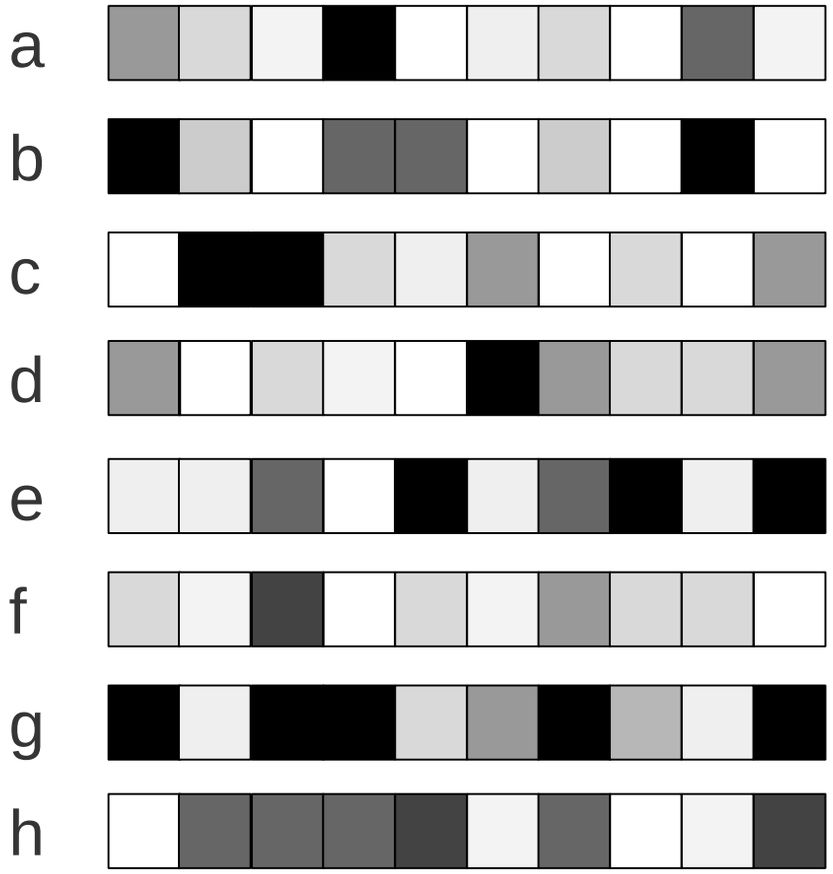
Graph Neural Networks

A brief introduction

Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Localized *vs* Distributed (Vector) Representations



