

A Structural Model for Contextual Code Changes

OOPSLA'2020



Shaked Brody
Technion



Uri Alon
Technion



Eran Yahav
Technion

Edit Completion

```
bool GetFileCharacteristics(out FileCharacteristics fileCharacteristics)
{
    fileCharacteristics = new FileCharacteristics(this.OpenTime, this.currentFileLength);
    return true;
}
```

Edit Completion

```
bool FileCharacteristics GetFileCharacteristics(out FileCharacteristics fileCharacteristics)
{
    fileCharacteristics = new FileCharacteristics(this.OpenTime, this.currentFileLength);
    return true;
}
```

Edit Completion

```
bool FileCharacteristics GetFileCharacteristics(out FileCharacteristics fileCharacteristics)  
{  
    fileCharacteristics = new FileCharacteristics(this.OpenTime, this.currentFileLength);  
    return true;  
}
```

Edit Completion

```
bool FileCharacteristics GetFileCharacteristics(out FileCharacteristics fileCharacteristics)  
{  
  fileCharacteristics = new FileCharacteristics(this.OpenTime, this.currentFileLength);  
  return true;  
}
```



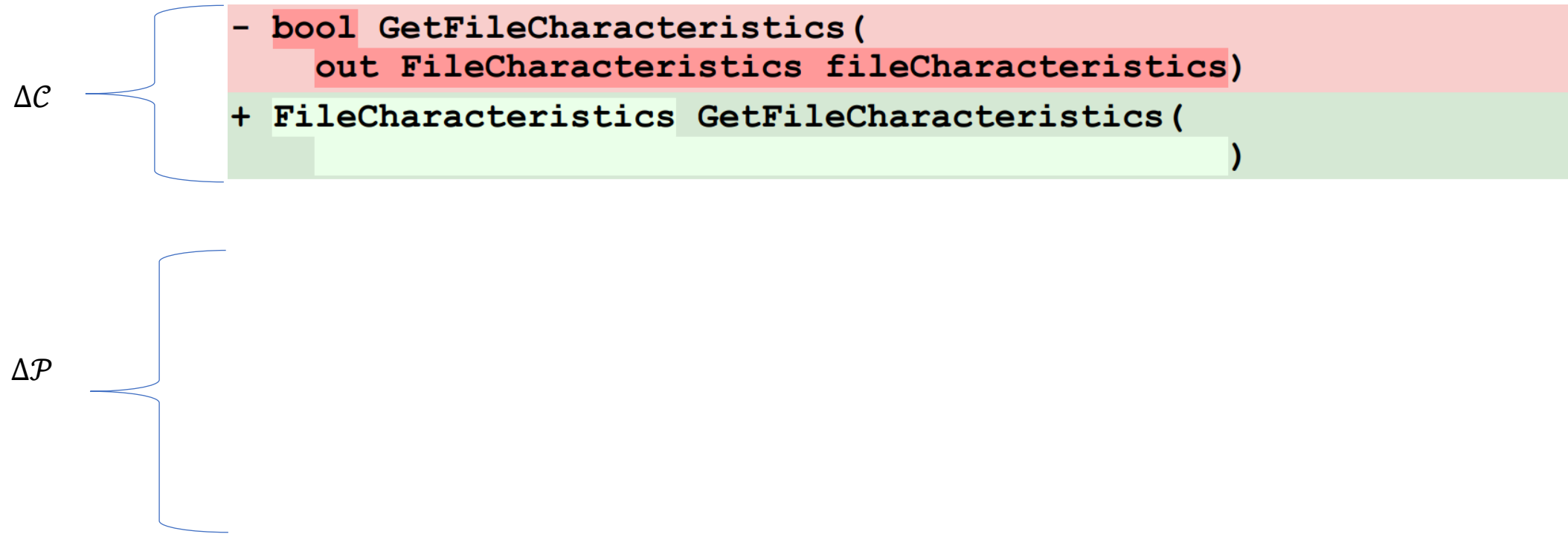
```
return new FileCharacteristics(this.OpenTime, this.currentFileLength);
```

Edit Completion

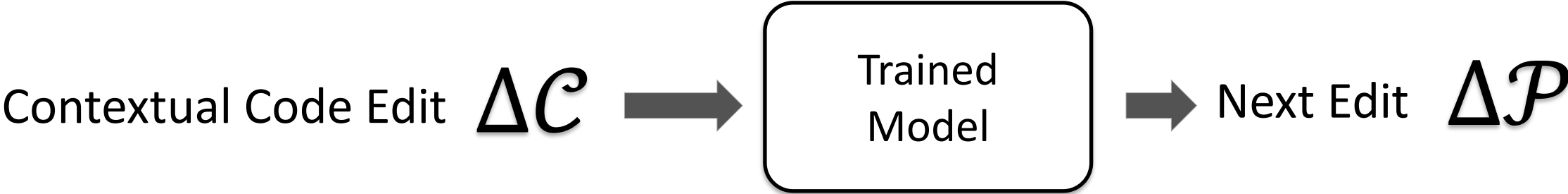
```
bool FileCharacteristics GetFileCharacteristics(out FileCharacteristics fileCharacteristics)  
{  
    fileCharacteristics = return new FileCharacteristics(this.OpenTime, this.currentFileLength);  
    return true;  
}
```

Edit Completion

Given a contextual code edit - $\Delta\mathcal{C}$, predict the next likely edit - $\Delta\mathcal{P}$



Edit Completion



Training data
(thousands of examples): { $(\Delta\mathcal{C}_1, \Delta\mathcal{P}_1)$
 $(\Delta\mathcal{C}_2, \Delta\mathcal{P}_2)$
... } Test data: $(\Delta\mathcal{C}', \Delta\mathcal{P}')$

Challenges

- How to represent code?
 - Textual representation
 - Structural representation
- How to represent code edits?
 - Code likelihood
 - Edit likelihood

