PLRG
Programming Language
Research Group

JavaScript Static Analysis with Evolving Engines and Specification

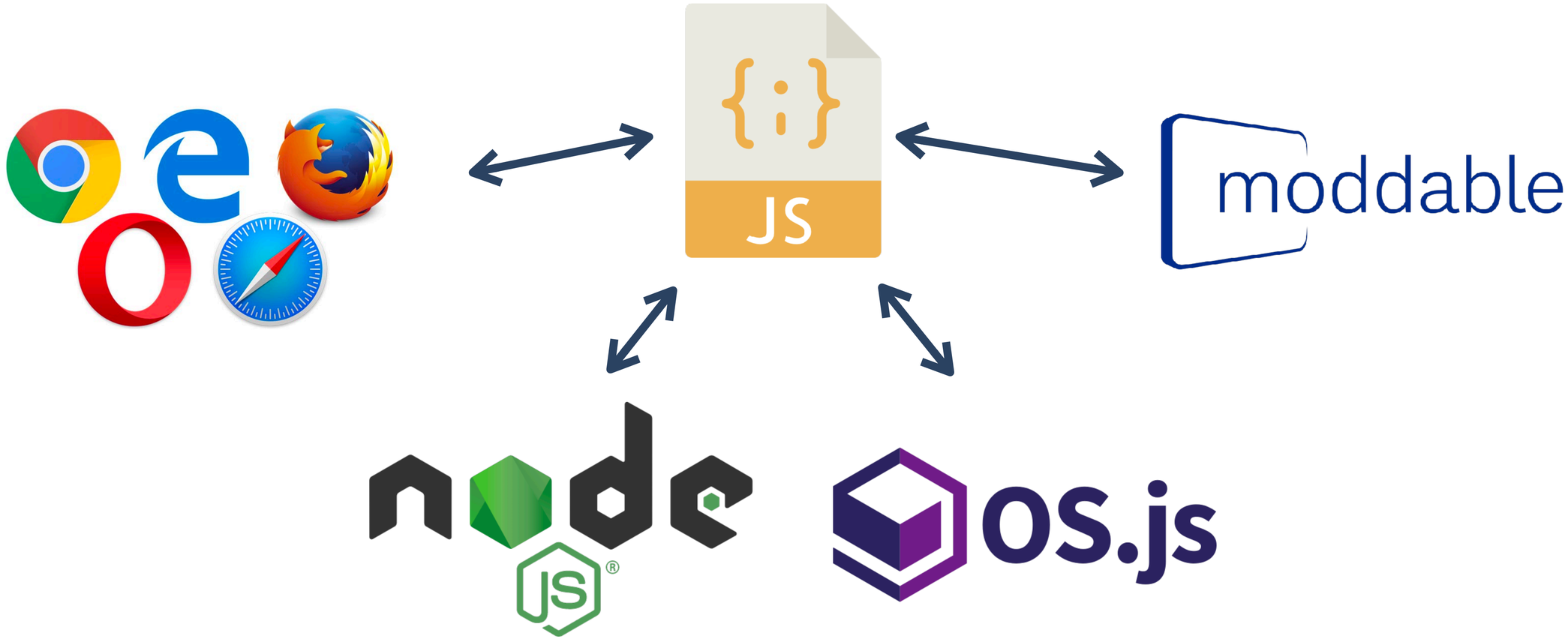
Jihyeok Park

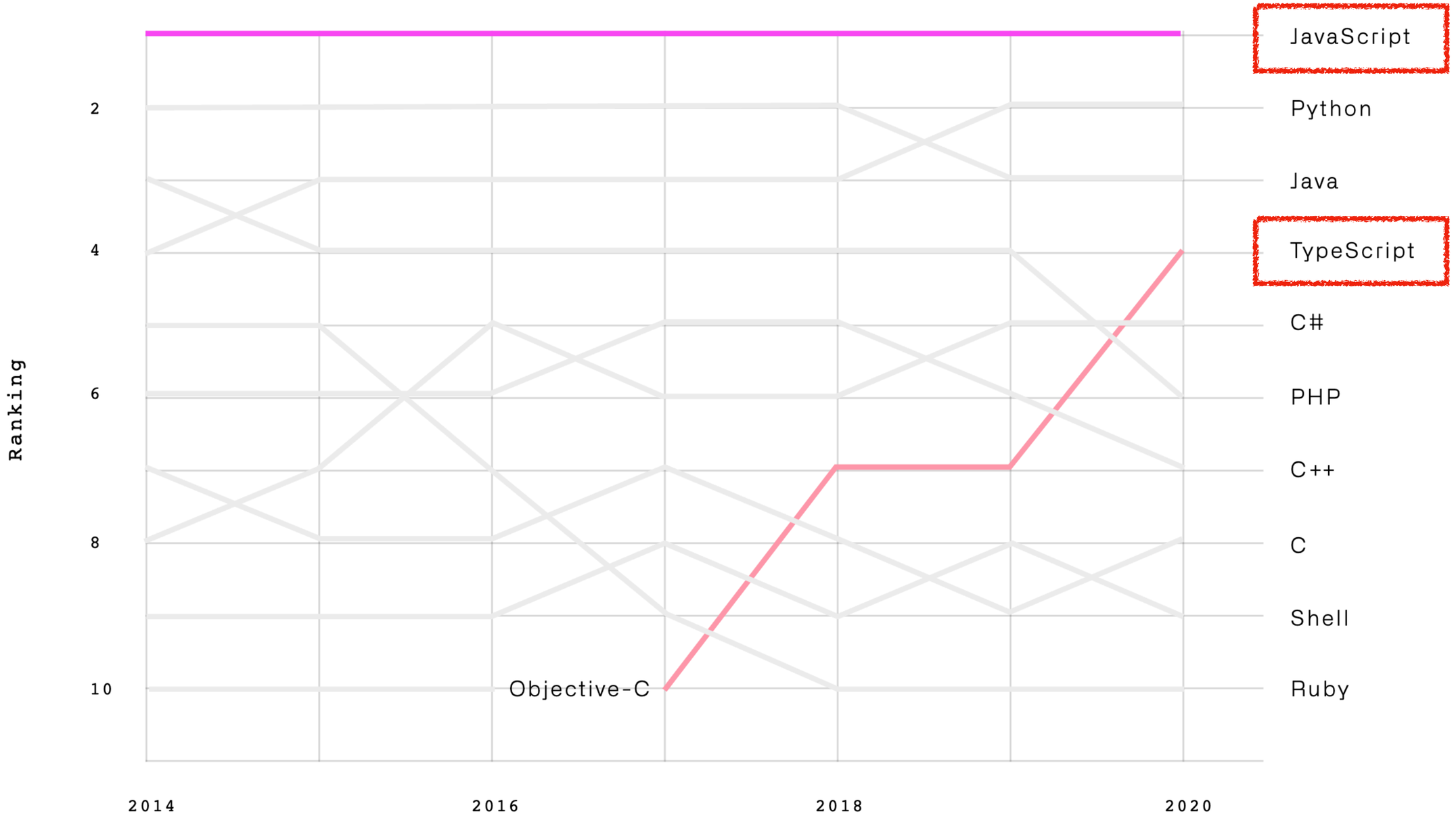
PLRG @ KAIST

ECOOP/ISSTA 2021 Doctoral Symposium

July 13, 2021

JavaScript is Everywhere





https://octoverse.github.com/

JavaScript Complex Semantics

```
function f(x) { return x == !x; }
```

Always return `false`?

JavaScript Complex Semantics

```
function f(x) { return x == !x; }
```

Always return `false`?

NO!!

```
f([]) -> [] == ![]
      -> [] == false
      -> +[] == +false
      -> 0 == 0
      -> true
```

ECMAScript: JavaScript Specification

Syntax

ArrayLiteral`[Yield, Await]` :