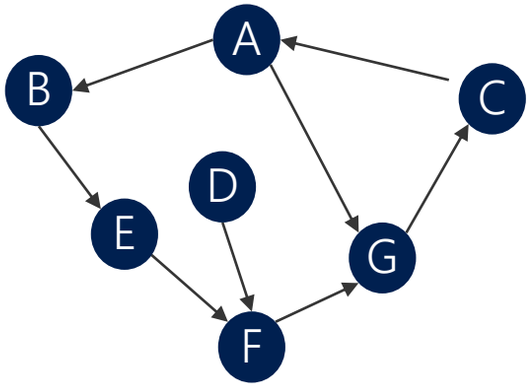


$$P_{\mathcal{D}}(\pi|f(\mathbb{c}))$$

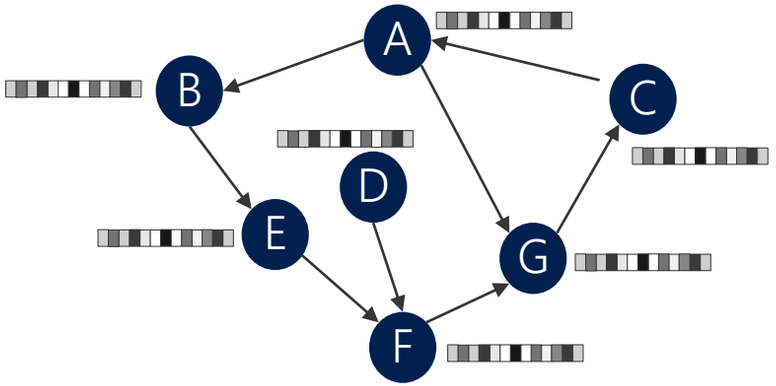
$$\downarrow$$

$$\mathbb{R}^D$$

Gated Graph Neural Networks



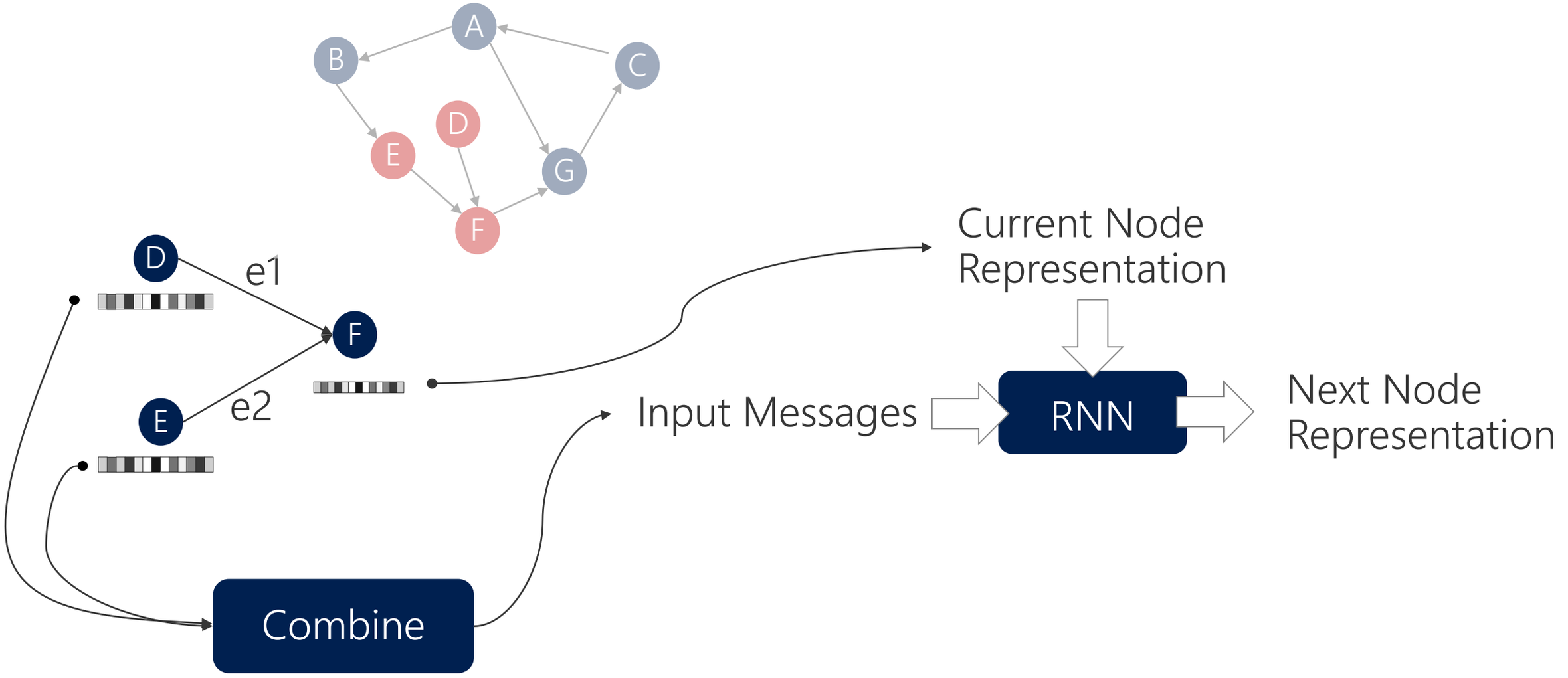
Graph Representation of Problem



Initial Representation of each node

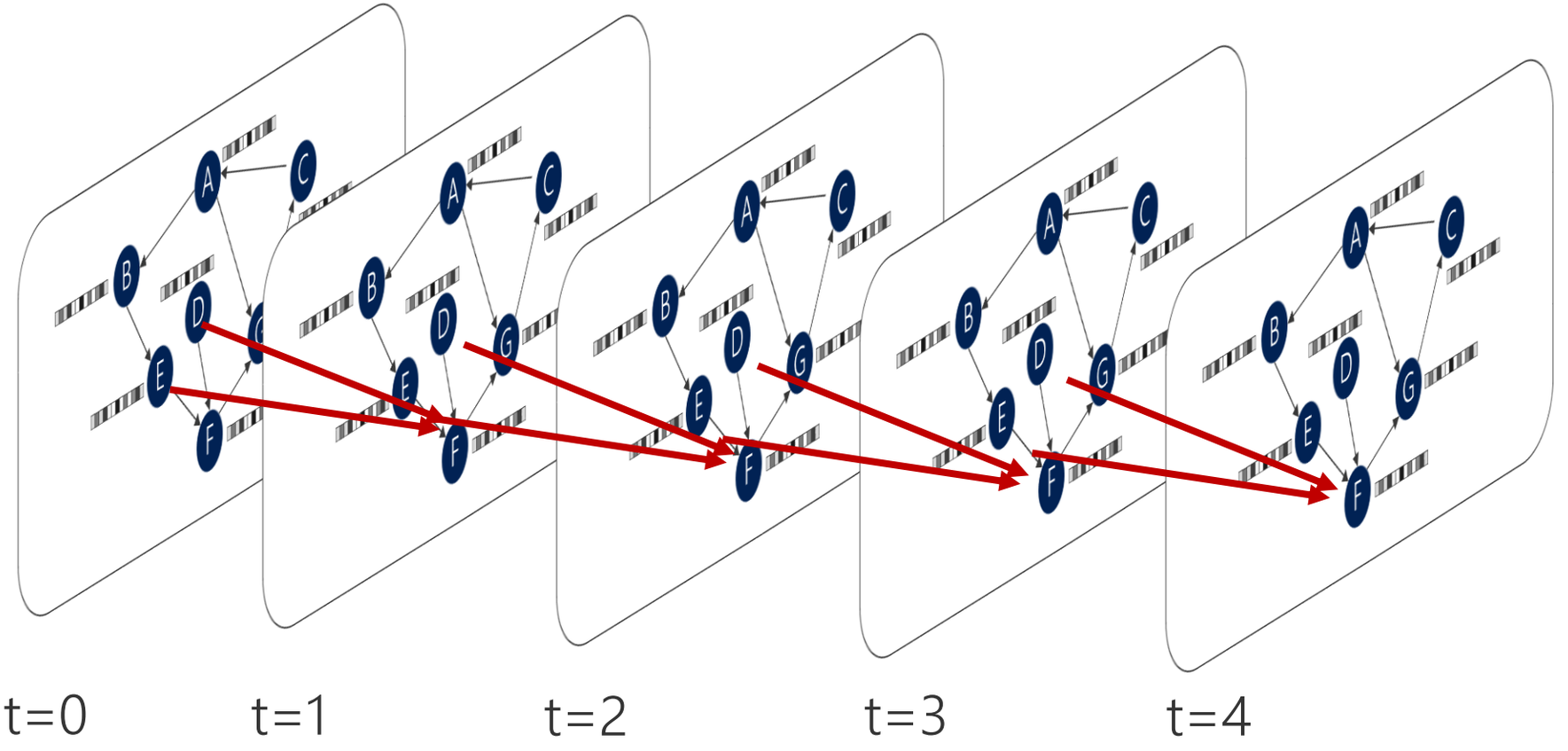
Li et al (2015). Gated graph sequence neural networks.