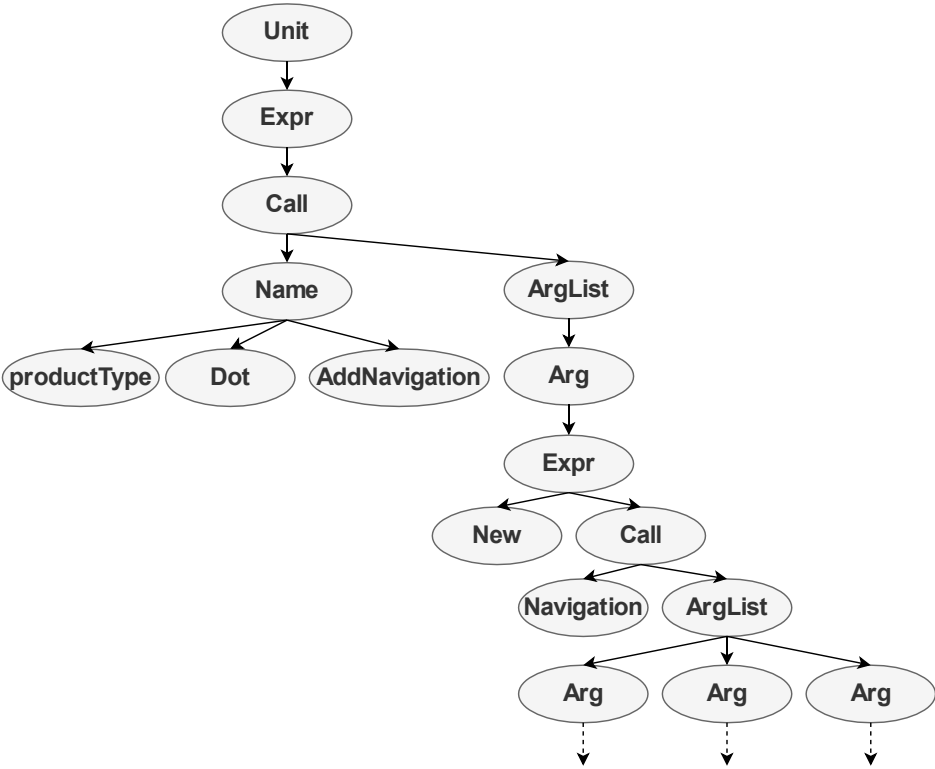
Representing Code

Textual Representation

```
productType.AddNavigation(  
    new Navigation(  
        featuredProductFk,  
        "FeaturedProductCategory",  
        pointsToPrincipal: false));
```

productType	.	AddNavidation	(new	Navigation	(...
-------------	---	---------------	---	-----	------------	---	-----

Structural Representation



Representing Code Edits

- Code Likelihood $P(\mathcal{Y}|\mathcal{X})$

```
productType.AddNavigation(  
    new Navigation(  
        featuredProductFk,  
        "FeaturedProductCategory",  
        pointsToPrincipal: false));
```



```
productType.AddNavigation(  
    "FeaturedProductCategory",  
    featuredProductFk,  
    pointsToPrincipal: false);
```


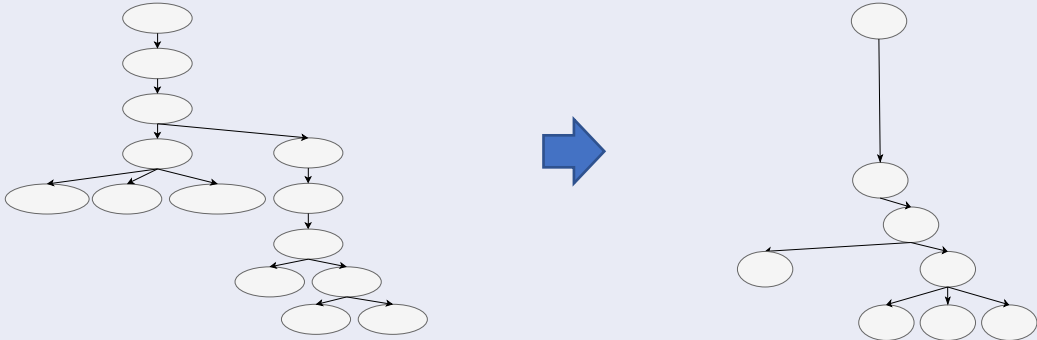

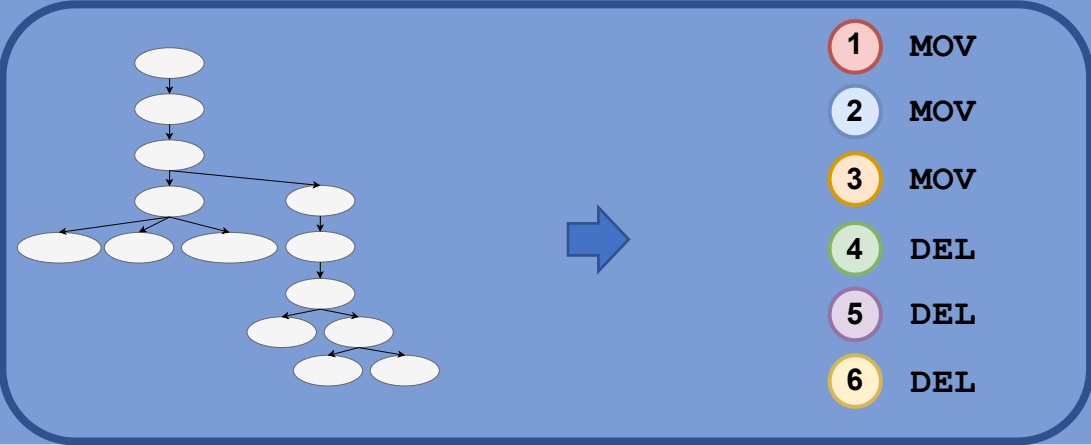
- Edit Likelihood $P(\Delta_{\mathcal{X} \rightarrow \mathcal{Y}}|\mathcal{X})$

```
productType.AddNavigation(  
    new Navigation(  
        featuredProductFk,  
        "FeaturedProductCategory",  
        pointsToPrincipal: false));
```

