

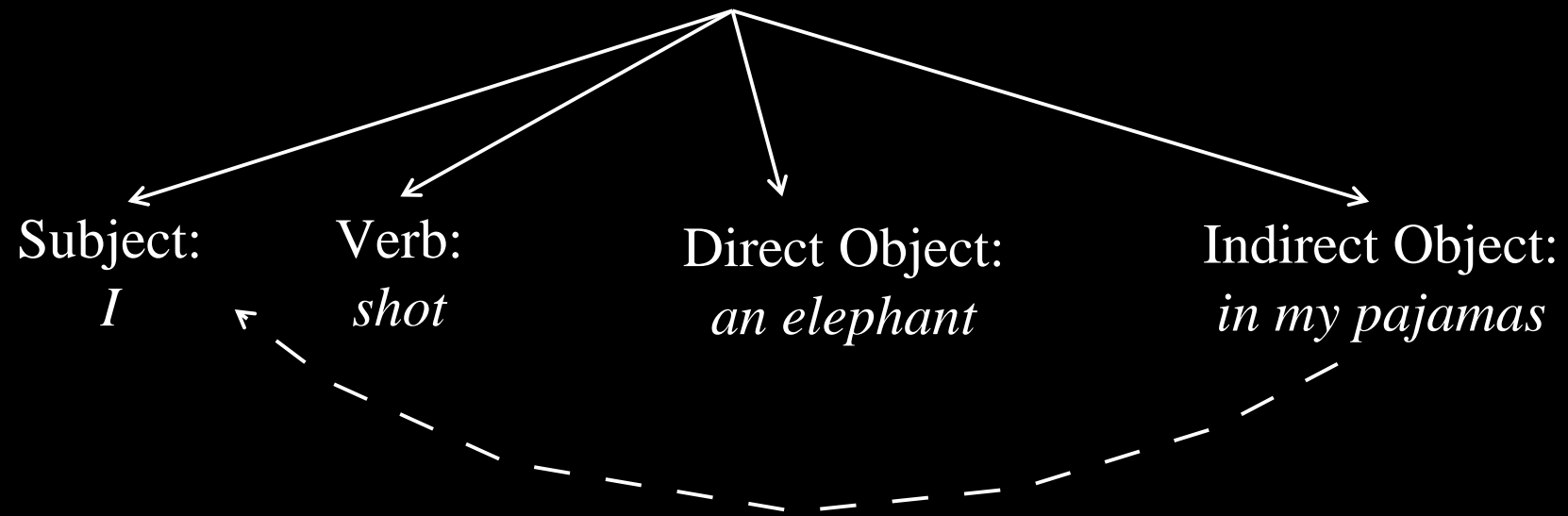


AST and Why You Should Care

Hamlet D'Arcy
Senior Software Development Engineer

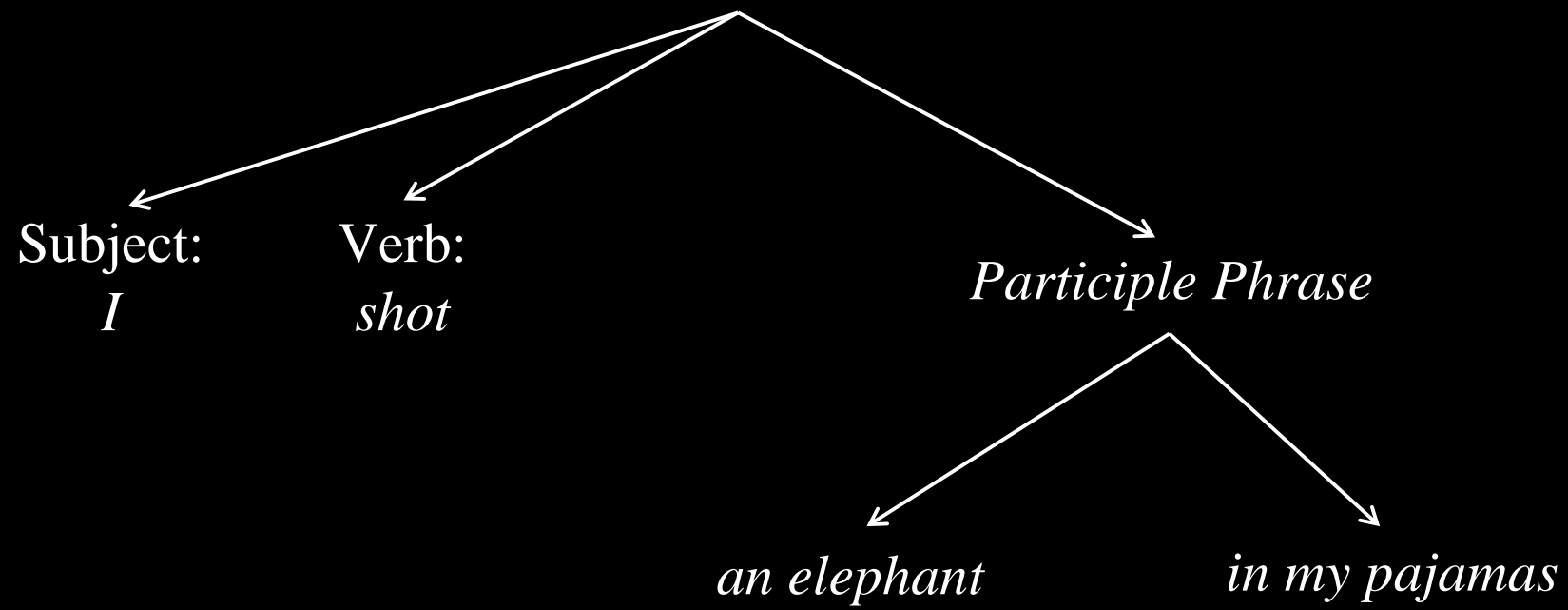


I shot an elephant in my pajamas