Graph Neural Networks: Message Propagation



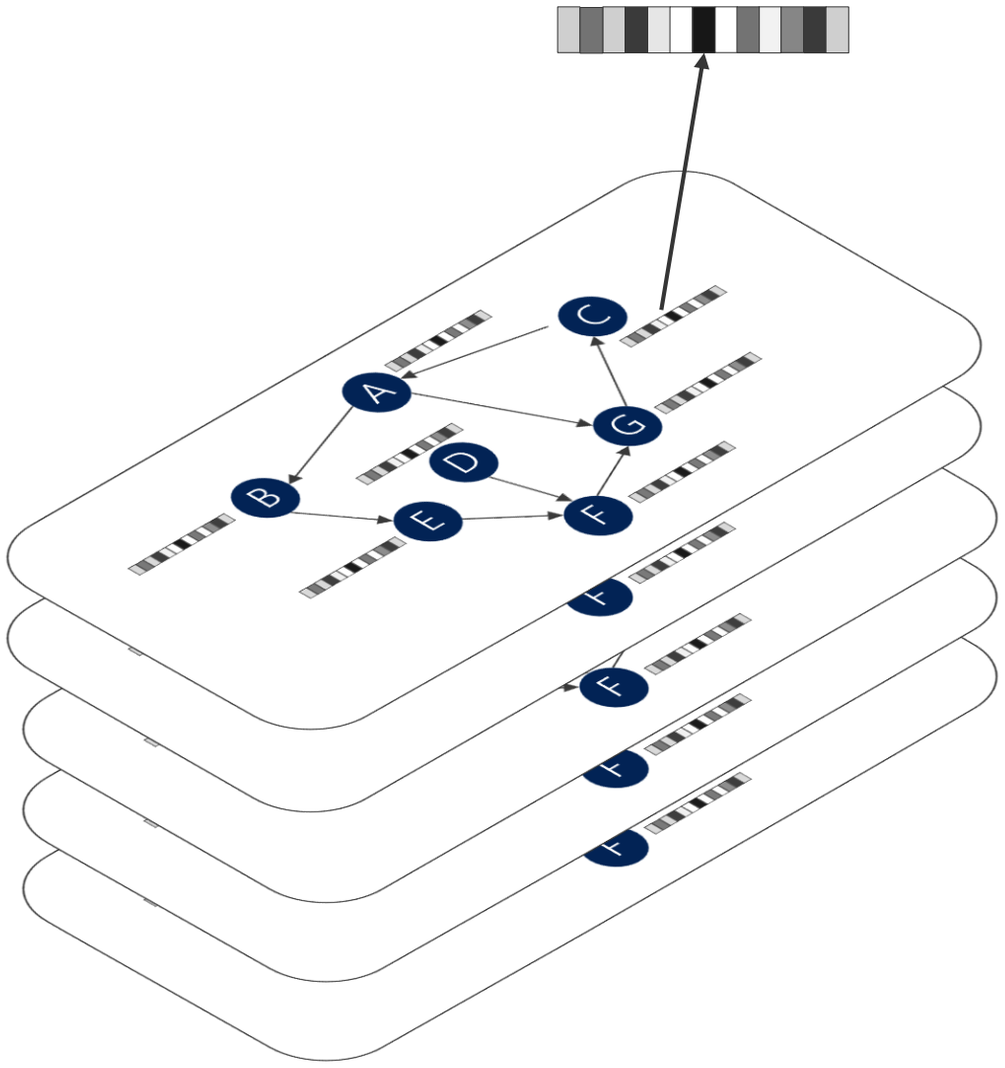
Li et al (2015). Gated graph sequence neural networks.

Graph Neural Networks: Unrolling



Li et al (2015). Gated graph sequence neural networks.

Graph Neural Networks: Unrolling



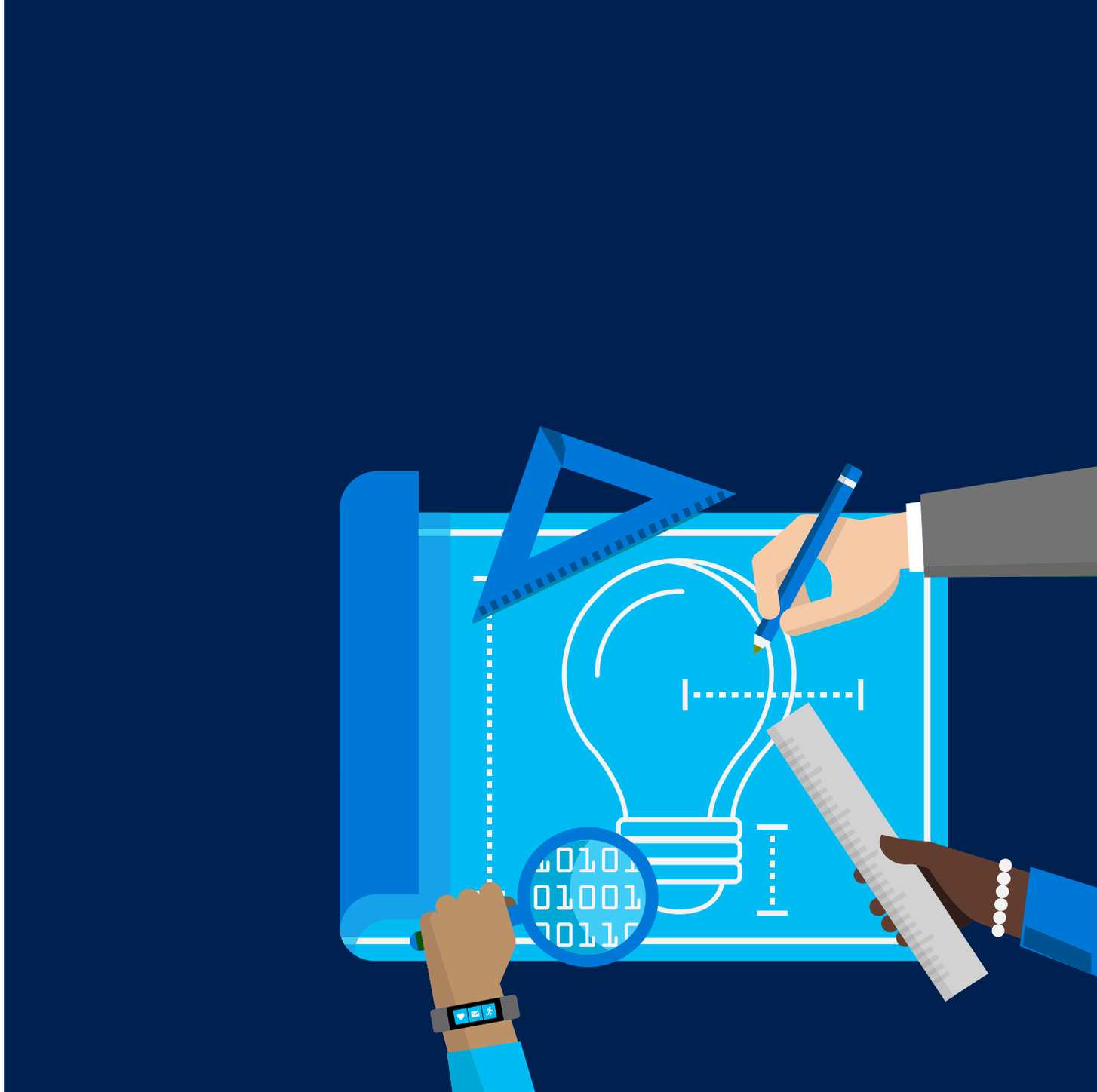
- node selection
- node classification
- graph classification