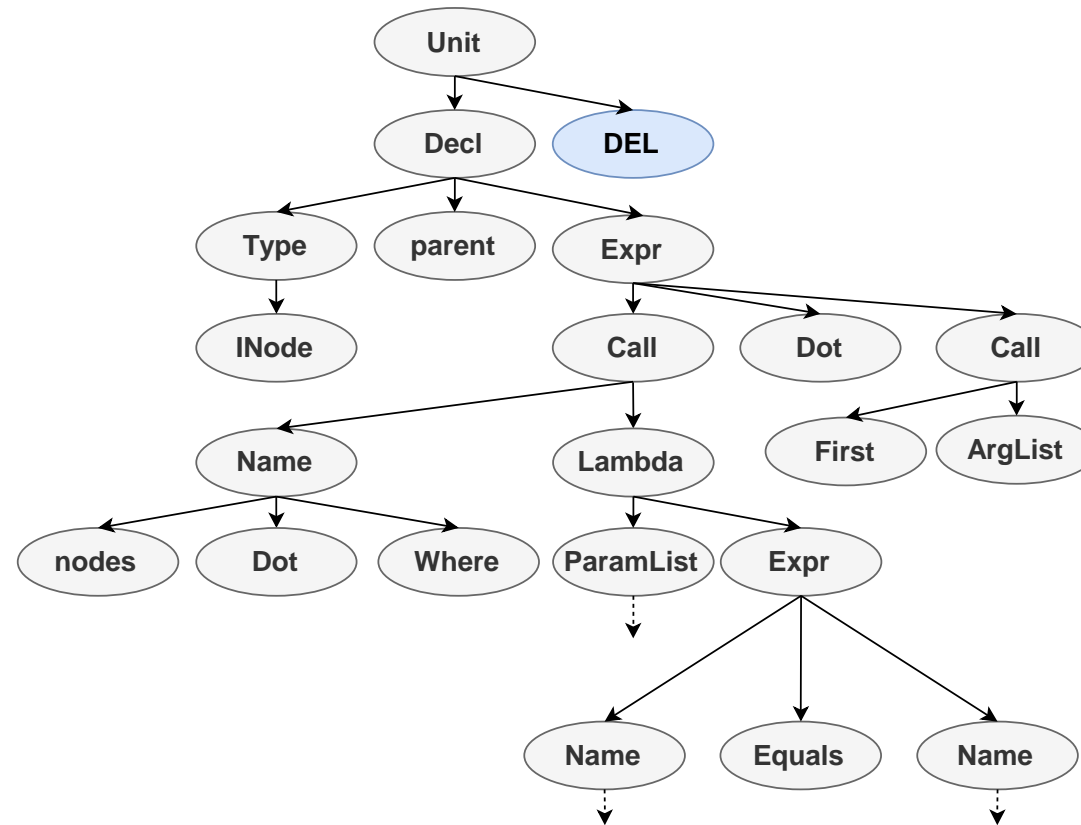


```
1) DELETE new Navigation()  
2) SWITCH Arg1, Arg2
```

Code and Edit Representations

Code Representation / Edit Representation	Textual	Structural
Code Likelihood	 <p>(Chen et al. 2019)</p>	 <p>(Aharoni and Goldberg 2017)</p>
Edit Likelihood	 <p>(Malmi et al. 2019)</p>	 <ul style="list-style-type: none"> 1 MOV 2 MOV 3 MOV 4 DEL 5 DEL 6 DEL

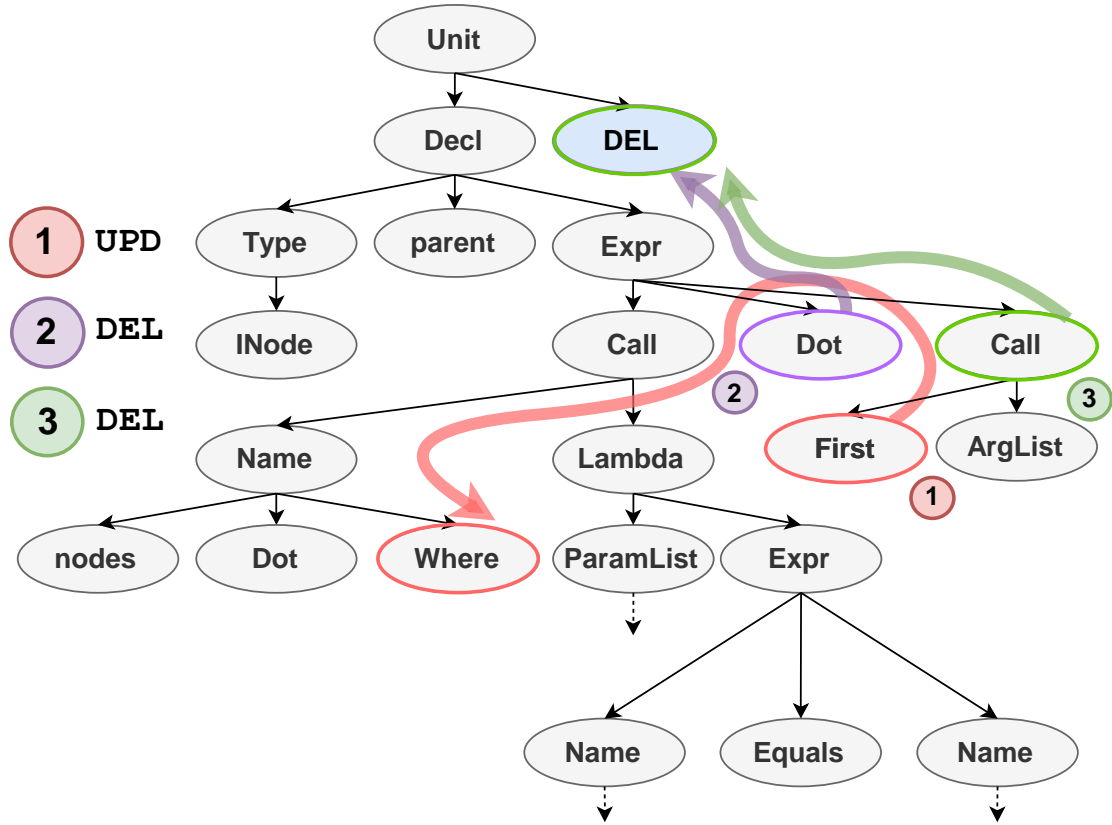
Edit Operations with AST Paths



```
- INode parent = nodes.Where(  
    x => x.Type == NodeType.Root).First();
```

```
+ INode parent = nodes.First(  
    x => x.Type == NodeType.Root) _____;
```