[*Elision*`opt` **]**

[*ElementList*`[?Yield, ?Await]` **]**

[*ElementList*`[?Yield, ?Await]` **,** *Elision*`opt` **]**

The production of *ArrayLiteral* in ES10

Semantics

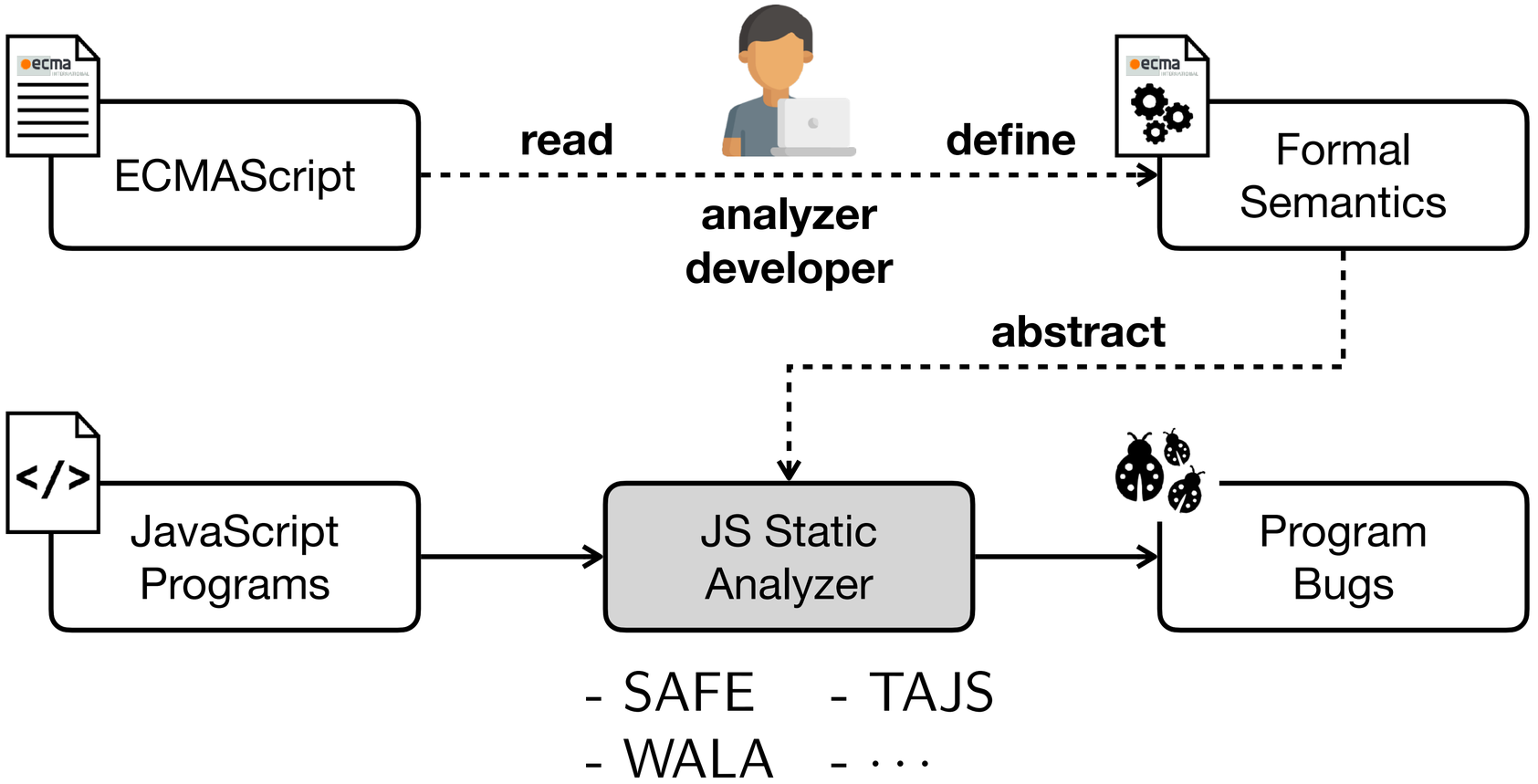
12.2.5.3 Runtime Semantics: Evaluation

ArrayLiteral **:** **[** *Elision* **]**

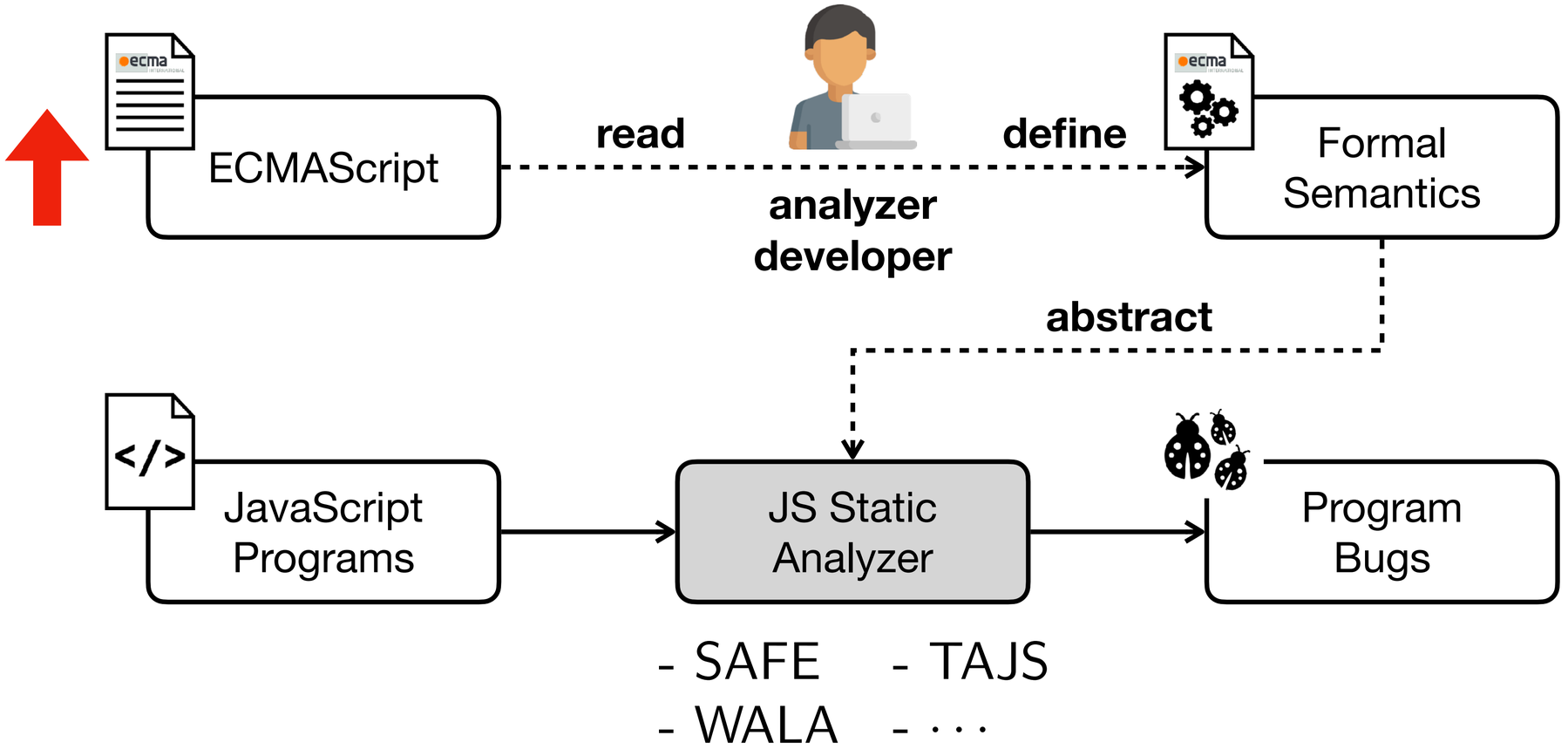
1. Let *array* be ! ArrayCreate(0).
2. Let *pad* be the ElisionWidth of *Elision*; if *Elision* is not present, use the numeric value zero.
3. Perform Set(*array*, `"length"`, ToUint32(*pad*), **false**).
4. NOTE: The above Set cannot fail because of the nature of the object returned by ArrayCreate.
5. Return *array*.