Li et al (2015). Gated graph sequence neural networks.

Microsoft

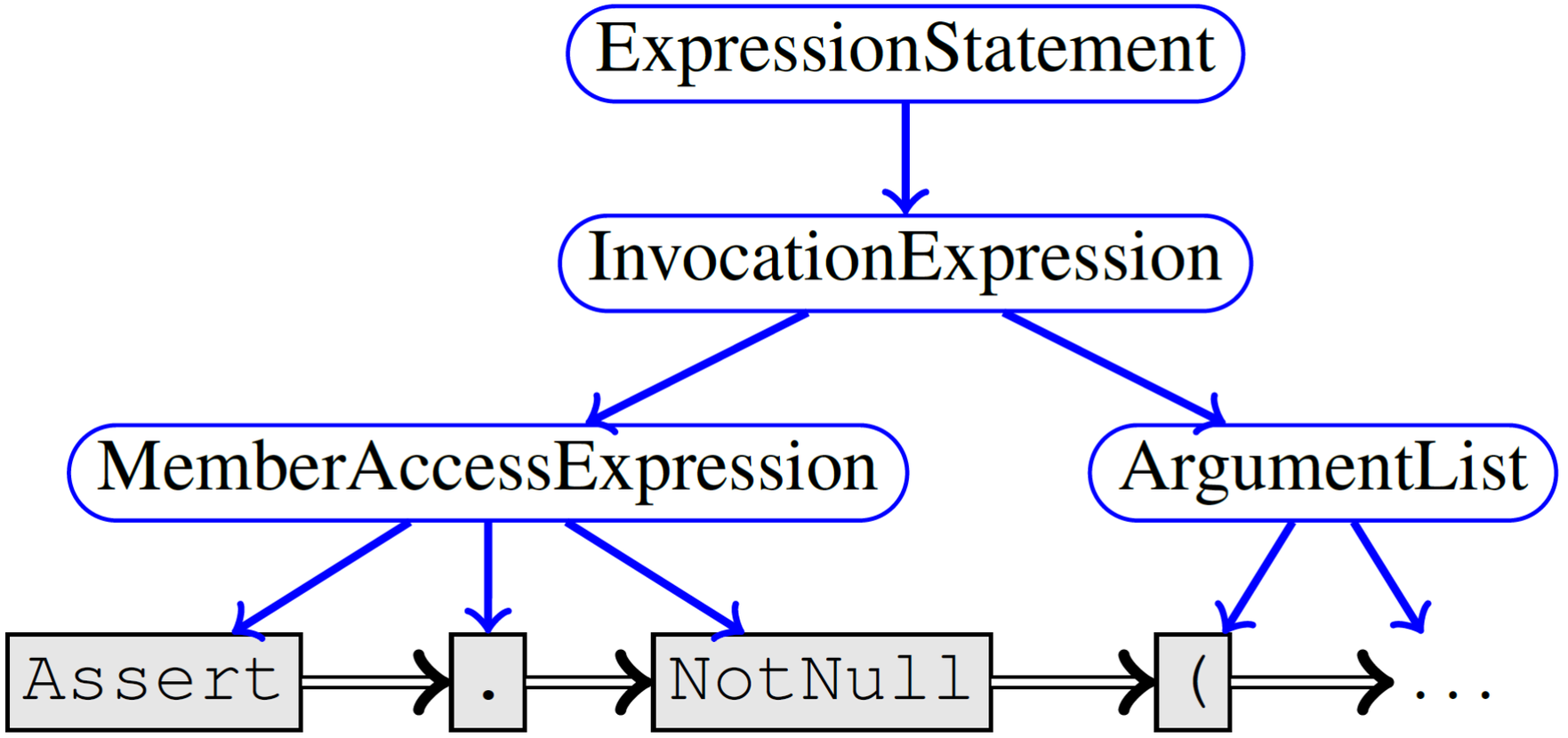
Detecting Variable Misuses

with Graph Neural Networks

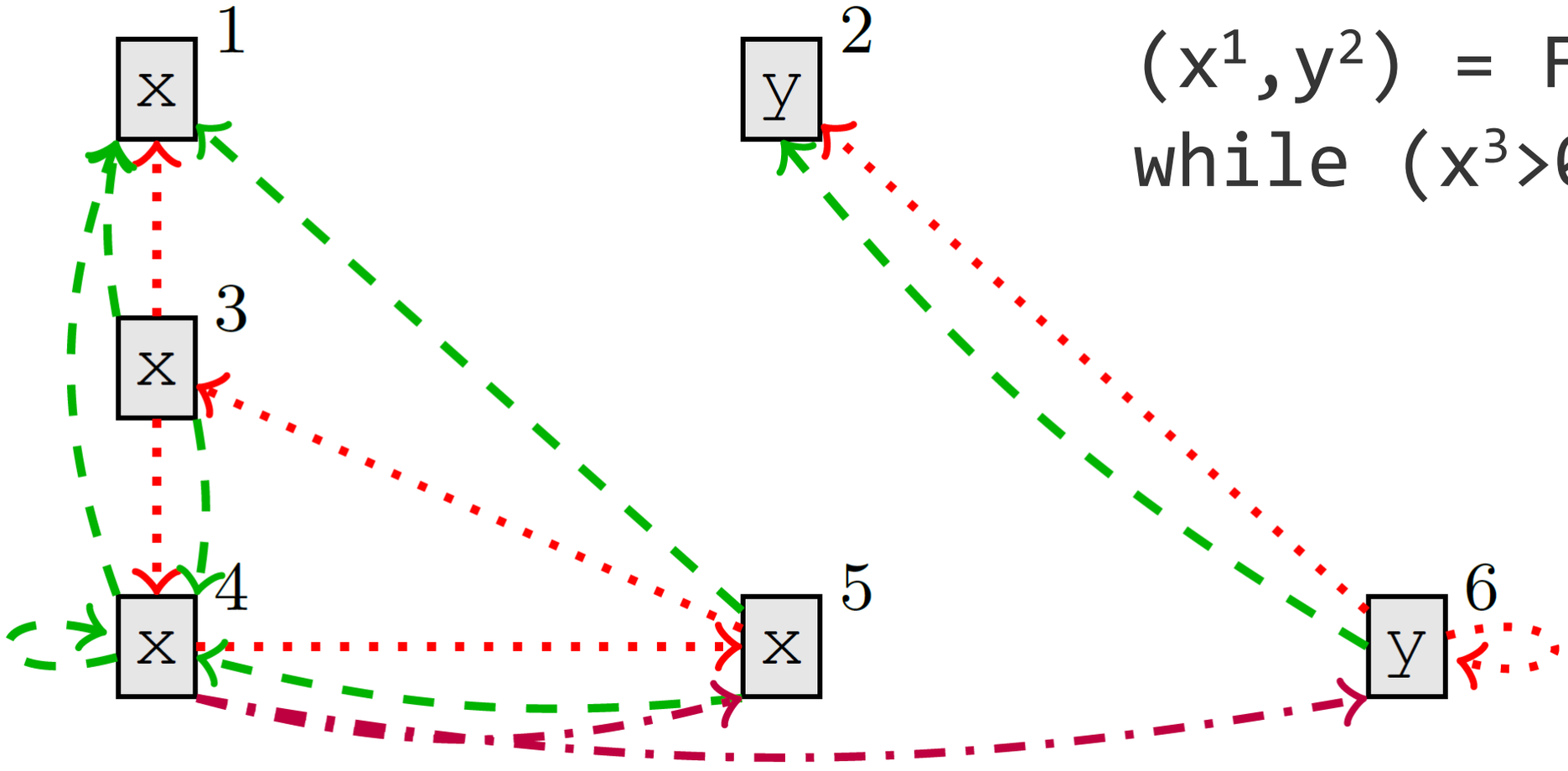


https://arxiv.org/abs/1711.00740

Representing Program Structure as a Graph



Representing Program Structure as a Graph