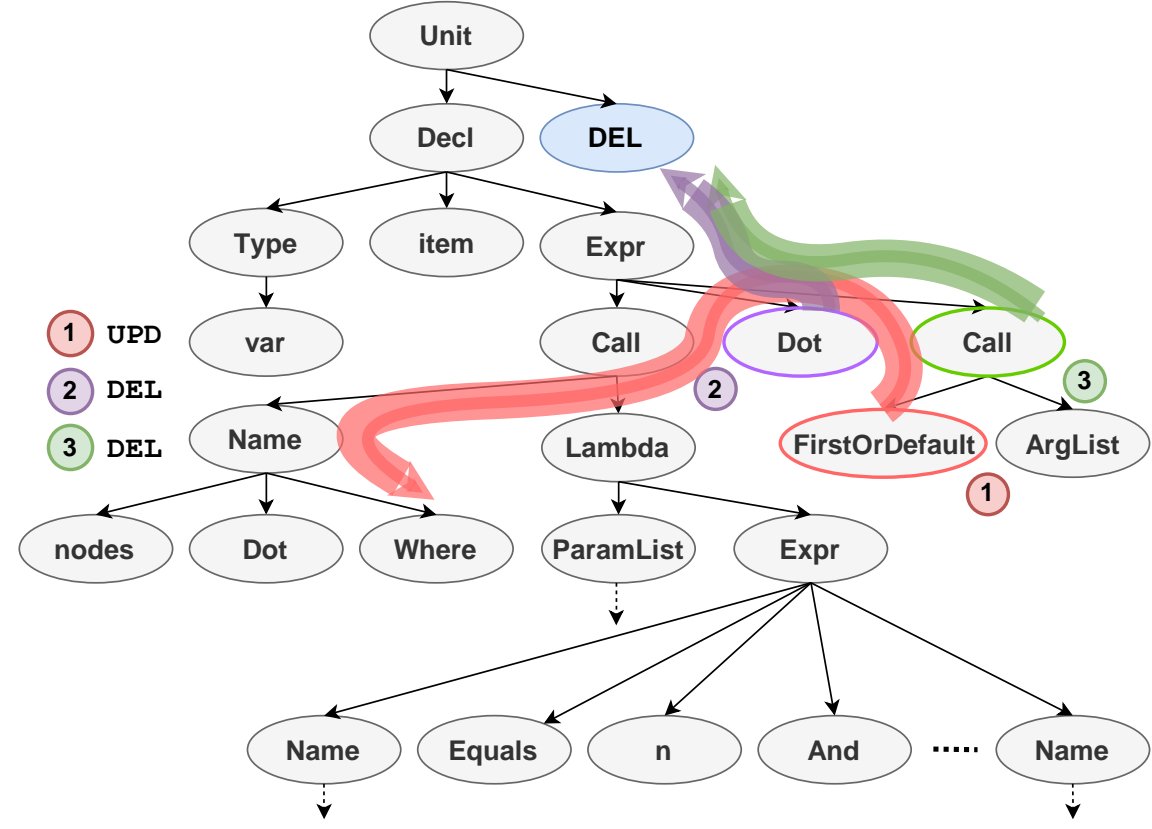
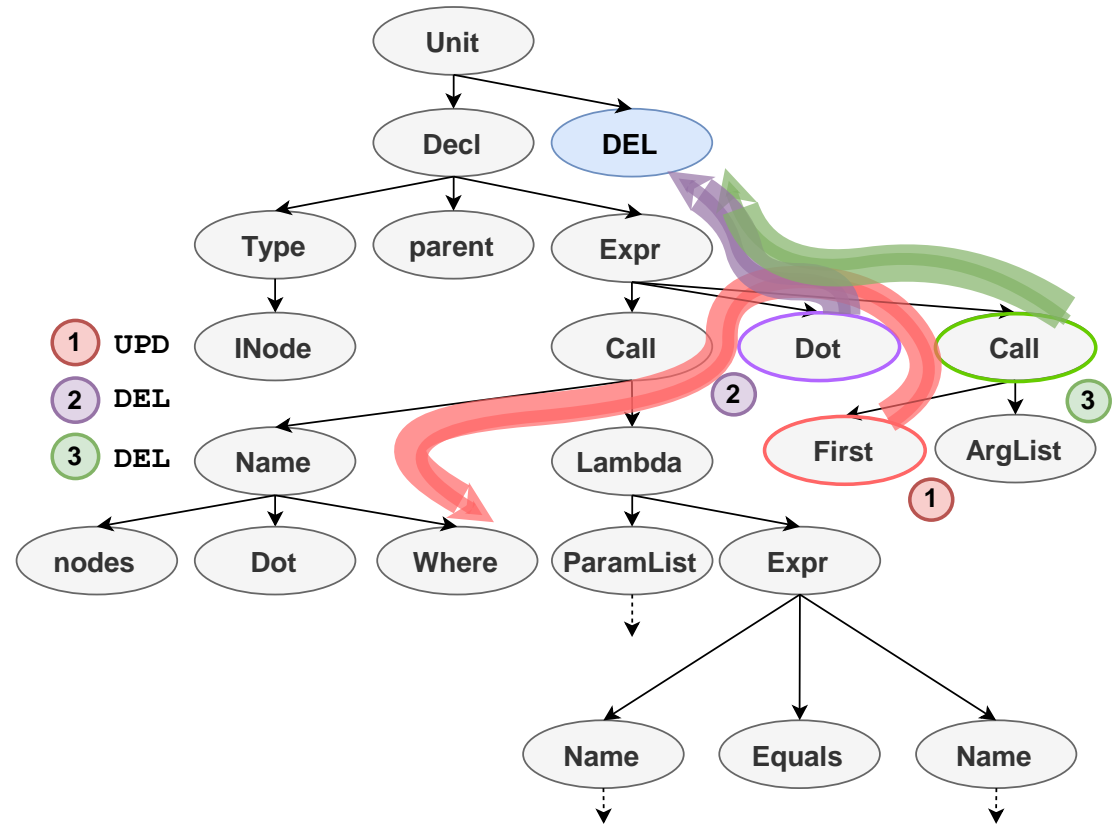
Edit Operations with AST Paths



```

- INode parent = nodes.Where(
    x => x.Type == NodeType.Root).First();
+ INode parent = nodes.First(
    x => x.Type == NodeType.Root) ;
  