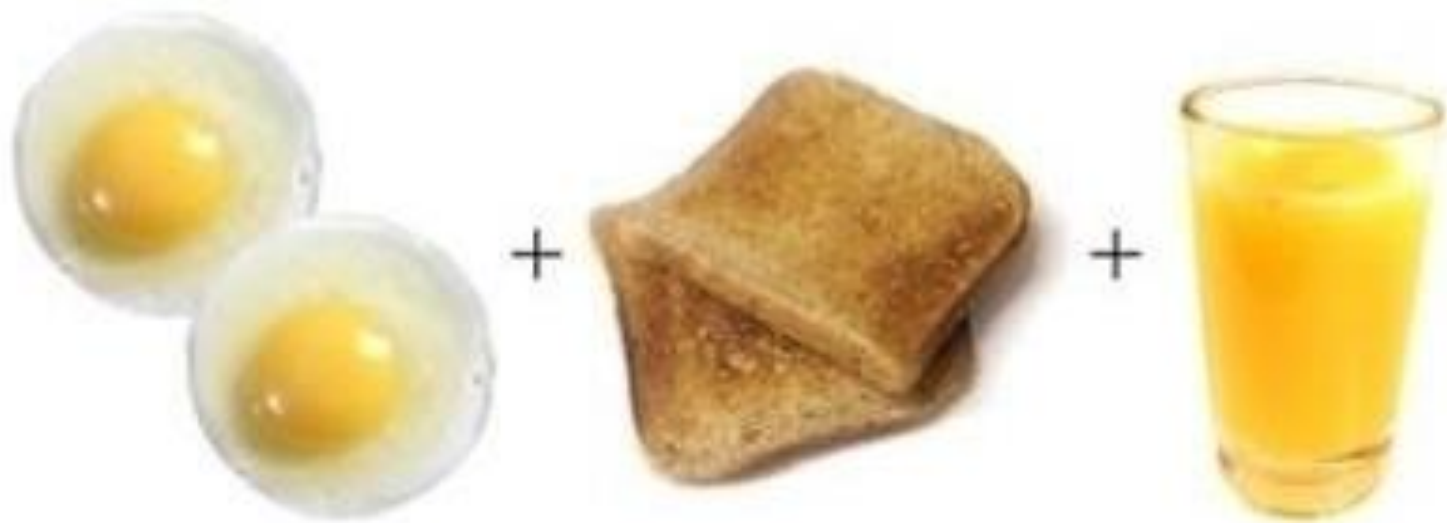


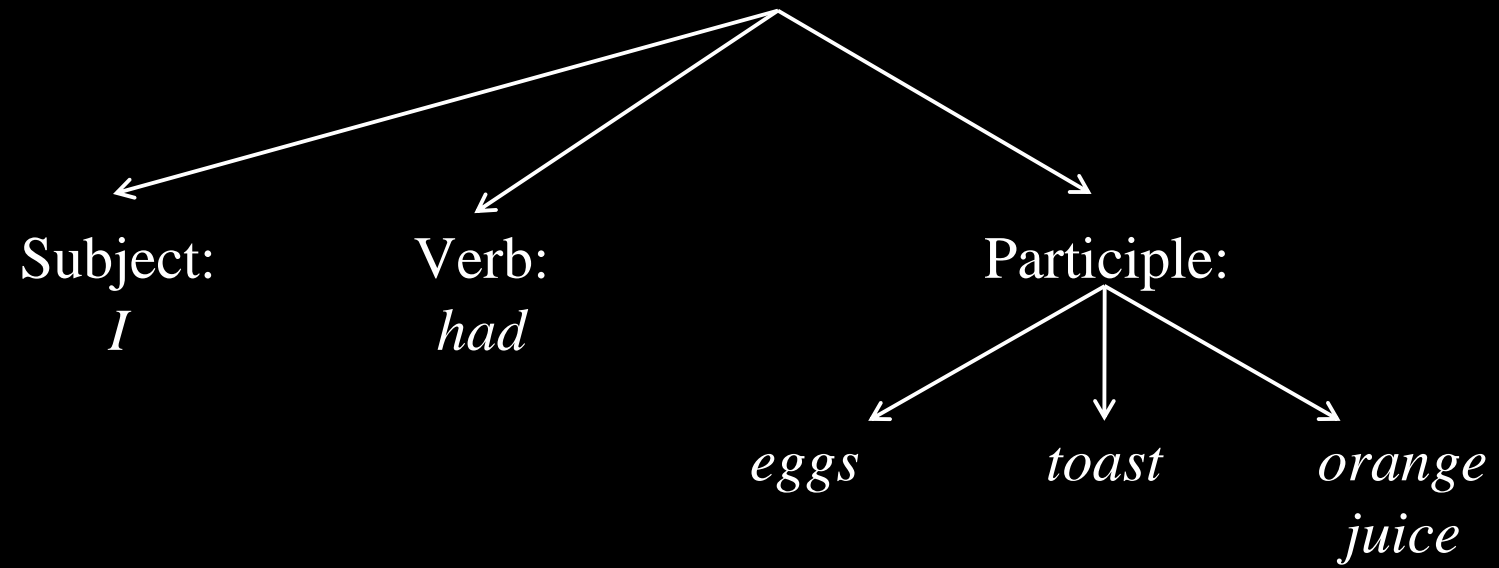
I shot an elephant in my pajamas

How he got in my pajamas, I'll never know

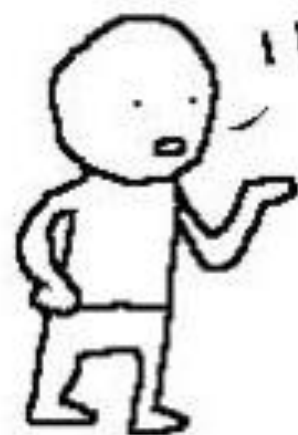


I had eggs, toast, and orange juice.