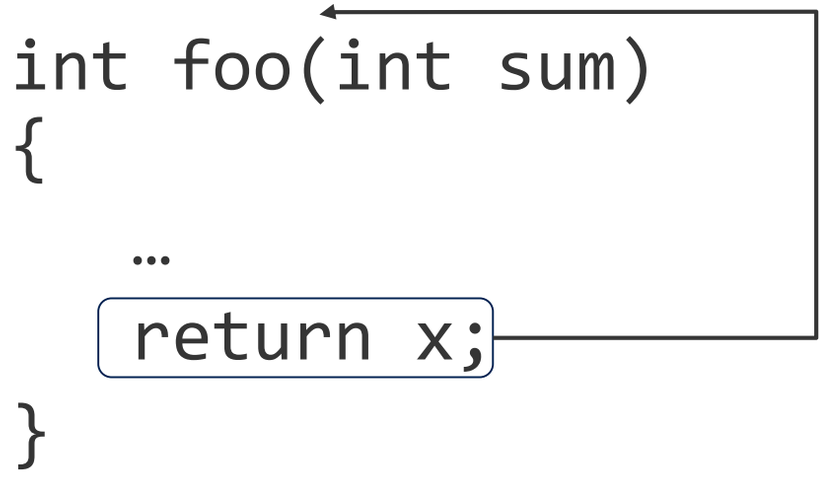

```
(x¹,y²) = Foo();
while (x³>0) x⁴=x⁵+y⁶
```

LastWrite
LastUse
ComputedFrom

Representing Program Structure as a Graph

Additional Edge Types:

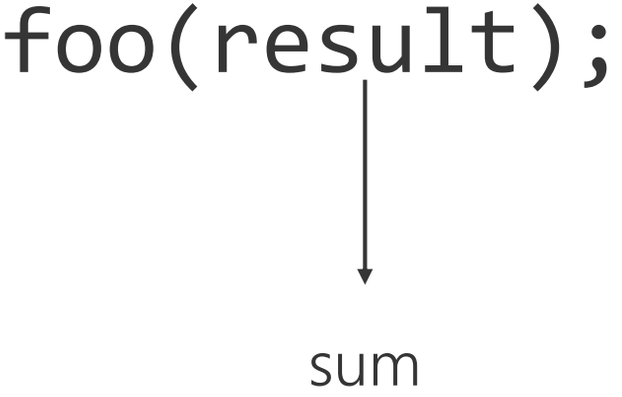
- LastLexicalUse
- ReturnsTo
- FormalArgName



Representing Program Structure as a Graph

Additional Edge Types:

- LastLexicalUse
- ReturnsTo
- FormalArgName



Graph Representation for Variable Misuse

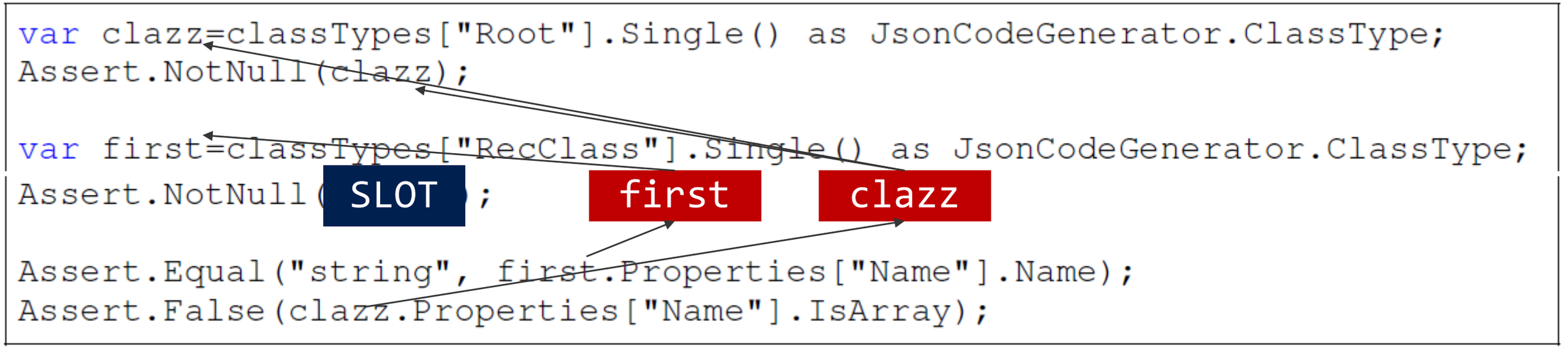
```
var clazz=classTypes["Root"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(clazz);

var first=classTypes["RecClass"].Single() as JsonCodeGenerator.ClassType;
Assert.NotNull(        );

Assert.Equal("string", first.Properties["Name"].Name);
Assert.False(clazz.Properties["Name"].IsArray);
```

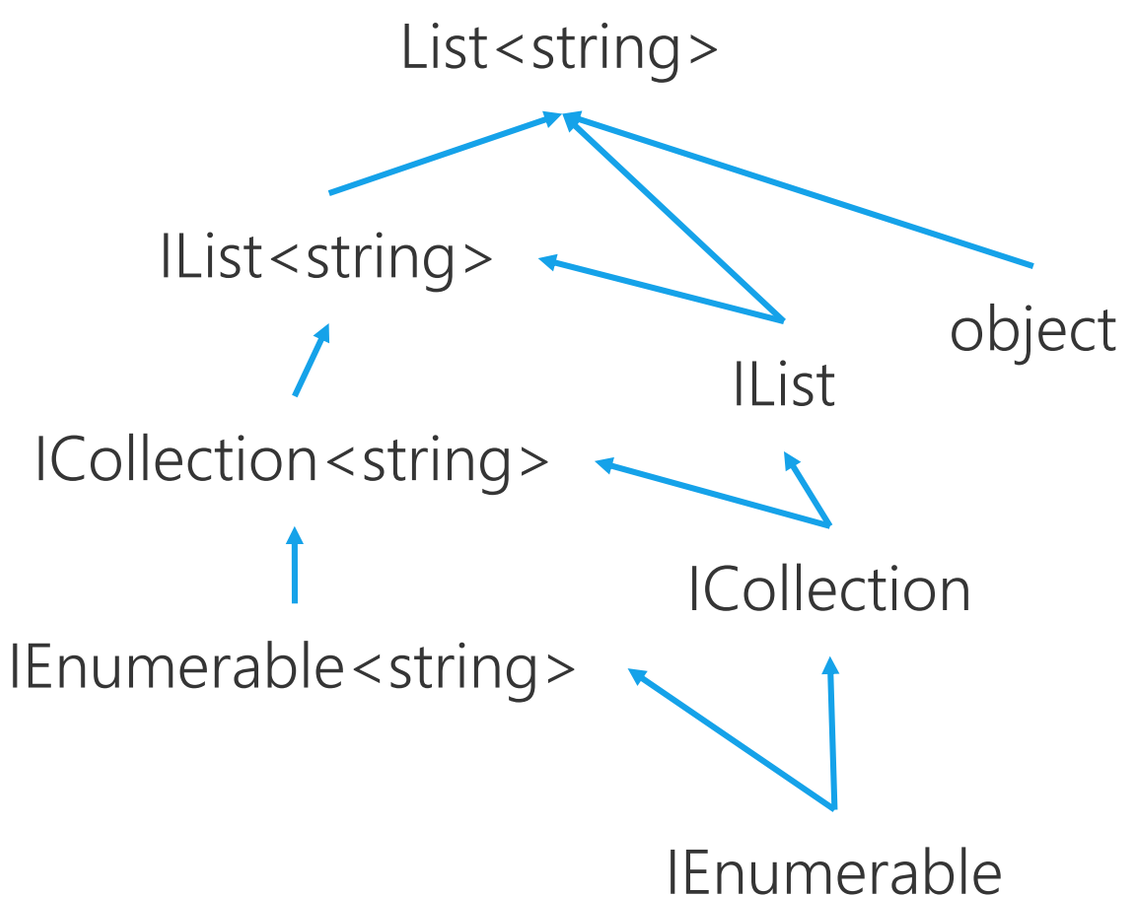
Possible type-correct options: **`clazz, first`**

Graph Representation for Variable Misuse



Goal: make the representation of `SLOT` as close as possible to the representation of the correct candidate node

Representing Variable Type Information



$$\tau^*(v) = \{\tau_{List<string>}, \tau_{IList}, \tau_{object}, \dots\}$$

$$\boldsymbol{r}_{List<string>} \quad \boldsymbol{r}_{IList} \quad \boldsymbol{r}_{object}$$