```

Why Structural and Edit Likelihood?



```

- INode parent = nodes.Where(
    x => x.Type == NodeType.Root).First();

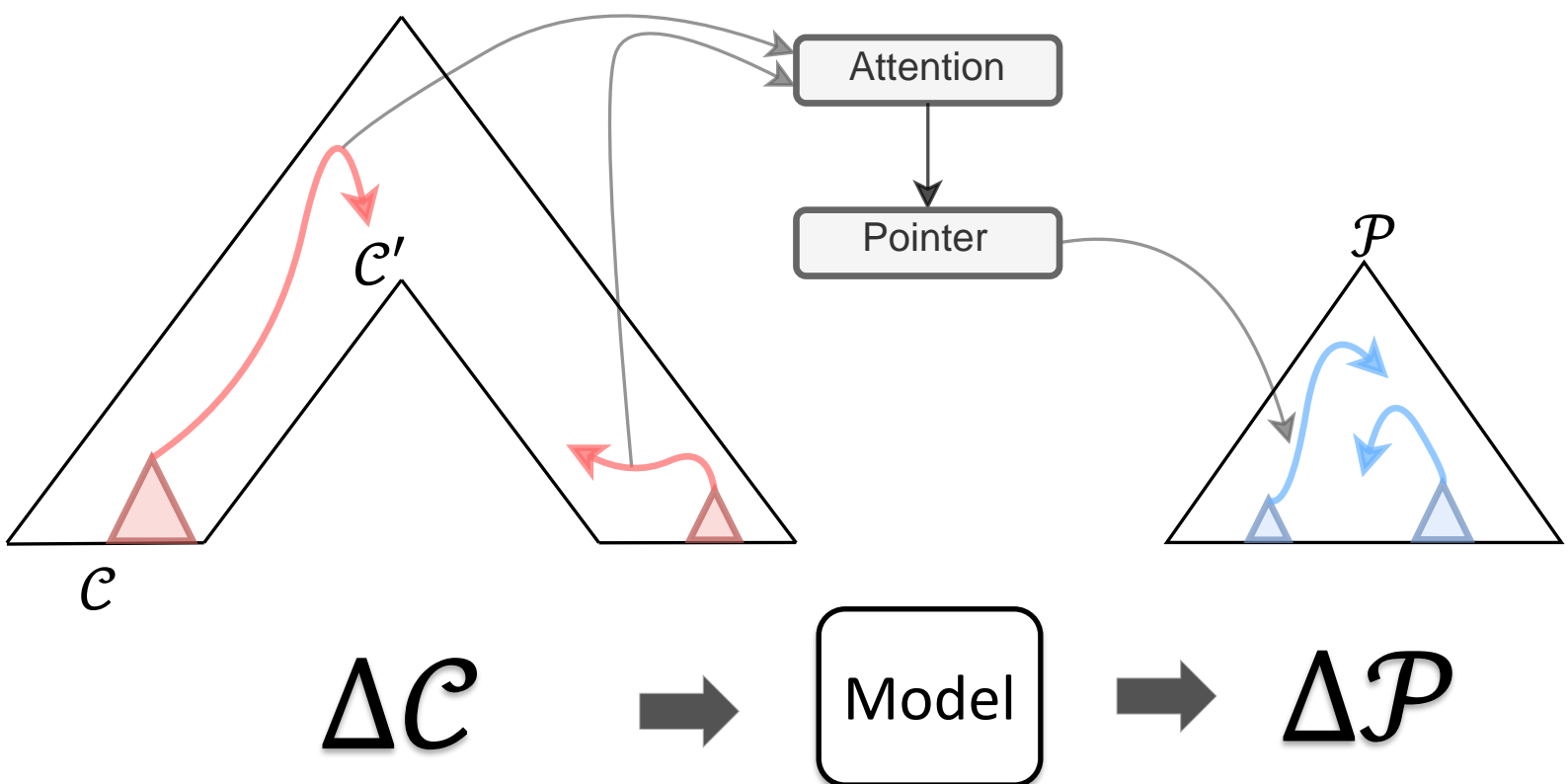
+ INode parent = nodes.First(
    x => x.Type == NodeType.Root) ;
    
```

```

- var item = nodes.Where(
    x => x.name == n &&
    x.Type == NodeType.Directory &&
    x.ParentId == parent.Id).FirstOrDefault();

+ var item = nodes.FirstOrDefault(
    x => x.name == n &&
    x.Type == NodeType.Directory &&
    x.ParentId == parent.Id) ;
    
```