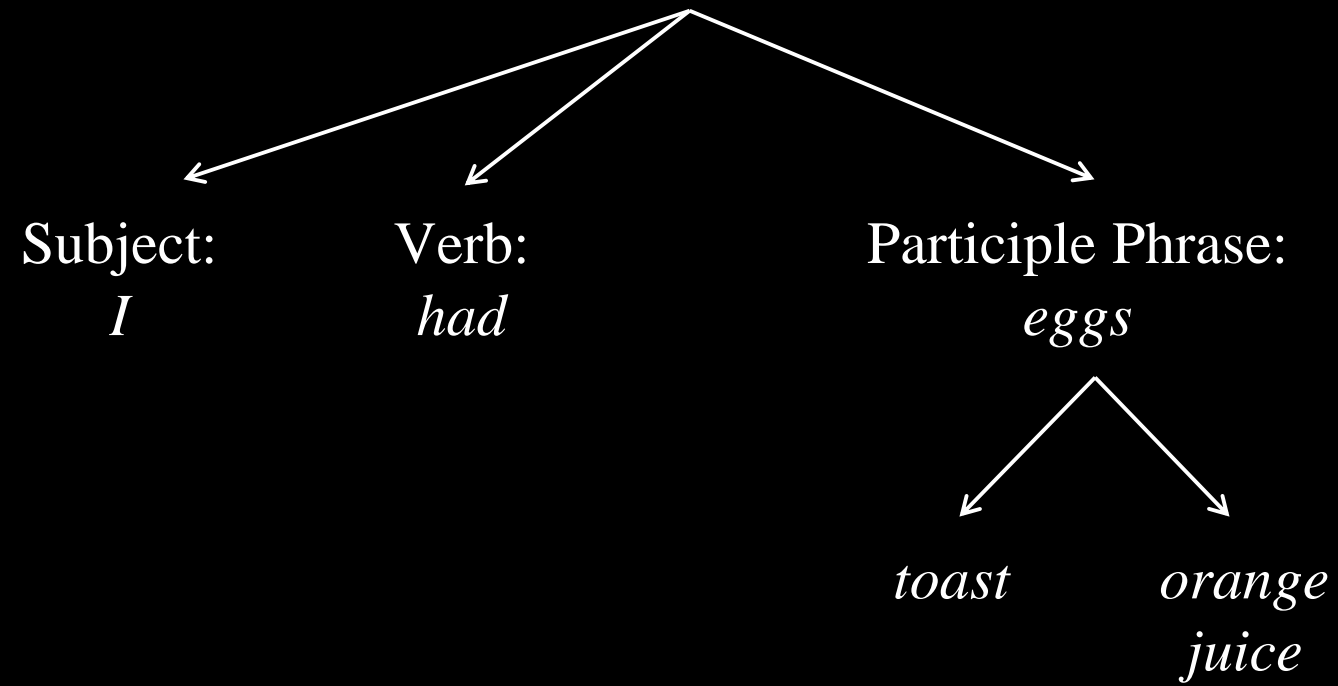


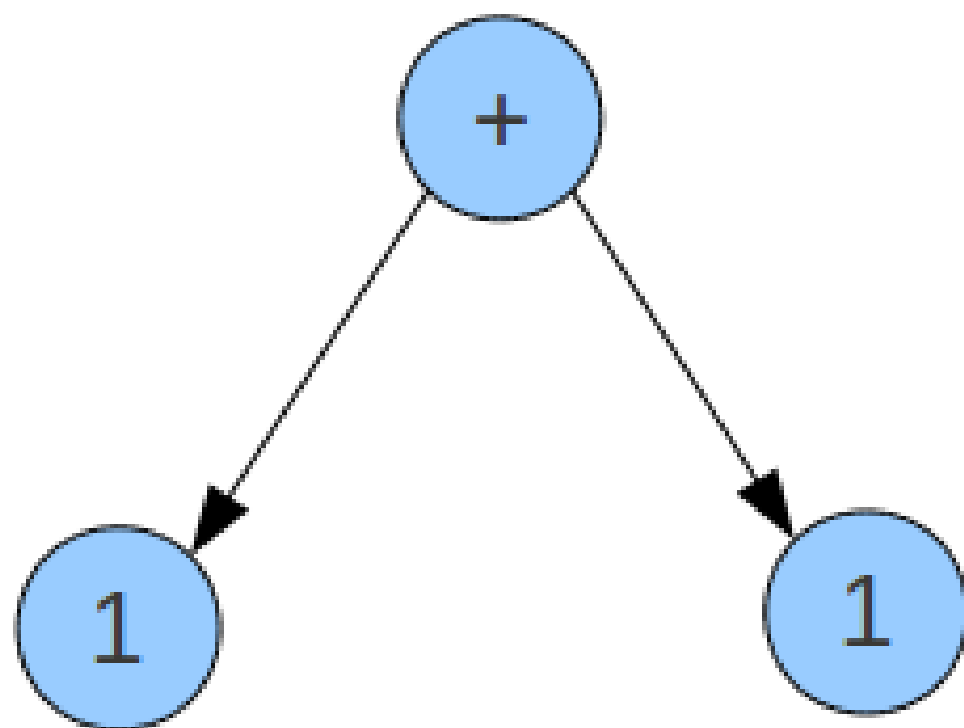
I had eggs, toast and orange juice.

