

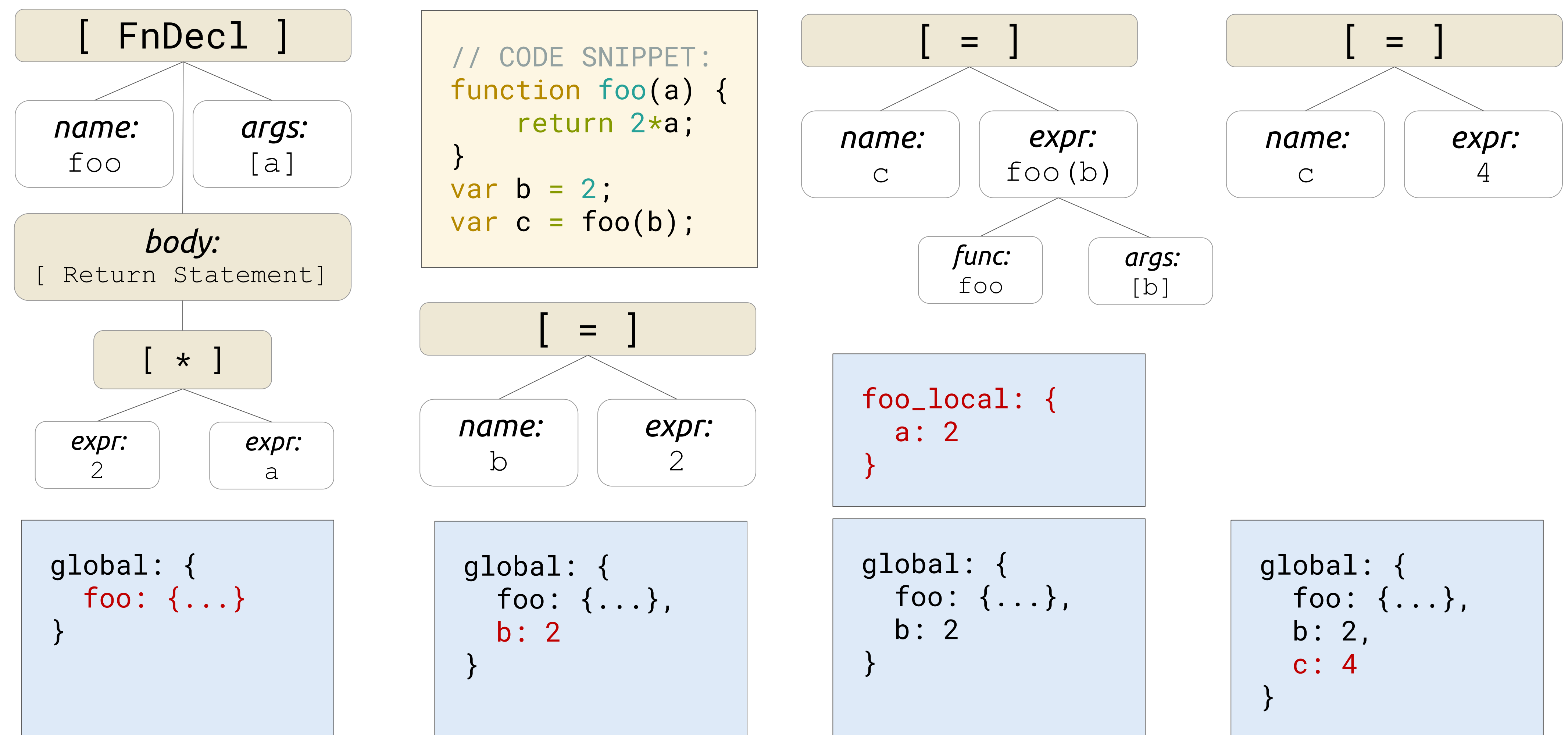
A Rustic JavaScript Interpreter

Terry Sun, Sam Rossi
Prof. Steve Zdancewic

Abstract

- Js.rs is a prototype server-side JavaScript interpreter, implementing a core subset of language features
- JavaScript is ubiquitous—runs on every computer and browser
- Written entirely in Rust, Js.rs provides memory safety guarantees, unlike existing C/C++ interpreters, with low performance overhead

Process



Language Features

- Functions & function closures

```
function add() {
  var x = 0;
  return function() { x += 1; };
}
```