The Evaluation algorithm for the first alternative of *ArrayLiteral* in ES10

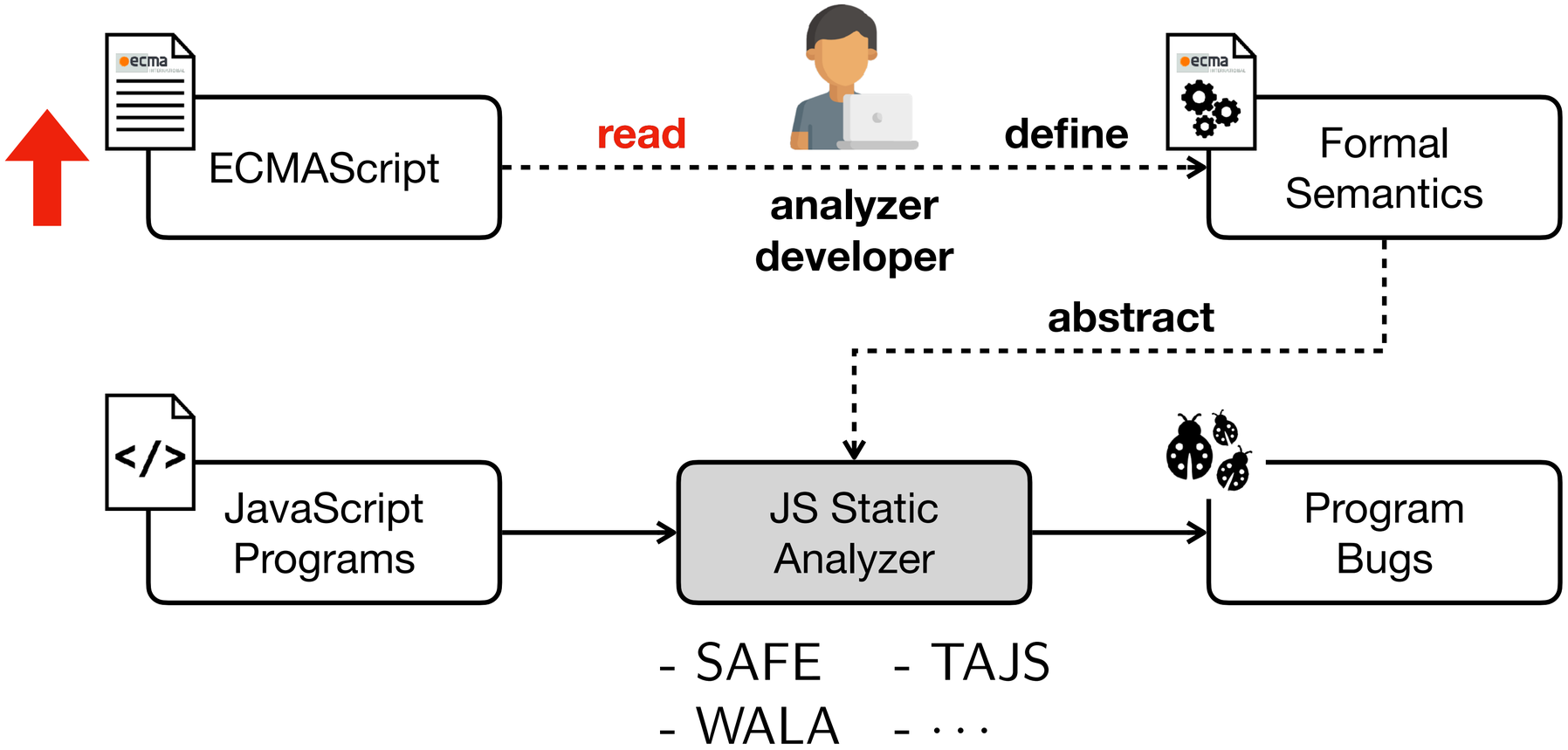
Problem: JavaScript Static Analyzer



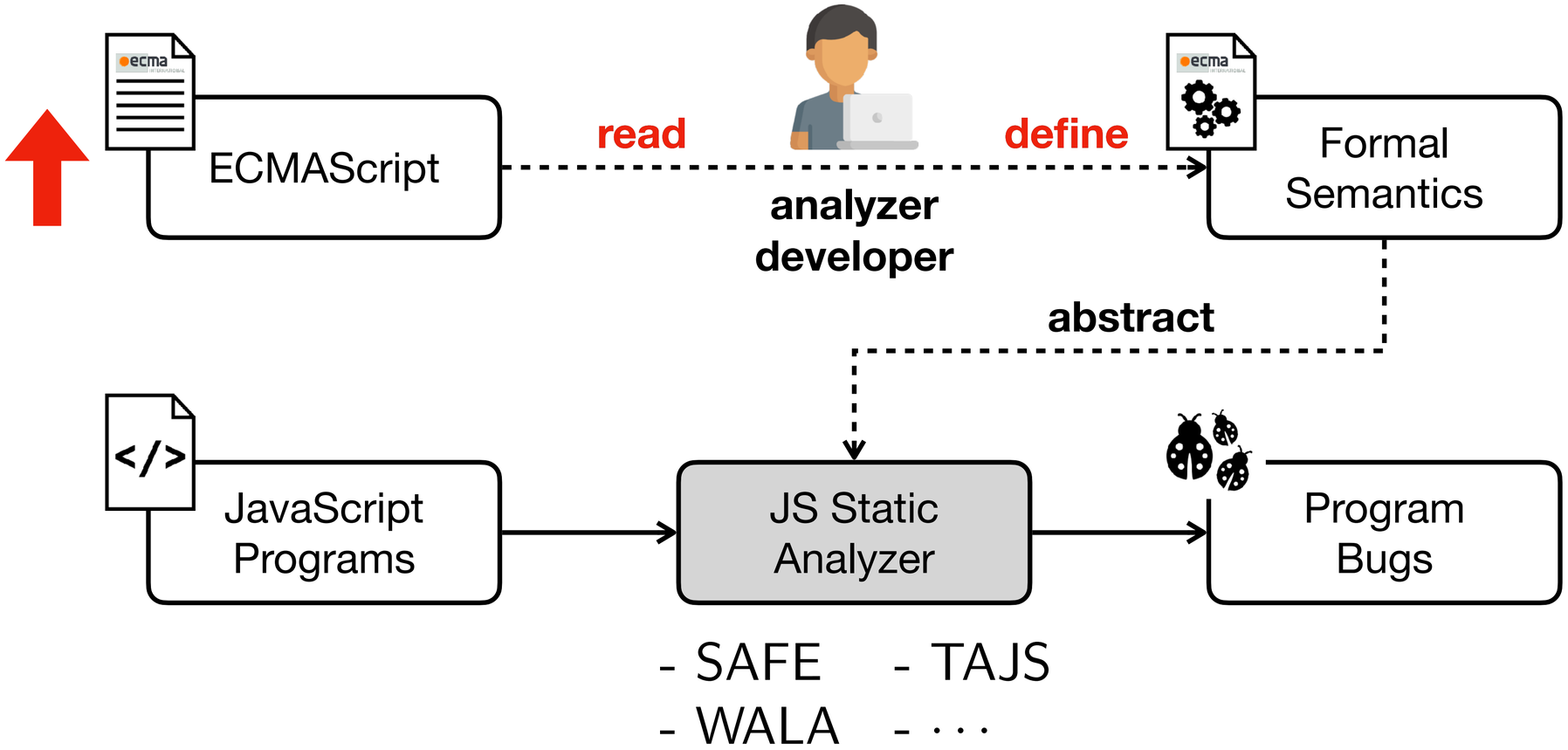
Problem: JavaScript Static Analyzer



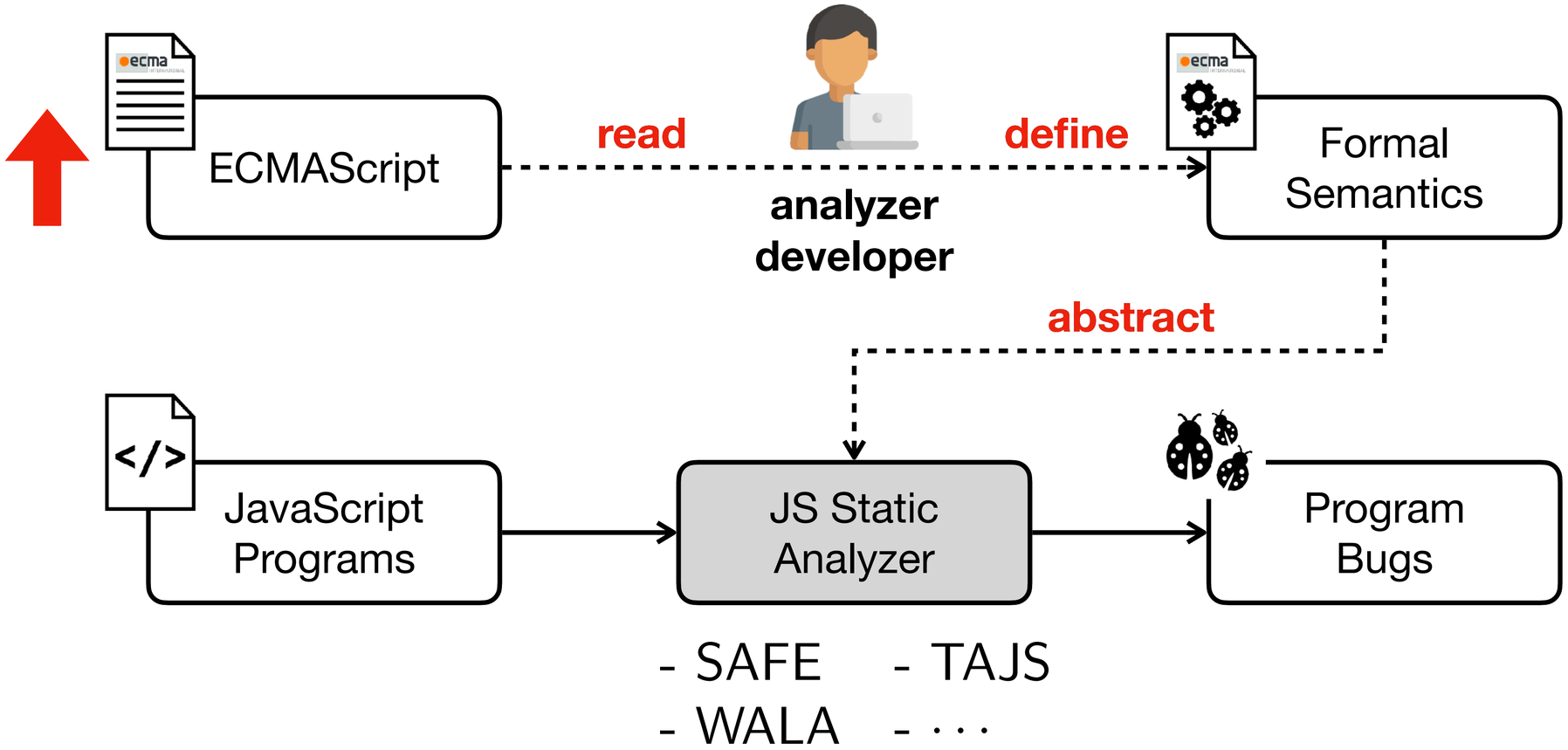
Problem: JavaScript Static Analyzer



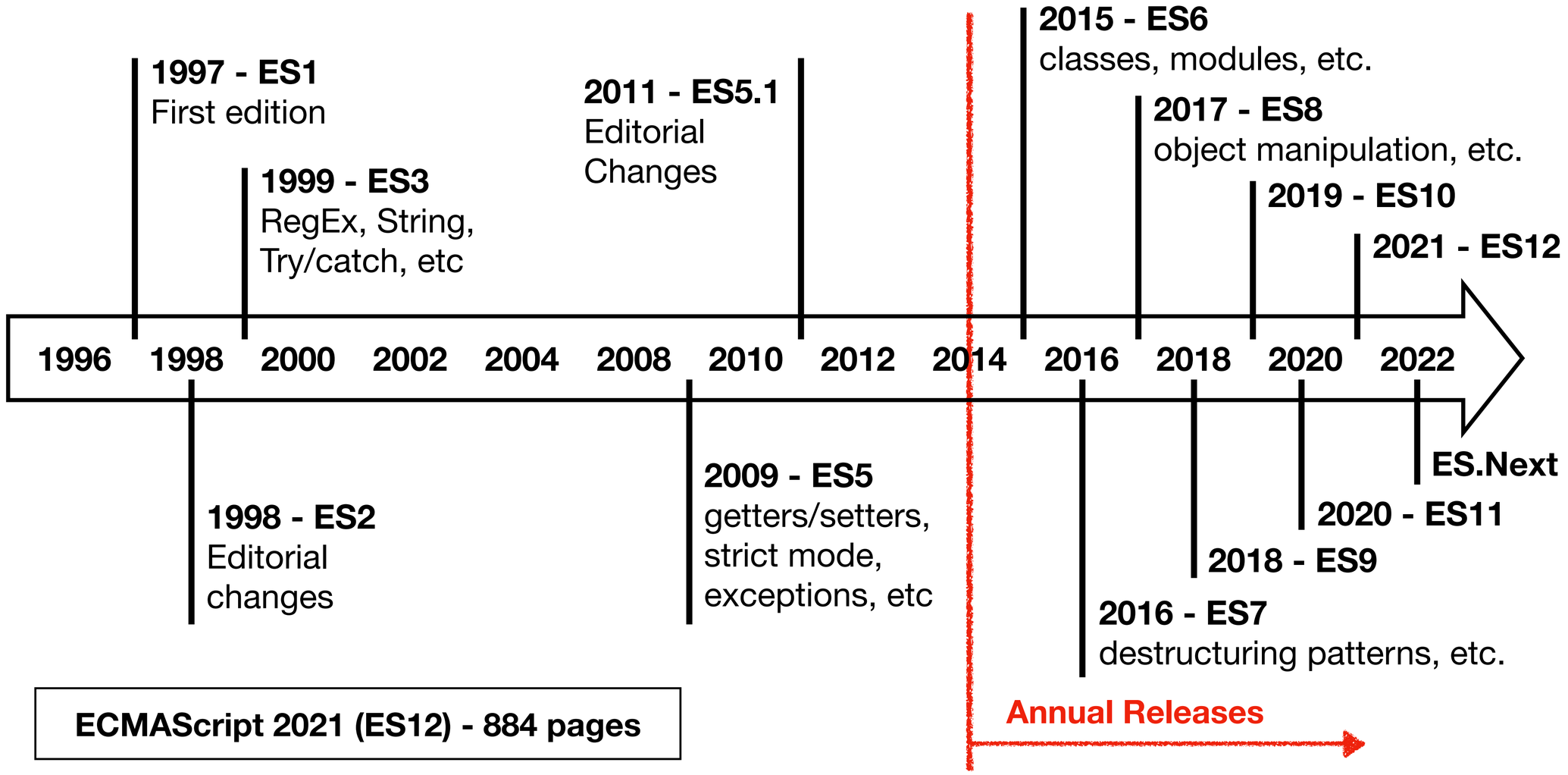
Problem: JavaScript Static Analyzer



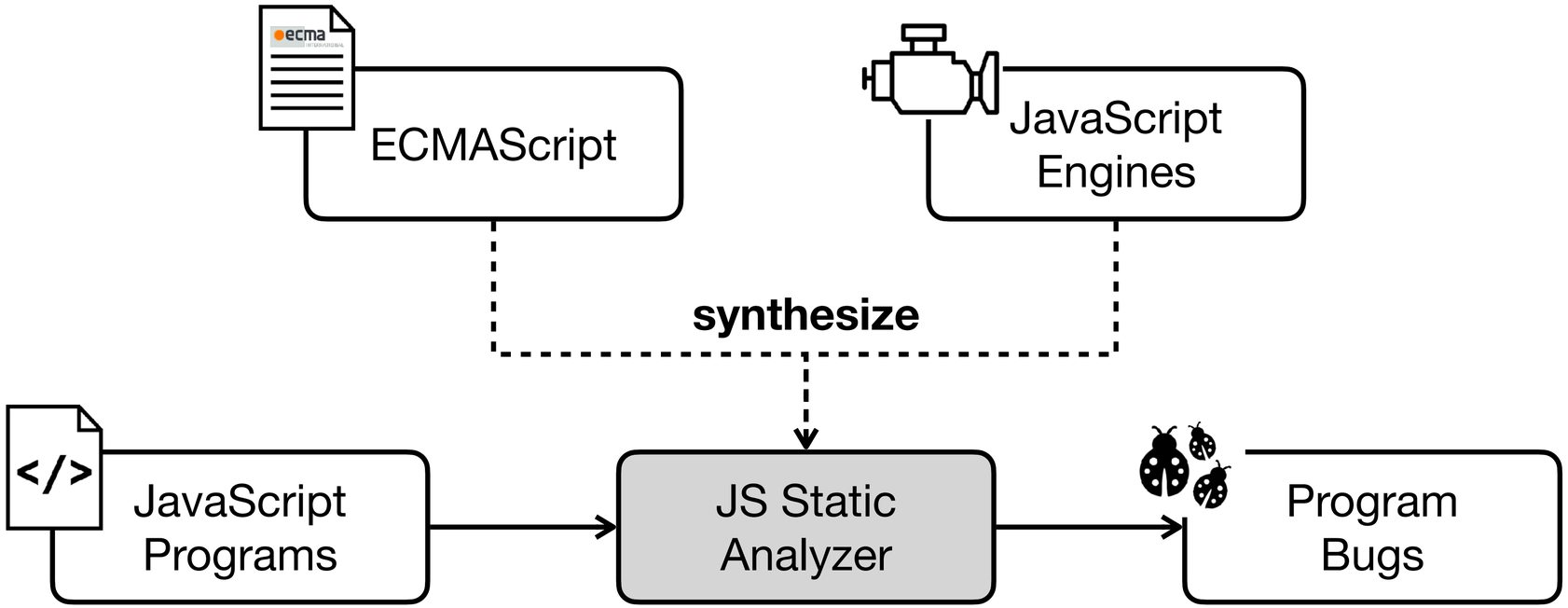