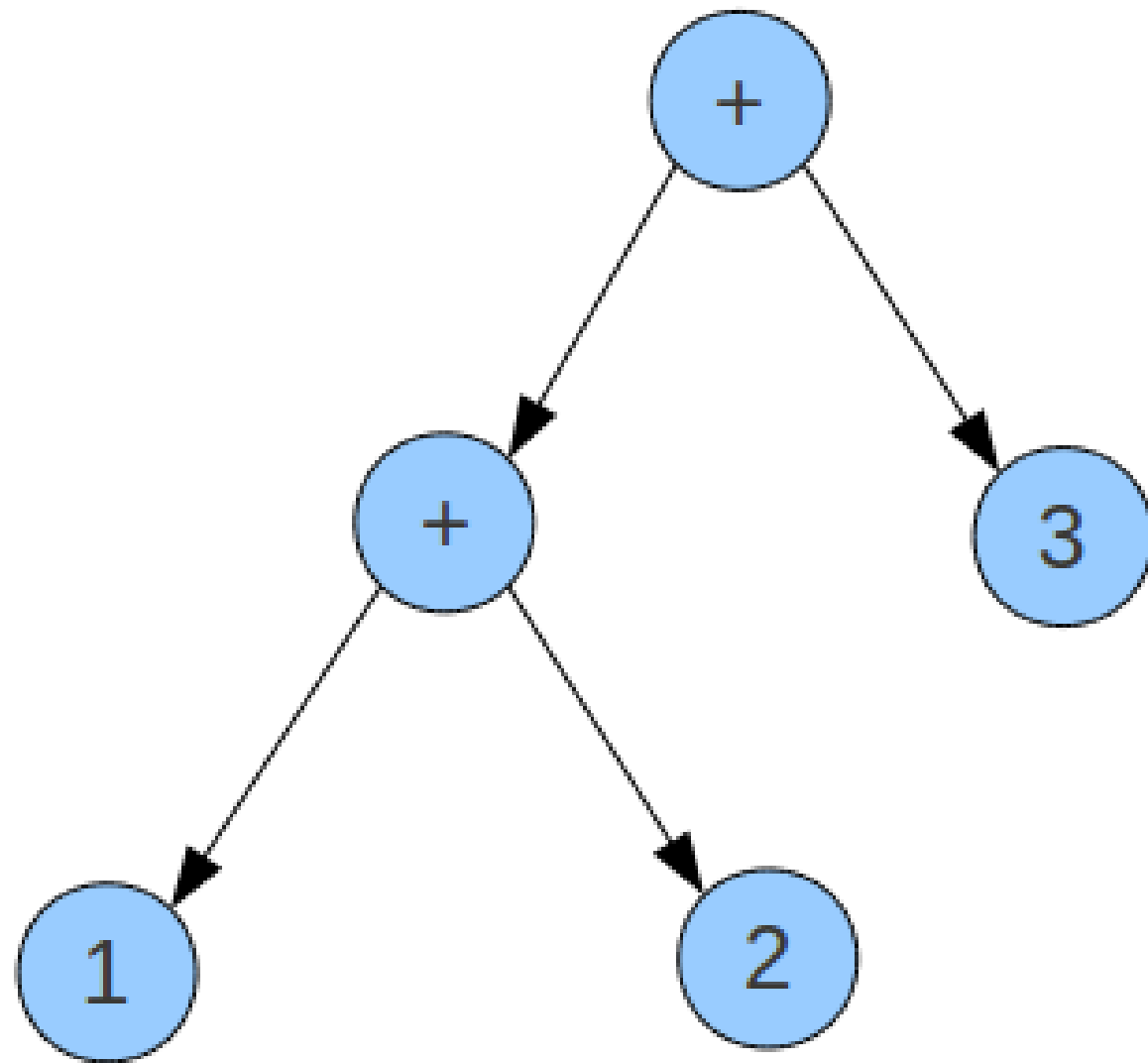


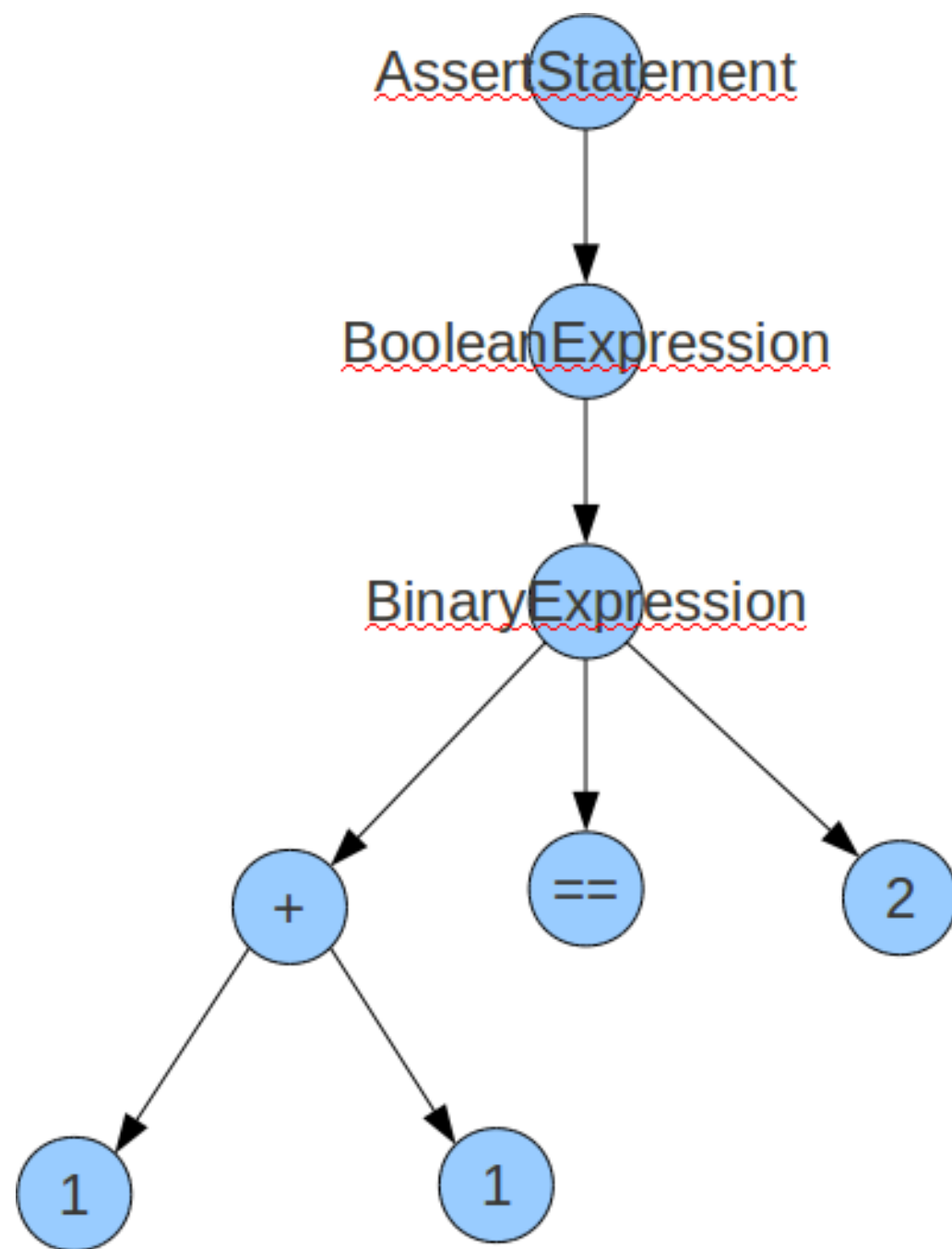
I HAD EGGS

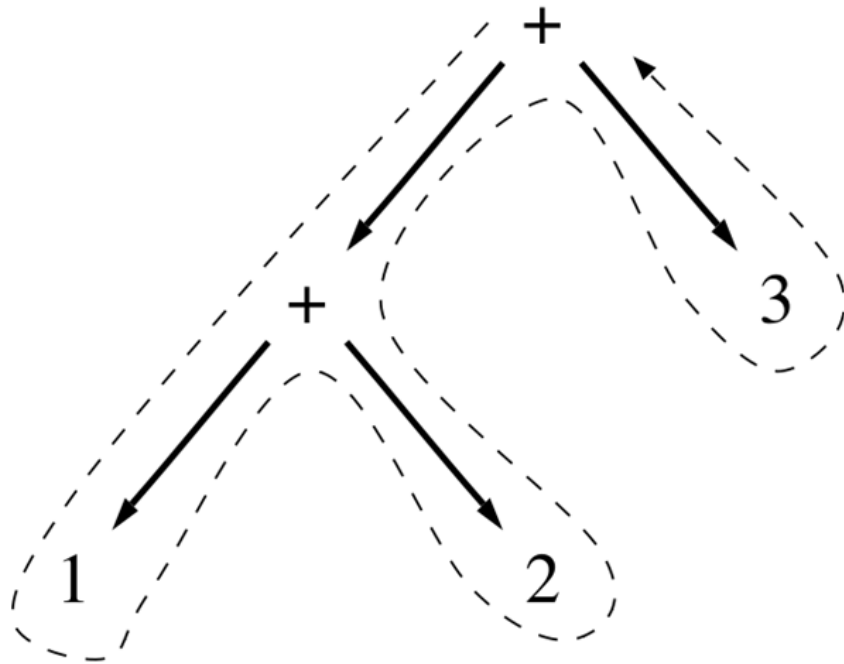