C^3PO - Contextual Code Changes via Path Operations



- 750K learnable parameters
- 9 hours of training

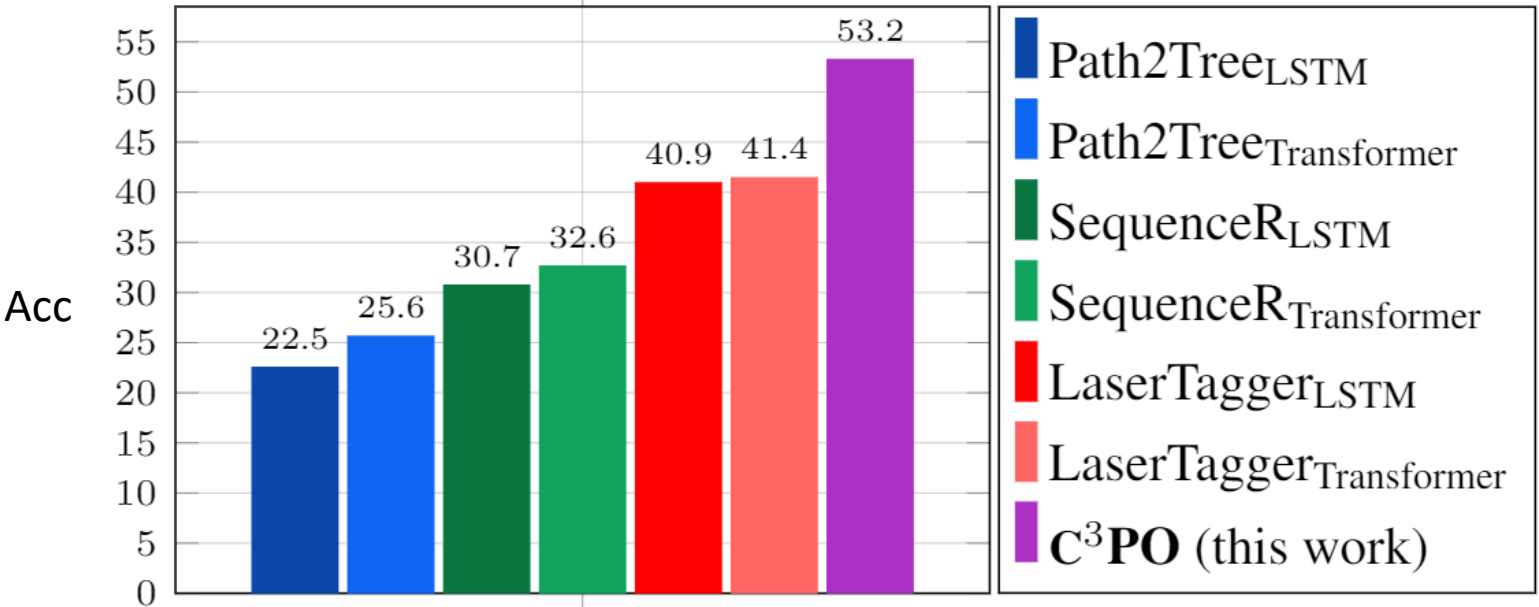
Data

- We scraped the 53 most popular C# repositories from GitHub
 - 39.5K train
 - 4.4K validation
 - 5.9K test
- <https://github.com/tech-srl/c3po/>

Baselines

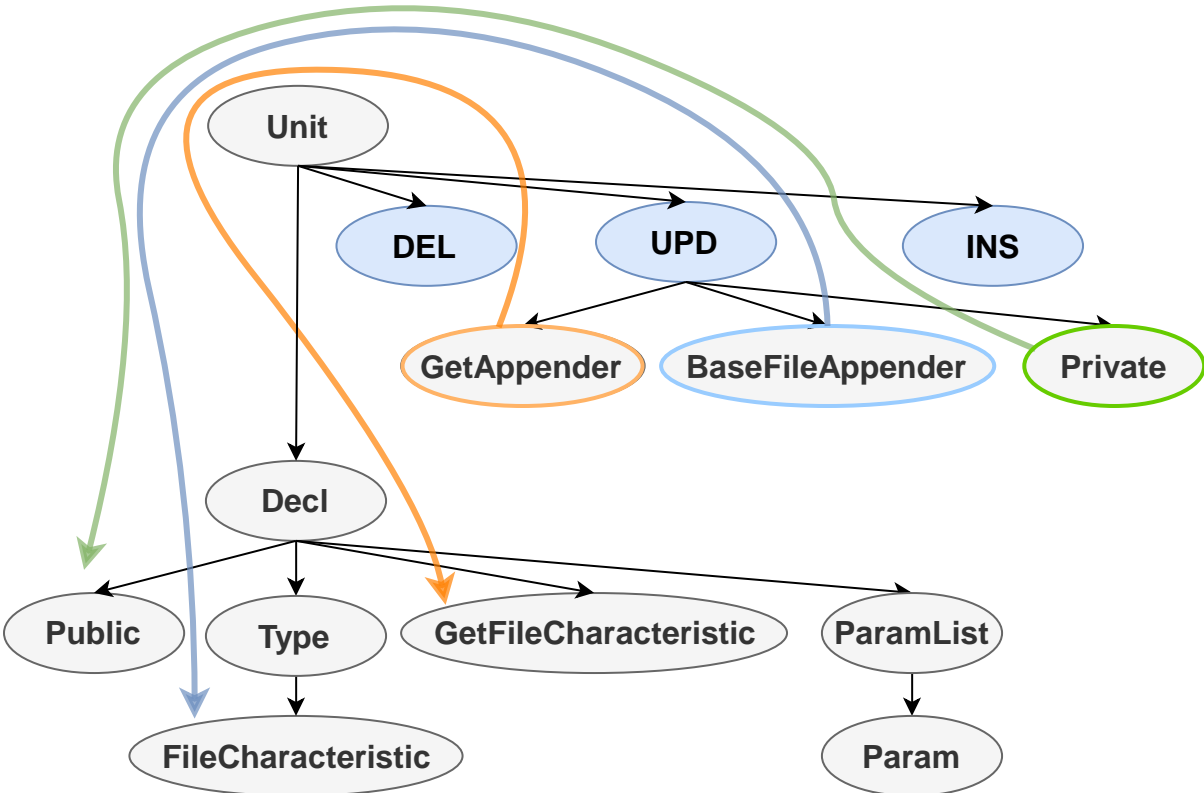
Code Representation Edit Representation	Textual	Structural
Code Likelihood	SequnceR (Chen et al. 2019)	Path2Tree (Aharoni and Goldberg 2017)
Edit Likelihood	LaserTagger (Malmi et al. 2019)	<i>C³PO</i> (this work)