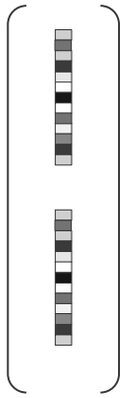
Elementwise Max

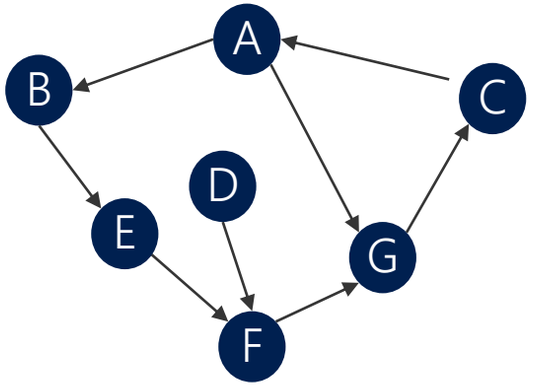
$$\boldsymbol{r}_{\tau^*(v)}$$

Representing Nodes

`classTypes` → `class`  `, types` 

Avg

$\boldsymbol{r}_{\tau^*(v)}$ 



Implementation

- Sparse TensorFlow implementation
- 16 edge types (forward + backward)
- ~900 nodes/graph ~8k edges/graph

Stats

- 55 graphs/s during training
- 219 graphs/s during testing

Dataset

2.9MLOC

GitHub

Name	Git SHA	kLOCs	Slots	Vars	Description
Akka.NET	719335a1	240	51.3k	51.2k	Actor-based Concurrent & Distributed Framework
AutoMapper	2ca7c2b5	46	3.7k	10.7k	Object-to-Object Mapping Library
BenchmarkDotNet	1670ca34	28	5.1k	6.1k	Benchmarking Library
BotBuilder	190117c3	44	6.4k	8.7k	SDK for Building Bots
choco	93985688	36	3.8k	5.2k	Windows Package Manager
commandline†	09677b16	11	1.1k	2.3k	Command Line Parser
CommonMark.NET^{Dev}	f3d54530	14	2.6k	1.4k	Markdown Parser
Dapper	931c700d	18	3.3k	4.7k	Object Mapper Library
EntityFramework	fa0b7ec8	263	33.4k	39.3k	Object-Relational Mapper
Hangfire	ffc4912f	33	3.6k	6.1k	Background Job Processing Library
Humanizer†	cc11a77e	27	2.4k	4.4k	String Manipulation and Formatting
Lean†	f574bfd7	190	26.4k	28.3k	Algorithmic Trading Engine
Nancy	72e1f614	70	7.5k	15.7	HTTP Service Framework
Newtonsoft.Json	6057d9b8	123	14.9k	16.1k	JSON Library
Ninject	7006297f	13	0.7k	2.1k	Code Injection Library
NLog	643e326a	75	8.3k	11.0k	Logging Library
Opserver	51b032e7	24	3.7k	4.5k	Monitoring System
OptiKey	7d35c718	34	6.1k	3.9k	Assistive On-Screen Keyboard
orleans	e0d6a150	300	30.7k	35.6k	Distributed Virtual Actor Model
Polly	0afdbc32	32	3.8k	9.1k	Resilience & Transient Fault Handling Library
quartznet	b33e6f86	49	9.6k	9.8k	Scheduler
ravendb^{Dev}	55230922	647	78.0k	82.7k	Document Database
RestSharp	70de357b	20	4.0k	4.5k	REST and HTTP API Client Library
Rx.NET	2d146fe5	180	14.0k	21.9k	Reactive Language Extensions
scriptcs	f3cc8bcb	18	2.7k	4.3k	C# Text Editor
ServiceStack	6d59da75	231	38.0k	46.2k	Web Framework
ShareX	718dd711	125	22.3k	18.1k	Sharing Application
SignalR	fa88089e	53	6.5k	10.5k	Push Notification Framework
Wox	cdaf6272	13	2.0k	2.1k	Application Launcher

Quantitative Results

Accuracy (%)	Local Model	Avg BiRNN	GGNN
Seen Projects	15.8	73.5	**82.1**
Unseen Projects	13.8	59.7	**68.6**

3.8 type-correct alternative variables per slot (median 3, σ= 2.6)

```
bool TryFindGlobalDirectivesFile(string baseDirectory, string fullPath, out string path) {
   baseDirectory = baseDirectory.TrimEnd(Path.DirectorySeparatorChar);
   var directivesDirectory = Path.GetDirectoryName(          )
                                 .TrimEnd(Path.DirectorySeparatorChar);
   while (directivesDirectory != null && directivesDirectory.Length >= baseDirectory.Length) {
      path = Path.Combine(directivesDirectory, GlobalDirectivesFileName);
      if (File.Exists(path)) return true;

      directivesDirectory = Path.GetDirectoryName(directivesDirectory)
                                .TrimEnd(Path.DirectorySeparatorChar);
   }
   path = null;
   return false;
}
```

What the model sees…

```
bool TryFindGlobalDirectivesFile(string baseDirectory, string fullPath, out string path) {
    baseDirectory = baseDirectory.TrimEnd(Path.DirectorySeparatorChar);
    var directivesDirectory = Path.GetDirectoryName(fullPath)
                                  .TrimEnd(Path.DirectorySeparatorChar);
    while (directivesDirectory != null && directivesDirectory.Length >= baseDirectory.Length) {
        path = Path.Combine(directivesDirectory, GlobalDirectivesFileName);
        if (File.Exists(path)) return true;

        directivesDirectory = Path.GetDirectoryName(directivesDirectory)
                                  .TrimEnd(Path.DirectorySeparatorChar);
    }
    path = null;
    return false;
}
```

What the model sees…

Qualitative Results

```
[global::System.Diagnostics.DebuggerNonUserCodeAttribute]
public void MergeFrom(pb::CodedInputStream input) {
  uint tag;
  while ((tag = input.ReadTag()) != 0) {
    switch(tag) {
      default:
        input.SkipLastField();
        break;
      case 10: {
        #1.AddEntriesFrom(input, _repeated_payload_codec);
        break;
      }
    }
  }
}
```