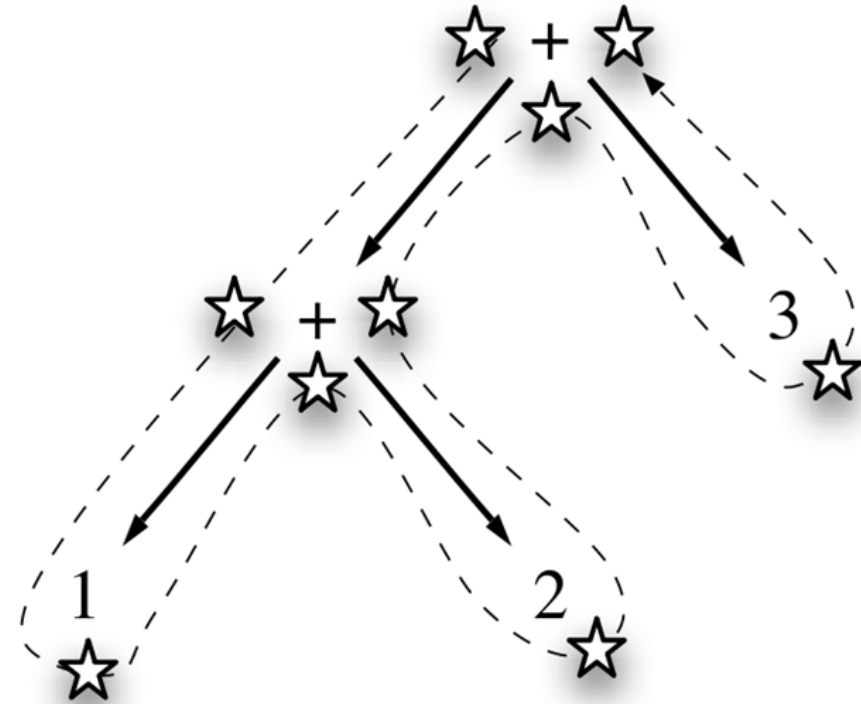








Depth first tree walk path:
node discovery, finishing



Traversal order:
action execution opportunities

From Terrence Parr's “Language Implementation Patterns”

