

Testing and Debugging

Project 2:

Mutation Testing

Project I Results

- Results are in your svn repository: *results/results.csv*
- Output of test scripts: *test-output/*
- Sample solution and secret tests:
<https://prog2.cs.uni-saarland.de/debugging/students/project1/0/>
- TarantulaTest was modified and run again - *results.csv* is up to date.

Mutation Transformer

- Insert mutations to the source code and record information.
- Manipulate AST by inserting conditions that switch between the original and mutated version of a statement.

Replacing Nodes

- `node.getLocationInParent()` gives a location to insert.

`getLocationInParent`

```
public final StructuralPropertyDescriptor getLocationInParent()
```

Returns the location of this node within its parent, or `null` if this is a root node.

```
ASTNode node = ...;
ASTNode parent = node.getParent();
StructuralPropertyDescriptor location = node.getLocationInParent();
assert (parent != null) == (location != null);
if ((location != null) && location.isChildProperty())
    assert parent.getStructuralProperty(location) == node;
if ((location != null) && location.isChildListProperty())
    assert ((List) parent.getStructuralProperty(location)).contains(node);
```

Note that the relationship between an AST node and its parent node may change over the lifetime of a node.

Returns:

the location of this node in its parent, or `null` if this node has no parent

Since:

3.0

- Either use `parent.setStructuralProperty()` or manipulate list.

Mutation Runner

- First check if the tests pass on unmutated version.
- Read information about previously applied mutations.
- For all mutations: enable mutation - run all tests - record result (detected/not detected) - disable mutation
- Finally write out results. (File written in previous step has to be overwritten/enhanced)

JUnitCore

- JUnitCore can be used to run JUnit tests.
- JUnitCore allows to register listeners.
- `Class.forName()` returns a class for a given String.

Pitfalls

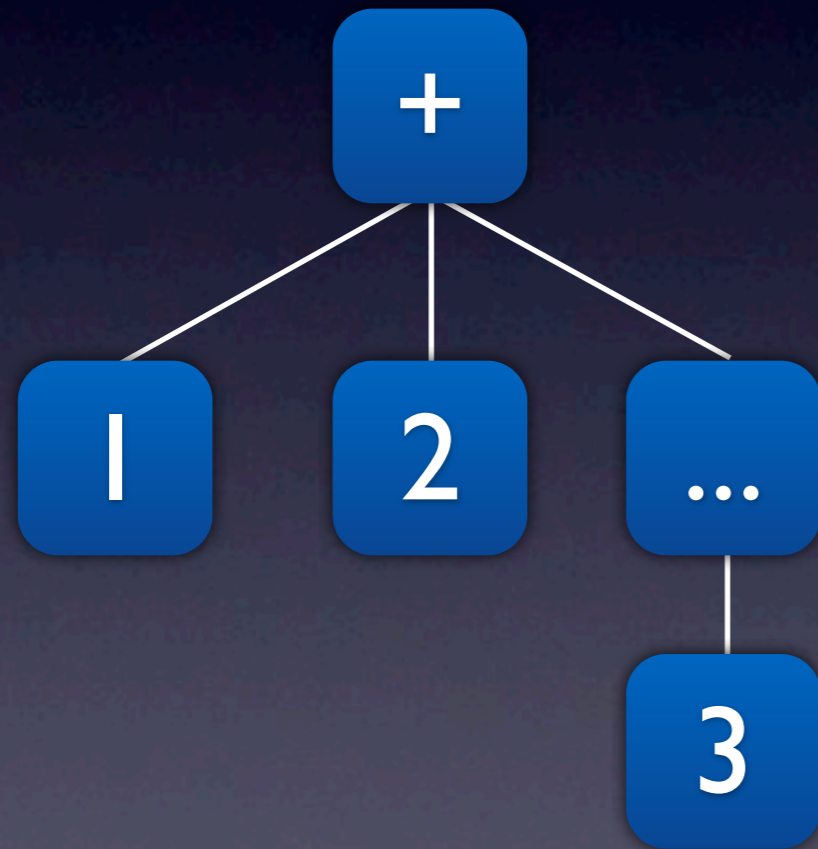
$1 + 2$

vs.

$1 + 2 + 3$

Pitfalls

AST



1 + 2

1 + 2 + 3

Pitfalls

1 + 2

vs.

1 + 2 + 3

- AstParser handles these expressions differently.
- `node.getLocationInParent()` can either return `ChildPropertyDescriptor` or `ChildListPropertyDescriptor`.

Where not to mutate?

```
switch (m | () * 2) {  
  case 1:  
    ...  
    break;  
  case 1+2:  
}
```

- Condition in switch can be mutated.
- Case statement cannot be mutated.
- Leads to compilation failures.
- Do not visit anything below a SwitchCase.

Where not to mutate?



$x + 0$

- Zero Literals should not be mutated.
- See *NullMutationClass*.