Problem: JavaScript Static Analyzer



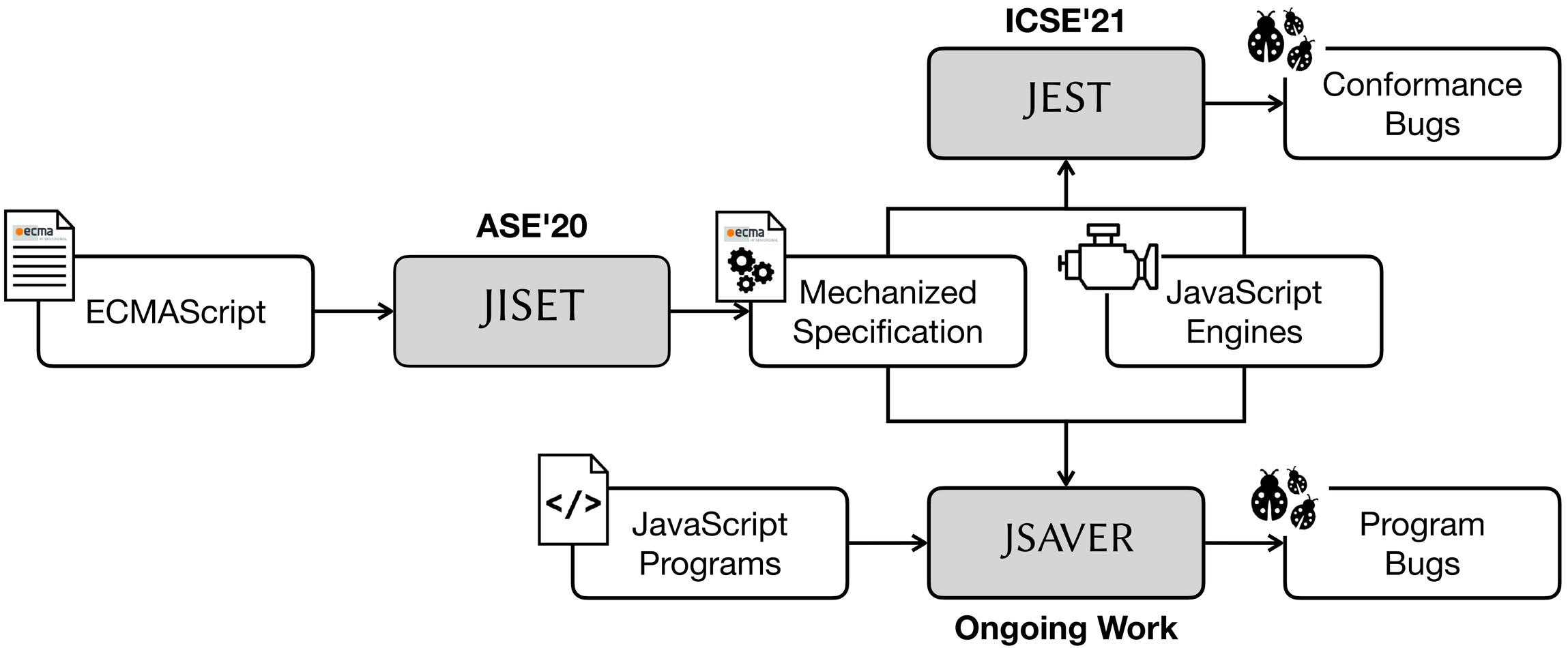
Problem: Fast Evolving JavaScript



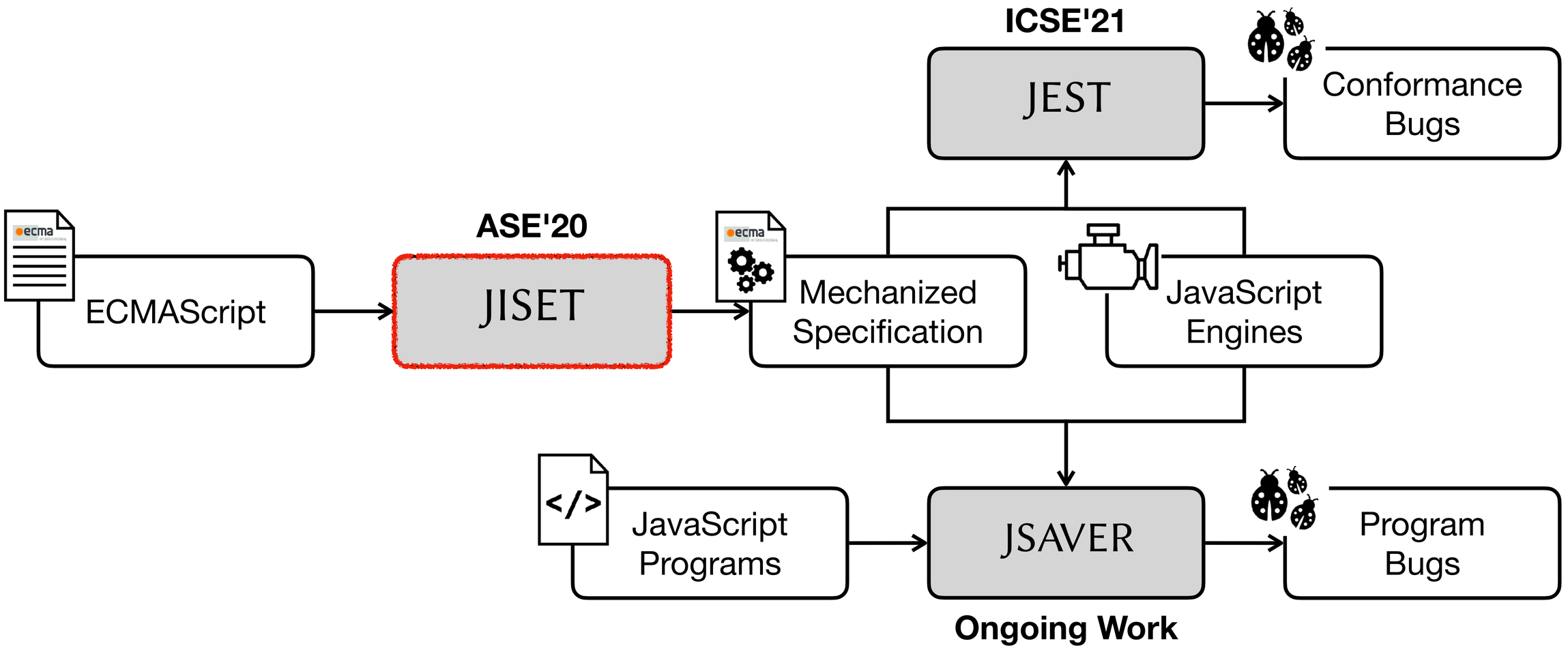
Core Idea: Synthesis of JS Static Analyzer



Overall Structure

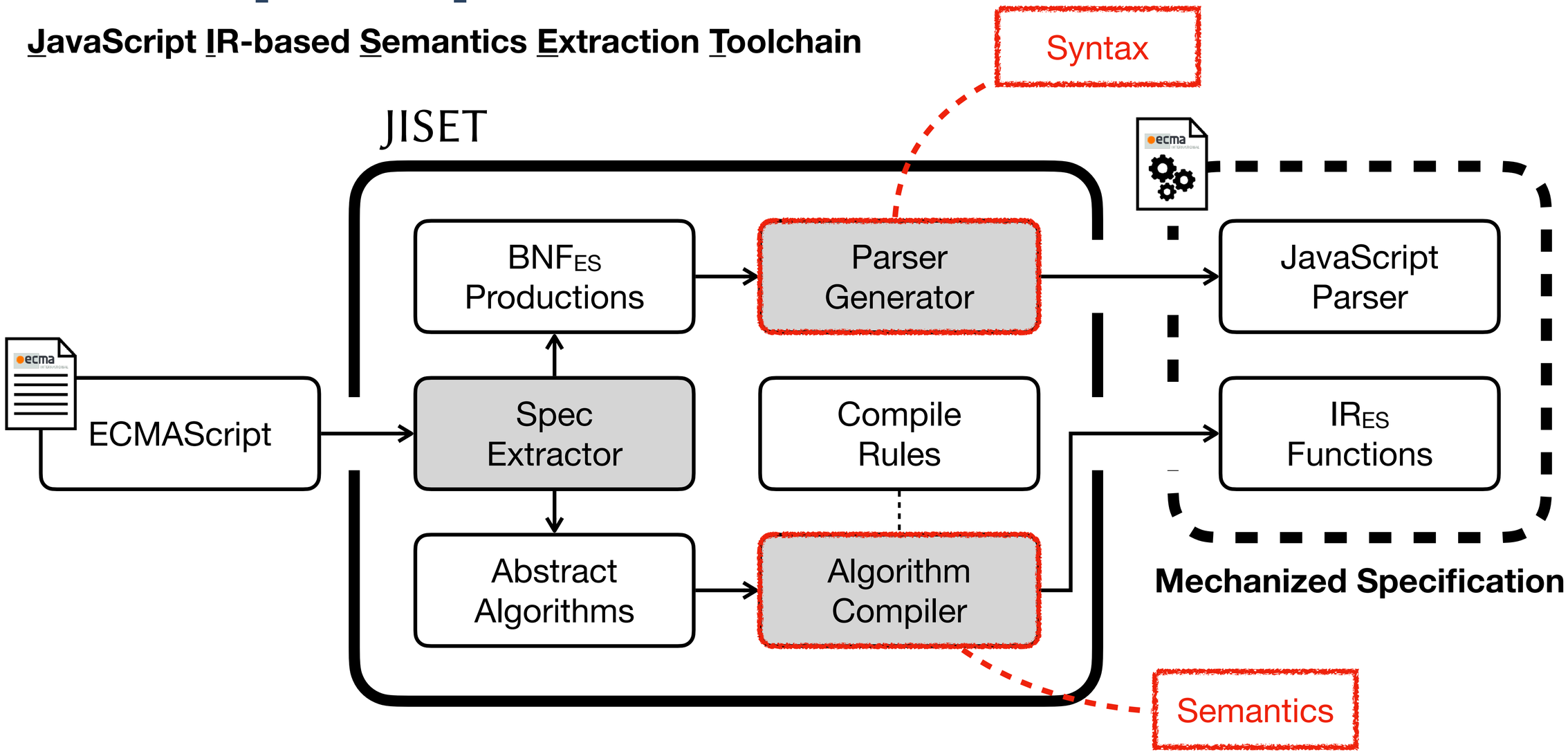


Overall Structure



JISET [ASE'20]

JavaScript IR-based Semantics Extraction Toolchain



JISET - Parser Generator (Syntax)

ArrayLiteral[Yield, Await] :

[*Elision*opt]

[*ElementList*[?Yield, ?Await]]

[*ElementList*[?Yield, ?Await] , *Elision*opt]

Parsing Expression Grammar
(+ Lookahed Parsing)

```
val ArrayLiteral: List[Boolean] => LAParser[T] = memo {
  case List(Yield, Await) =>
  "[" ~ opt(Elision) ~ "]"                ^^ ArrayLiteral0 |
  "[" ~ ElementList(Yield,Await) ~ "]" ^^ ArrayLiteral1 |
  "[" ~ ElementList(Yield,Await) ~ ","
      ~ opt(Elision)~ "]"                 ^^ ArrayLiteral2
}
```

JISET - Algorithm Compiler (Semantics)