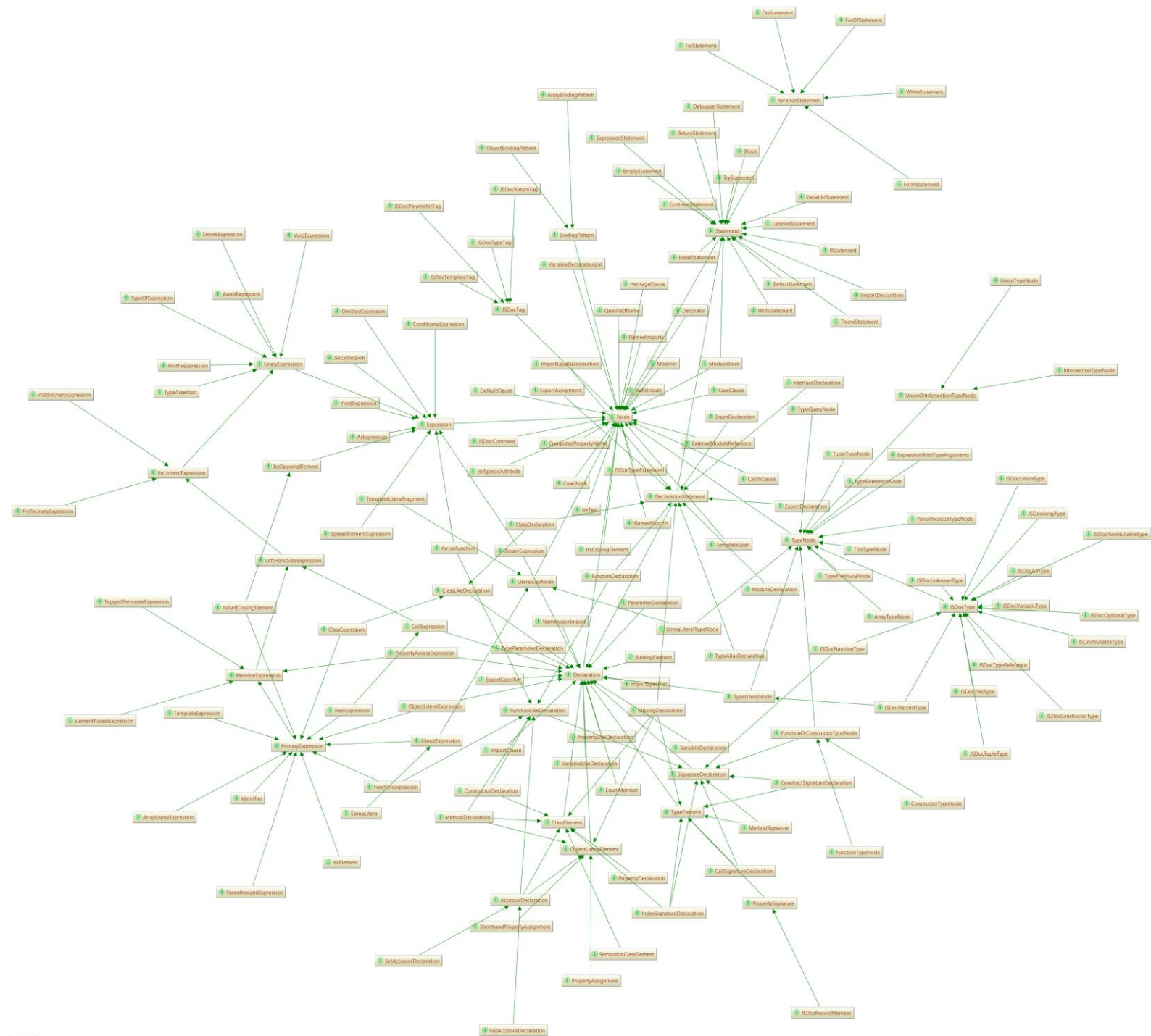
What's in the AST?

- dense
- convenient
- meaningful

What's in the AST?

- Homogenous?
- Normalized Heterogeneous?
- Irregular Heterogeneous?

https://github.com/HamletDRC/presentations/blob/master/asttransformations/ast_homo_vs_hetero.txt



What's in the AST?