- Control Flow

```
if { ... } else if { ... } else { ... }
for (var i = 0; i < 10; i++) { ... }
```

- Error Handling

```
try { ... } catch { ... } finally { ... }
```

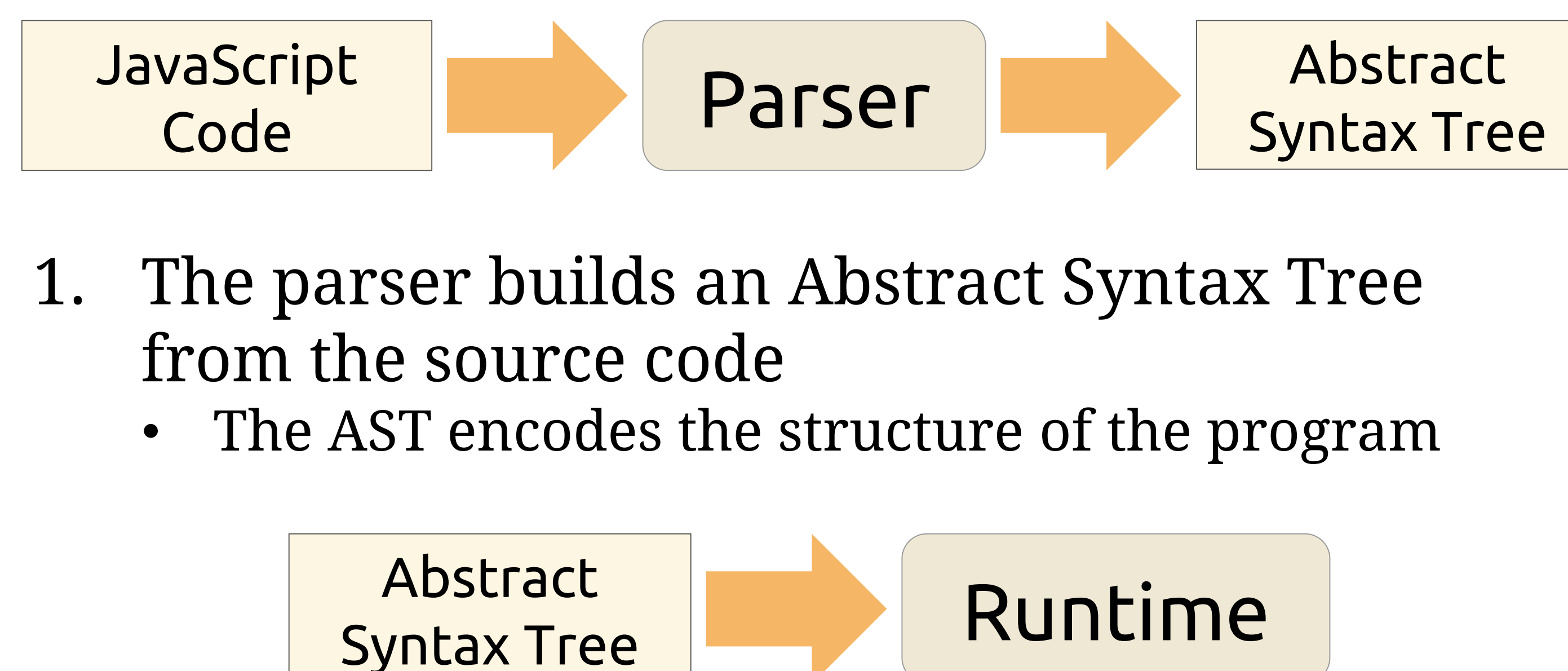
- Unary, Binary Operators

```
x <= (2 * y);
```

- Objects, Arrays, and Prototypes

```
var a = [1, 2, 3];
var o = { a: "hello, world!" };
```

Design



1. The parser builds an Abstract Syntax Tree from the source code
 - The AST encodes the structure of the program

2. The runtime traverses the AST:
 - Executes the code in each node
 - Maintains a state of local variables

Results

- Sputnik is Google's ECMAScript 3 conformance test suite
- Js.rs passes tests in 73 of 111 (66%) selected categories from Sputnik
- Rust is a robust language to work in, but available tools are fragile



<https://github.com/rustscript/js.rs>