#1 Payload: 66%, payload_: 44%

Qualitative Results

```
public override bool IsDisposed
{
  get
  {
    lock ( #1 )
    {
       return #2 ;
    }
  }
}
```

#1 _gate: 99%, _observers: 1%

#2 _isDisposed: 90%, _isStopped: 8%, HasObservers: 2%

Qualitative Results

```
var response = ResultsFilter(typeof(TResponse), #1, #2, request);
```

#1 httpMethod: 99%, absoluteUrl: 1%, UserName: 0%, UserAgent: 0%

#2 absoluteUrl: 99%, httpMethod: 1%, UserName: 0%, UserAgent: 0%

```
private bool BecomingCommand(object message)
{
    if (ReceiveCommand( #1 ) return true;
    if ( #2 .ToString() == #3 ) #4 .Tell( #5 );
    else return false;
    return true;
}
```

#1	message: 100%, Response: 0%, Message: 0%
#2	message: 100%, Response: 0%, Message: 0%
#3	Response: 91%, Message: 9%
#4	Probe: 98%, AskedForDelete: 2%
#5	Response: 98%, Message: 2%

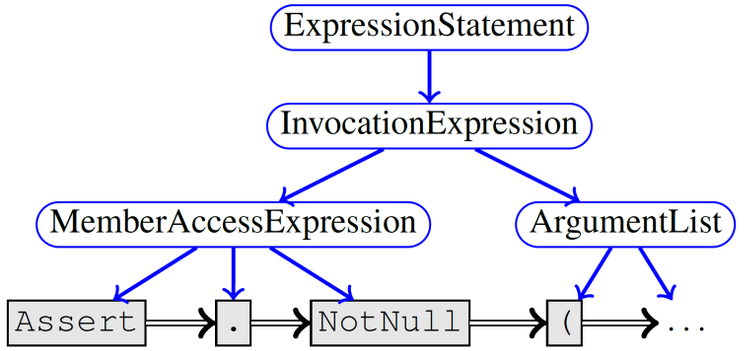
Detecting Real-Life Bugs

```
public ArraySegment<byte> ReadBytes(int length){
    int size = Math.Min(length, _len - _pos);
    var buffer = EnsureTempBuffer( length );
    var used = Read(buffer, 0, size);
    return new ArraySegment<byte>(buffer, 0, used);
}
```

Detecting Real-Life Bugs

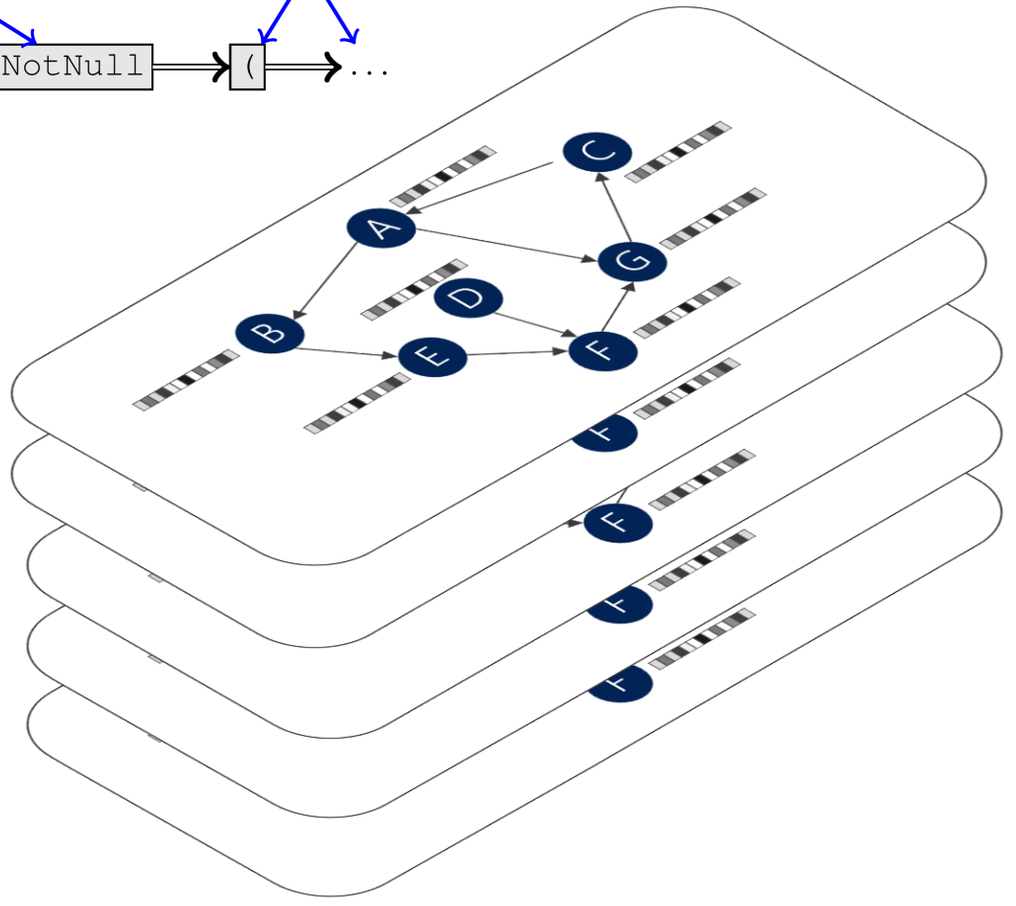
```
protected void ValidateRestorePreconditions(string backupFilename) {
  if (IsValidBackup(backupFilename) == false) {
    output("Error:"+ backupLocation +" doesn't look like a valid backup");
    output("Error: Restore Canceled");
    throw new InvalidOperationException(
        backupLocation + " doesn't look like a valid backup");
  }
  ...
```

Closing Thoughts



Exploit Rich Structure of Code

Graph Neural Networks Scale



A survey: https://ml4code.github.io [Please, contribute]