- Embedded Heterogeneous Tree Walker
- External Tree Visitor
- Tree Grammar
- Tree Pattern Matcher

https://github.com/HamletDRC/presentations/blob/master/asttransformations/ast_walking.txt

visitAnyKeyword(node: ts.Node)	visitForInStatement(node: ts.ForInStatement)	visitPrefixUnaryExpression(node: ts.PrefixUnaryExpression)
visitArrayLiteralExpression(node: ts.ArrayLiteralExpression)	visitForOfStatement(node: ts.ForOfStatement)	visitPropertyAccessExpression(node: ts.PropertyAccessExpression)
visitArrowFunction(node: ts.FunctionLikeDeclaration)	visitFunctionDeclaration(node: ts.FunctionDeclaration)	visitPropertyAssignment(node: ts.PropertyAssignment)
visitBinaryExpression(node: ts.BinaryExpression)	visitFunctionExpression(node: ts.FunctionExpression)	visitPropertyDeclaration(node: ts.PropertyDeclaration)
visitBindingElement(node: ts.BindingElement)	visitFunctionType(node: ts.FunctionOrConstructorTypeNode)	visitPropertySignature(node: ts.Node)
visitBindingPattern(node: ts.BindingPattern)	visitGetAccessor(node: ts.AccessorDeclaration)	visitRegularExpressionLiteral(node: ts.Node)
visitBlock(node: ts.Block)	visitIdentifier(node: ts.Identifier)	visitReturnStatement(node: ts.ReturnStatement)
visitBreakStatement(node: ts.BreakOrContinueStatement)	visitIfStatement(node: ts.IfStatement)	visitSetAccessor(node: ts.AccessorDeclaration)
visitCallExpression(node: ts.CallExpression)	visitImportDeclaration(node: ts.ImportDeclaration)	visitSourceFile(node: ts.SourceFile)
visitCallSignature(node: ts.SignatureDeclaration)	visitImportEqualsDeclaration(node: ts.ImportEqualsDeclaration)	visitStringLiteral(node: ts.StringLiteral)
visitCaseClause(node: ts.CaseClause)	visitIndexSignatureDeclaration(node: ts.IndexSignatureDeclaration)	visitSwitchStatement(node: ts.SwitchStatement)
visitClassDeclaration(node: ts.ClassDeclaration)	visitInterfaceDeclaration(node: ts.InterfaceDeclaration)	visitTemplateExpression(node: ts.TemplateExpression)
visitClassExpression(node: ts.ClassExpression)	visitJsxAttribute(node: ts.JsxAttribute)	visitThrowStatement(node: ts.ThrowStatement)
visitCatchClause(node: ts.CatchClause)	visitJsxElement(node: ts.JsxElement)	visitTryStatement(node: ts.TryStatement)
visitConditionalExpression(node: ts.ConditionalExpression)	visitJsxExpression(node: ts.JsxExpression)	visitTypeAssertionExpression(node: ts.TypeAssertion)
visitConstructorDeclaration(node: ts.ConstructorDeclaration)	visitJsxSelfClosingElement(node: ts.JsxSelfClosingElement)	visitTypeLiteral(node: ts.TypeLiteralNode)
visitConstructorType(node: ts.FunctionOrConstructorTypeNode)	visitLabeledStatement(node: ts.LabeledStatement)	visitTypeReference(node: ts.TypeReferenceNode)
visitContinueStatement(node: ts.BreakOrContinueStatement)	visitMethodDeclaration(node: ts.MethodDeclaration)	visitVariableDeclaration(node: ts.VariableDeclaration)
visitDebuggerStatement(node: ts.Statement)	visitMethodSignature(node: ts.SignatureDeclaration)	visitVariableStatement(node: ts.VariableStatement)
visitDefaultClause(node: ts.DefaultClause)	visitModuleDeclaration(node: ts.ModuleDeclaration)	visitWhileStatement(node: ts.WhileStatement)
visitDoStatement(node: ts.DoStatement)	visitNamedImports(node: ts.NamedImports)	visitWithStatement(node: ts.WithStatement)
visitElementAccessExpression(node: ts.ElementAccessExpression)	visitNamespaceImport(node: ts.NamespaceImport)	visitNode(node: ts.Node)
visitEnumDeclaration(node: ts.EnumDeclaration)	visitNewExpression(node: ts.NewExpression)	walkChildren(node: ts.Node)
visitExportAssignment(node: ts.ExportAssignment)	visitObjectLiteralExpression(node: ts.ObjectLiteralExpression)	
visitExpressionStatement(node: ts.ExpressionStatement)	visitParameterDeclaration(node: ts.ParameterDeclaration)	
visitForStatement(node: ts.ForStatement)	visitPostfixUnaryExpression(node: ts.PostfixUnaryExpression)	

tslint-microsoft-contrib Demo

Resources

- [typescriptServices.d.ts](#)
- [syntaxWalker.d.ts](#)
- Using the Compiler API
 - <https://github.com/Microsoft/TypeScript/wiki/Using-the-Compiler-API>
- Investigating Typescript Compiler APIs
 - <http://blog.scottlogic.com/2015/01/20/typescript-compiler-api.html>
- “typescript-compiler-docs” project
 - <https://github.com/basarat/typescript-compiler-docs>