12.2.5.3 Runtime Semantics: Evaluation

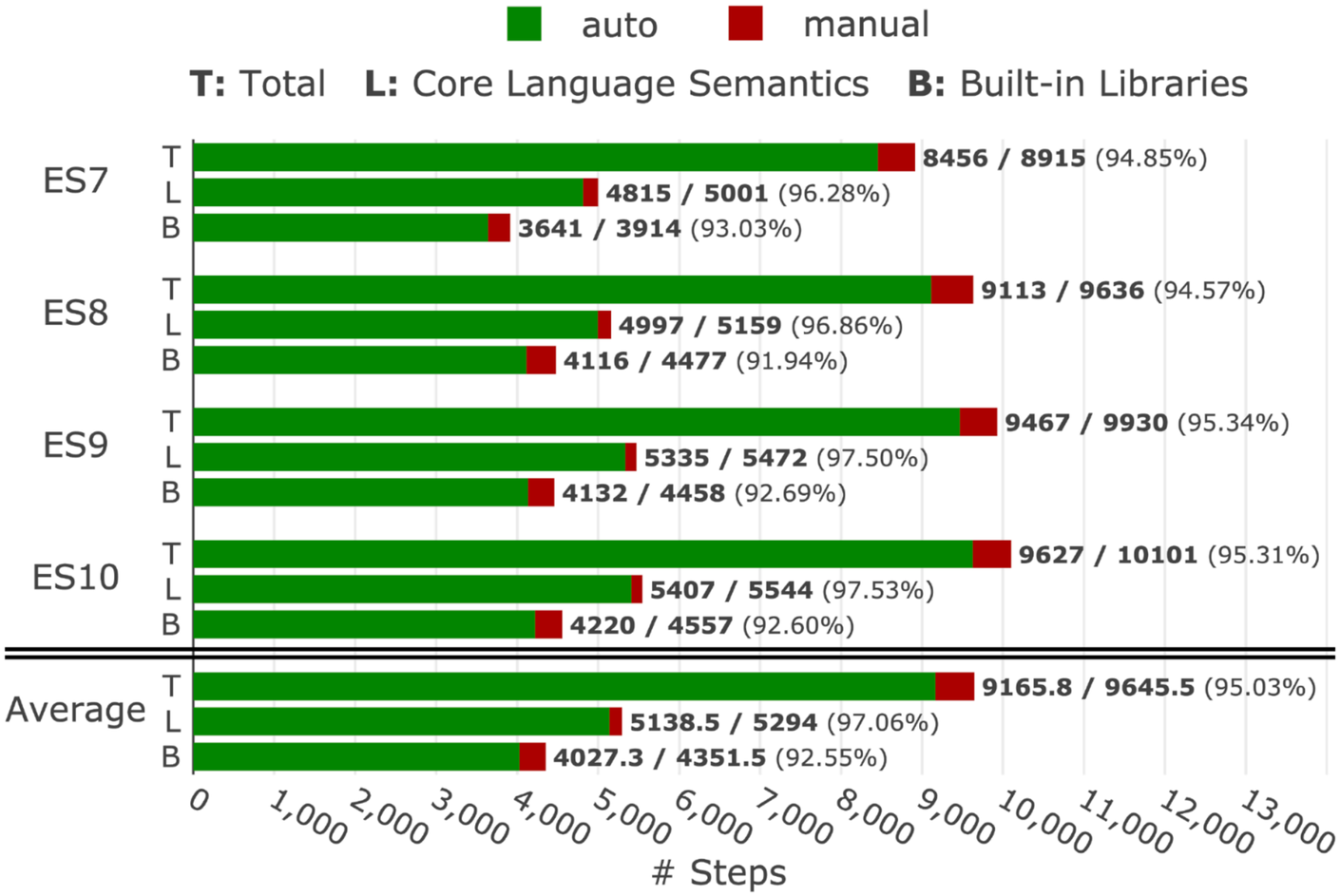
ArrayLiteral : [*Elision*]

1. Let *array* be ! ArrayCreate(0).
2. Let *pad* be the ElisionWidth of *Elision*; if *Elision* is not present, use the numeric value zero.
3. Perform Set(*array*, "length", ToUint32(*pad*), **false**).
4. NOTE: The above Set cannot fail because of the nature of the object returned by ArrayCreate.
5. Return *array*.

Compile Rules for Steps in Abstract Algorithms

```
ArrayLiteral[0].Evaluation (Elision) => {
  let array = [! (ArrayCreate 0)]
  if (! (= Elision absent)) {
    let len = (Elision.ArrayAccumulation array 0)
    [? len]
  }
  return array
}
```

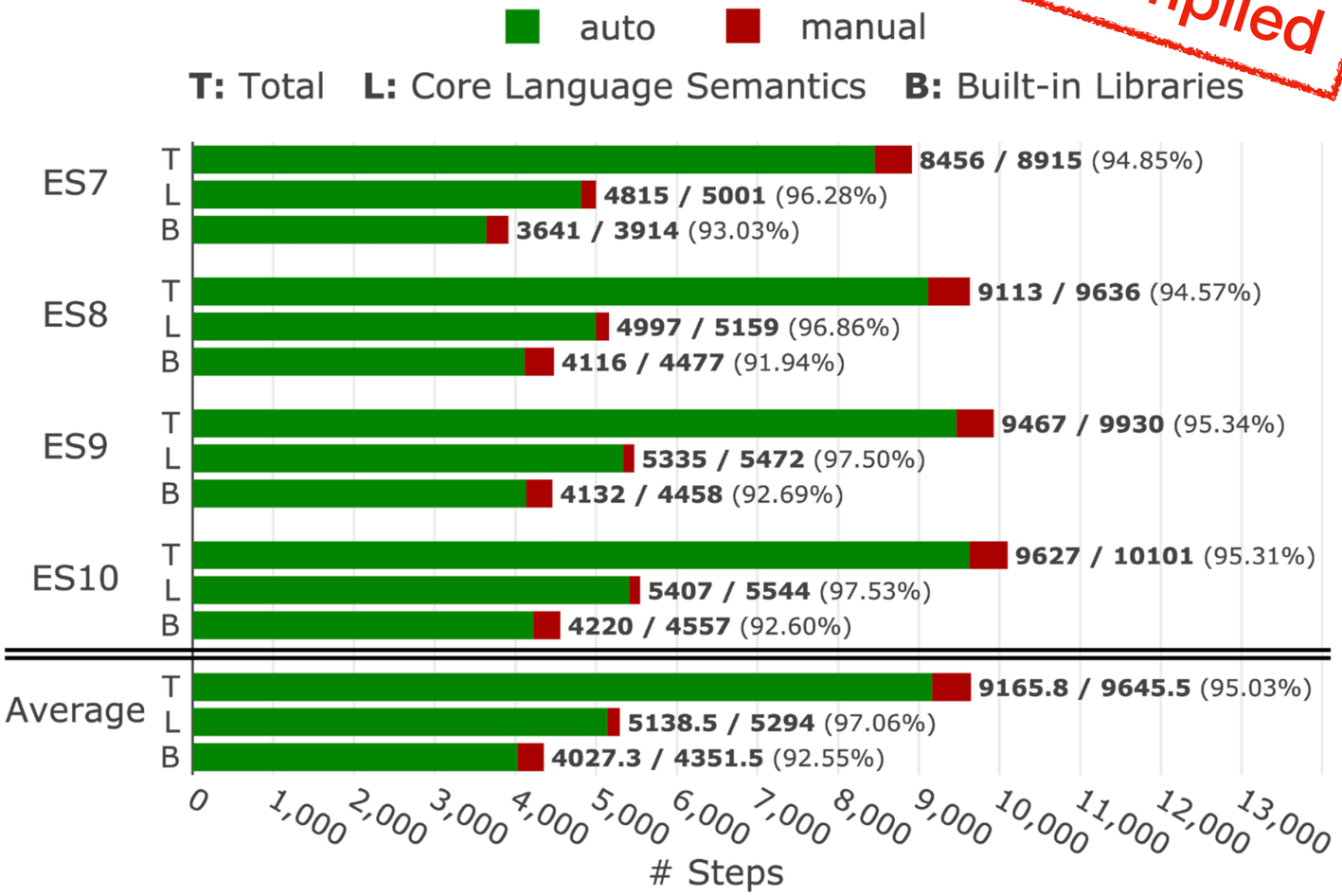
JISET - Evaluation



- **Test262**
  (Official Conformance Tests)
  - 18,064 applicable tests
- **Parsing tests**
  - Passed all 18,064 tests
- **Evaluation Tests**
  - Passed all 18,064 tests