Results



	Textual	Structural
Code Likelihood	<i>SequenceR_{LSTM}</i>	<i>Path2Tree_{LSTM}</i>
	<i>SequenceR_{Transformer}</i>	<i>Path2Tree_{Transformer}</i>
Edit Likelihood	<i>LaserTagger_{LSTM}</i>	C³PO (this work)
	<i>LaserTagger_{Transformer}</i>	

Examples



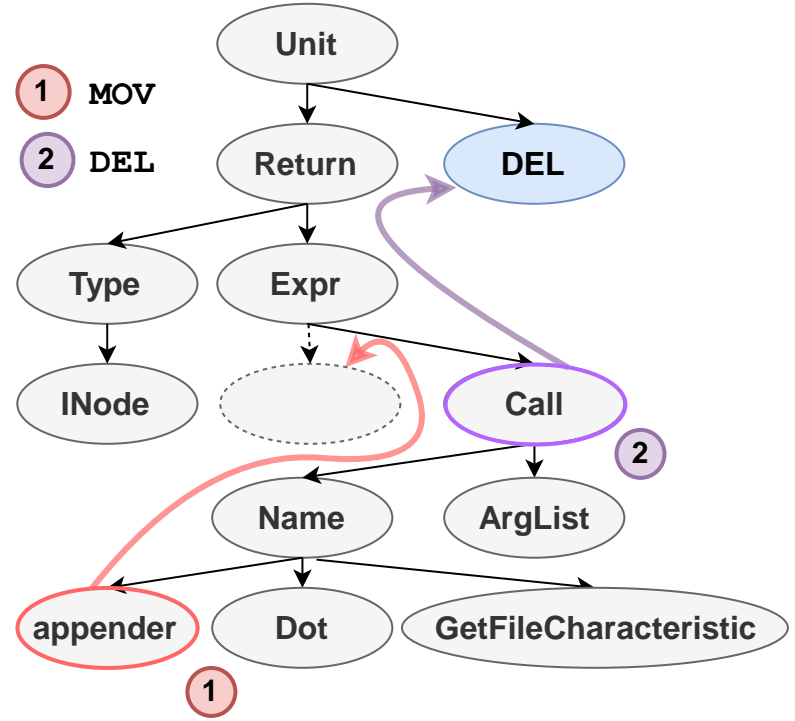
Context Edits ($\mathcal{C} \rightarrow \mathcal{C}'$)

```
public FileCharacteristic GetFileCharacteristic(string fileName)
```



```
private BaseFileAppender GetAppender(string fileName)
```

(Given)



Program Edits ($\mathcal{P} \rightarrow \mathcal{P}'$)

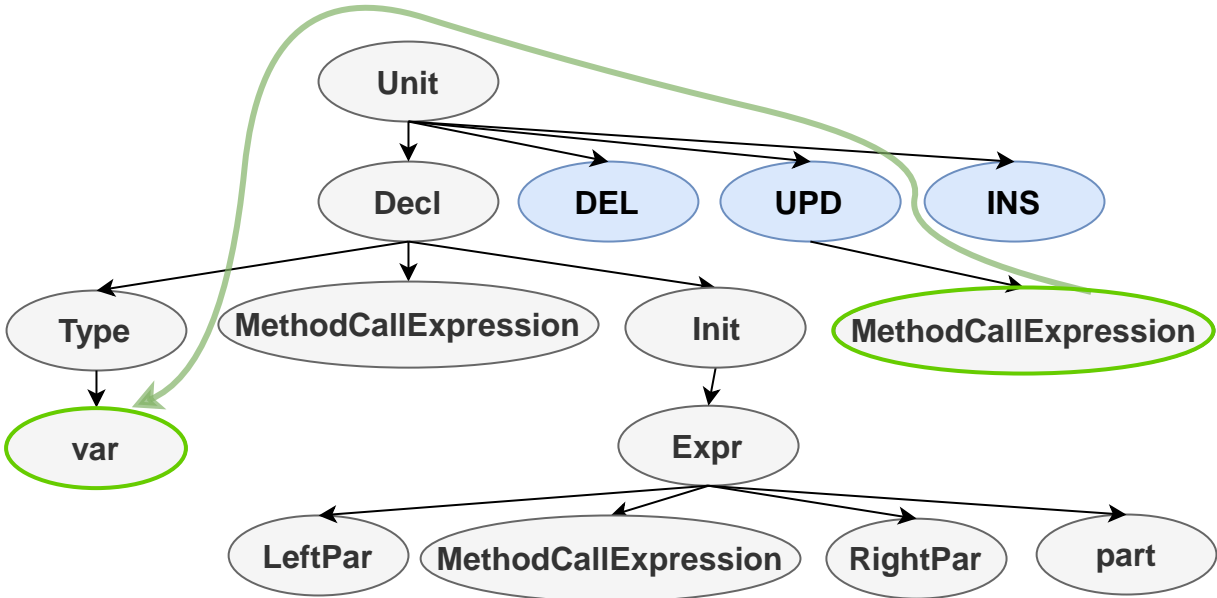
```
return appender.GetFileCharacteristic();
```



```
return appender;
```

(Predicted)