JISET - Evaluation

≈ 95% Compiled

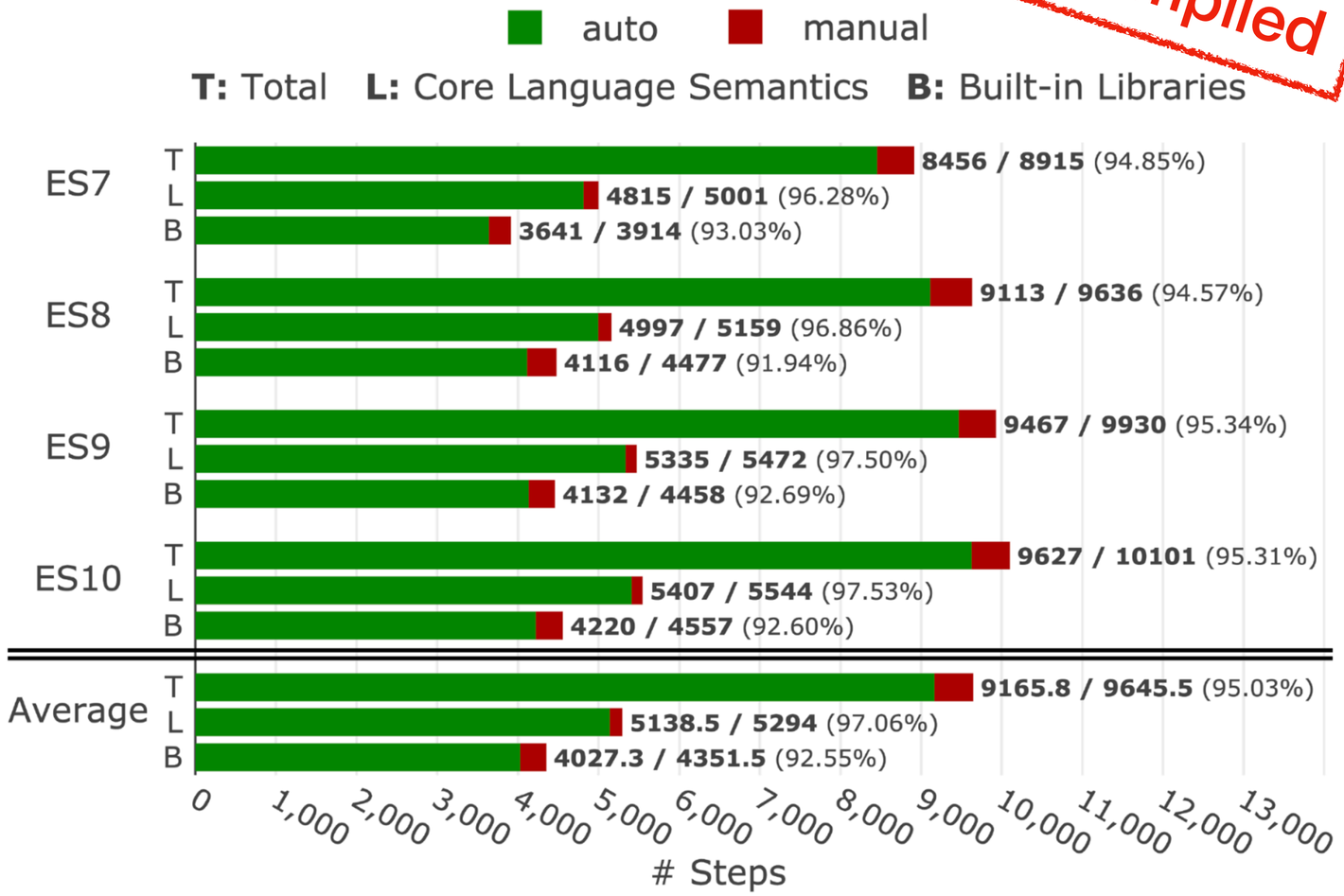


- **Test262**
  (Official Conformance Tests)
  - 18,064 applicable tests
- **Parsing tests**
  - Passed all 18,064 tests
- **Evaluation Tests**
  - Passed all 18,064 tests

JISET - Evaluation

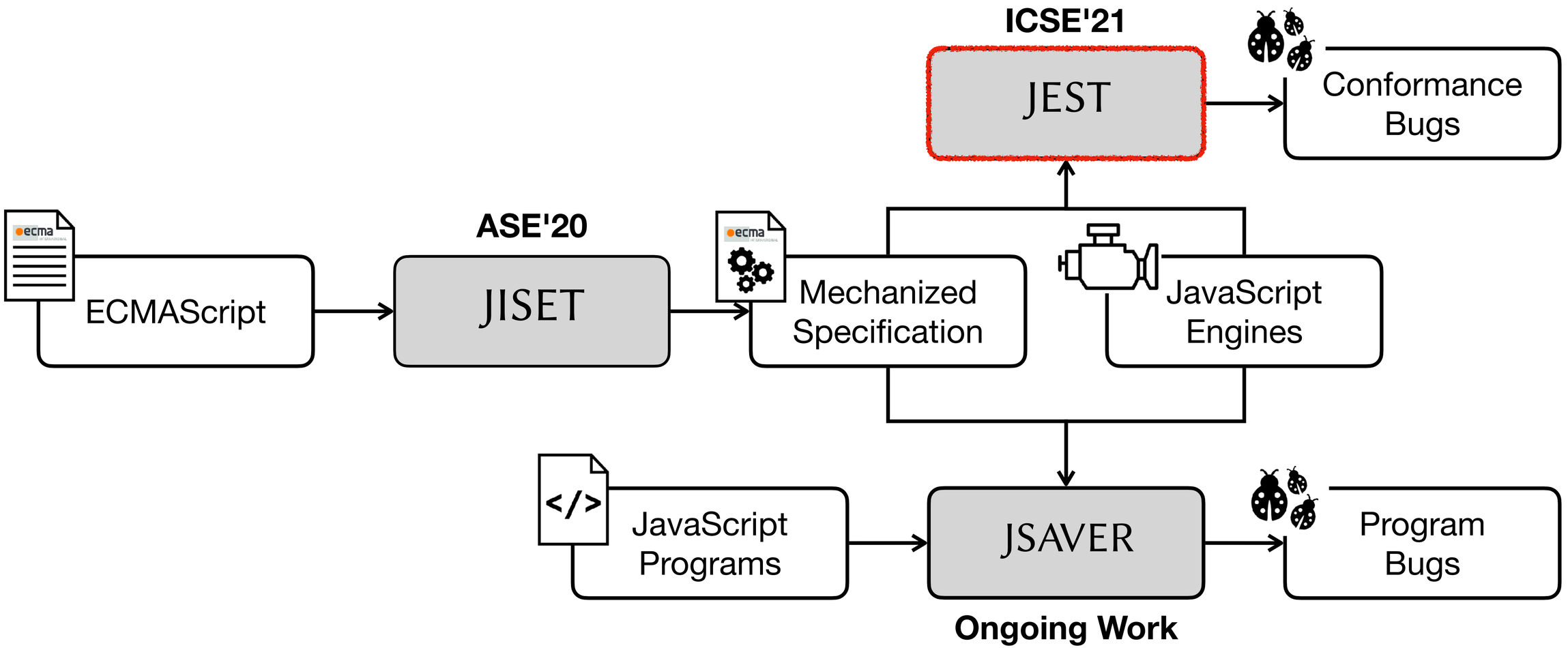
≈ 95% Compiled

Passed All Tests



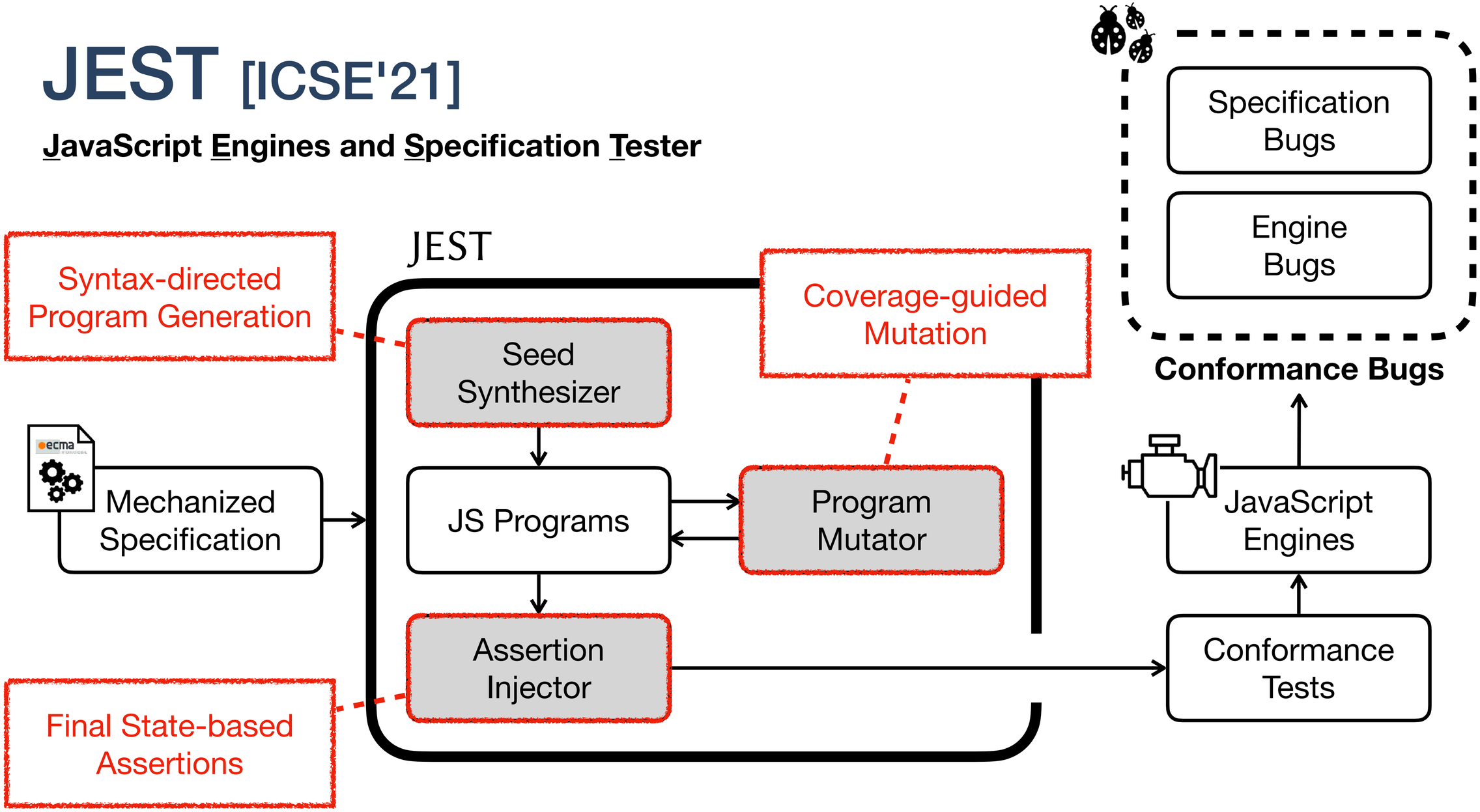
- **Test262**
  (Official Conformance Tests)
  - 18,064 applicable tests
- **Parsing tests**
  - Passed all 18,064 tests
- **Evaluation Tests**
  - Passed all 18,064 tests