Examples

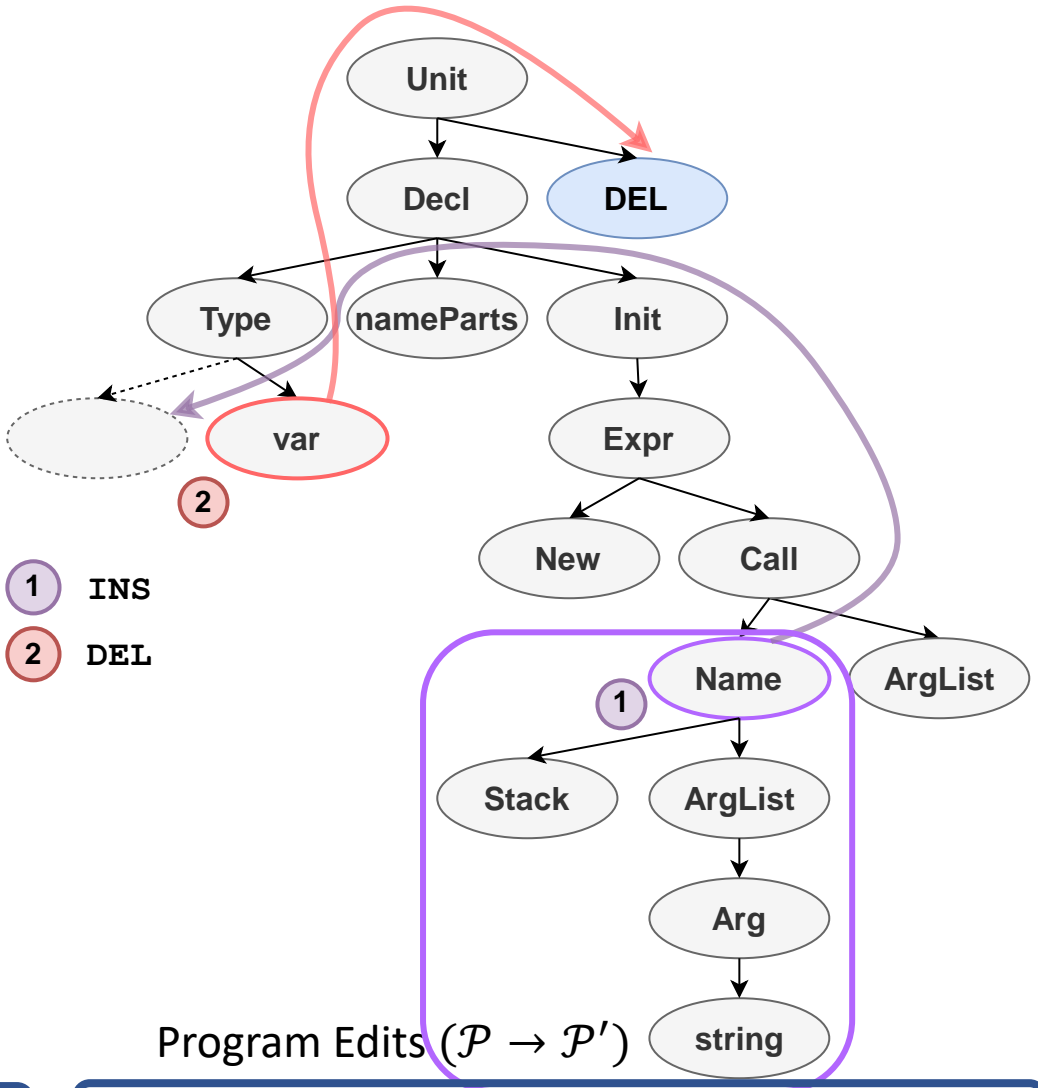


Context Edits ($\mathcal{C} \rightarrow \mathcal{C}'$)

```
var methodExpression = (MethodCallExpression)part;
```

```
MethodCallExpression methodExpression = (MethodCallExpression)part;
```

(Given)



- 1 INS
- 2 DEL

Program Edits ($\mathcal{P} \rightarrow \mathcal{P}'$)

```
var nameParts = new Stack<string>();
```

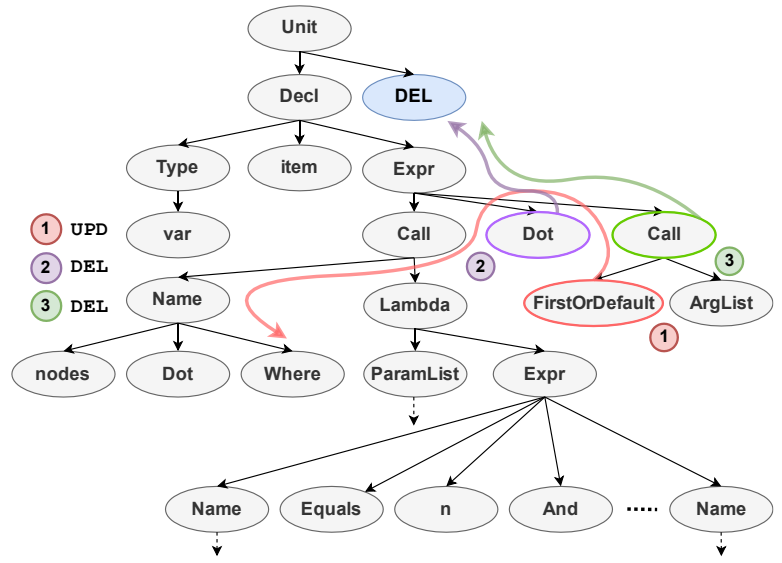
```
Stack<string> nameParts = new Stack<string>();
```

(Predicted)

Summary

- Representations
 - Edit likelihood
 - Structural code representation
- Edit Completion task
- Novel neural architecture
 - AST paths as edits operations
- <https://github.com/tech-srl/c3po/>

shakedbr@cs.technion.ac.il



```
bool FileCharacteristics GetFileCharacteristics(out FileCharacteristics fileCharacteristics)
{
    fileCharacteristics = new FileCharacteristics(this.OpenTime, this.currentFileLength);
    return true;
}

return new FileCharacteristics(this.OpenTime, this.currentFileLength);
```