Overall Structure



JEST [ICSE'21]

JavaScript Engines and Specification Tester



JEST - Test Synthesis

JavaScript Engines and Specification Tester

ArrayLiteral`[Yield, Await]` :

[*Elision*`opt`]

[*ElementList*`[?Yield, ?Await]`]

[*ElementList*`[?Yield, ?Await]` , *Elision*`opt`]

JEST - Test Synthesis

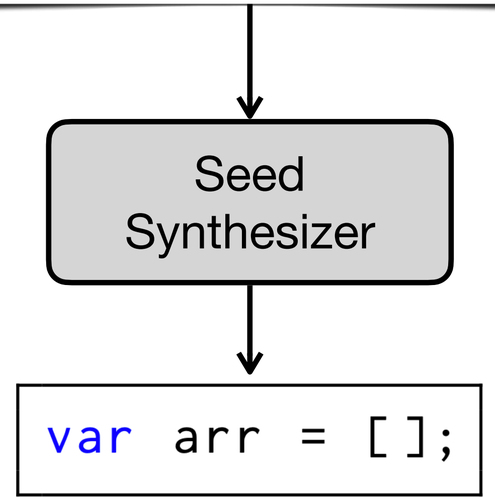
JavaScript Engines and Specification Tester

ArrayLiteral`[Yield, Await]` :

[*Elision*`opt`]

[*ElementList*`[?Yield, ?Await]`]

[*ElementList*`[?Yield, ?Await]` , *Elision*`opt`]



JEST - Test Synthesis

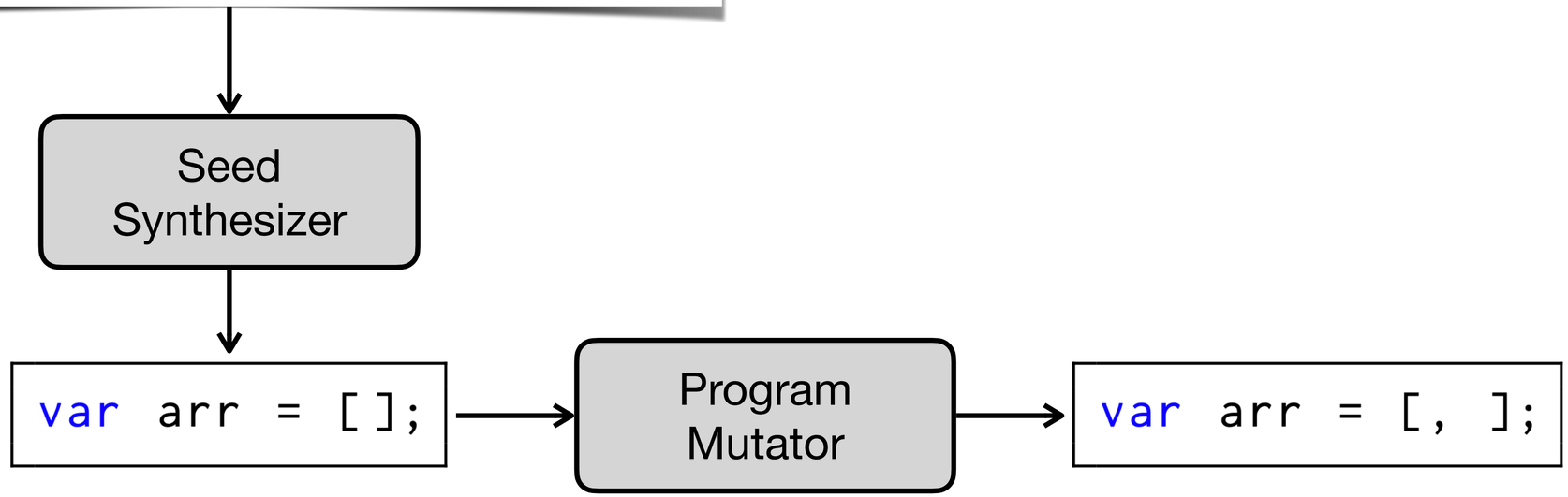
JavaScript Engines and Specification Tester

ArrayLiteral[Yield, Await] :

[*Elision*opt]

[*ElementList*[?Yield, ?Await]]

[*ElementList*[?Yield, ?Await] , *Elision*opt]



JEST - Test Synthesis

JavaScript Engines and Specification Tester

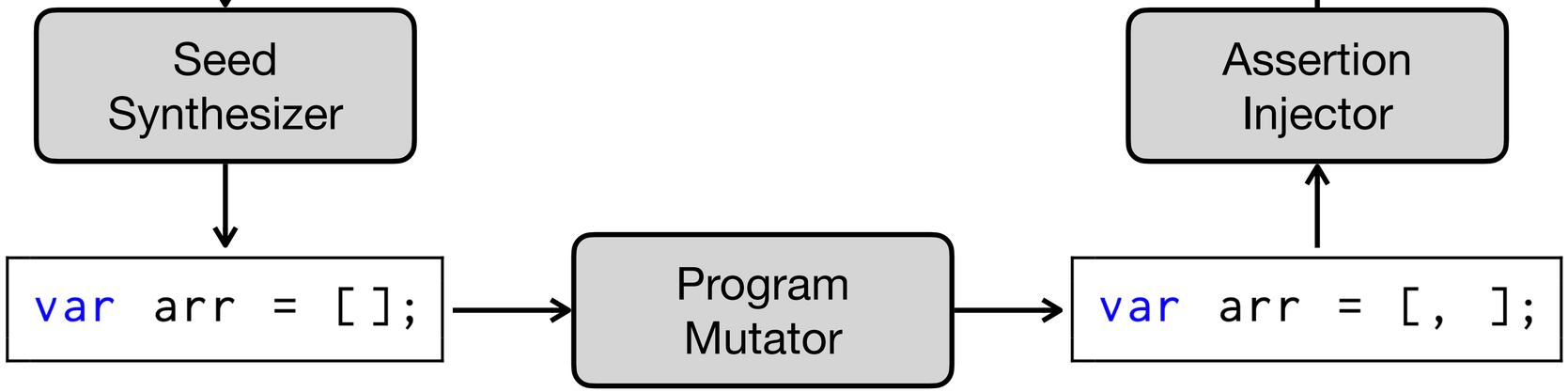
ArrayLiteral[Yield, Await] :

[*Elision*opt]

[*ElementList*[?Yield, ?Await]]

[*ElementList*[?Yield, ?Await] , *Elision*opt]

```
var arr = [, ];
$equal(Object.getPrototypeOf(arr),
  Array.prototype);
$equal(Object.isExtensible(arr), true);
$notCallable(arr);
$notConstructable(arr);
$compareArray(Reflect.ownKeys(arr),
  ['length'], arr);
$verifyProperty(arr, "length", {
  value      : 1,      writable     : true,
  enumerable: false, configurable: false
});
```



JEST - Test Synthesis

JavaScript Engines and Specification Tester

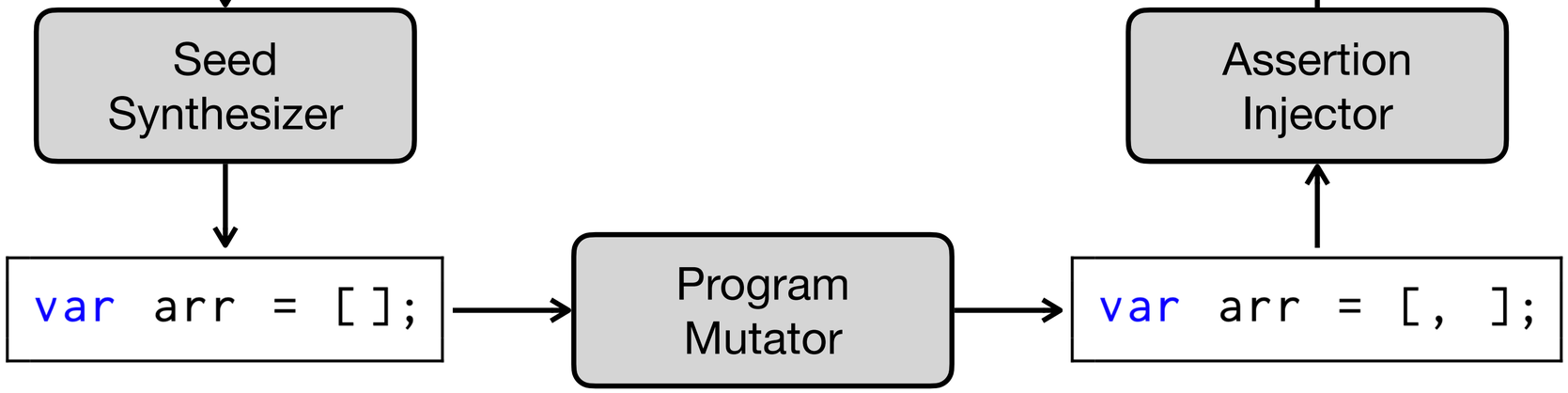
ArrayLiteral[Yield, Await] :

[*Elision*opt]

[*ElementList*[?Yield, ?Await]]

[*ElementList*[?Yield, ?Await] , *Elision*opt]

```
var arr = [, ];
$equal(Object.getPrototypeOf(arr),
  Array.prototype);
$equal(Object.isExtensible(arr), true);
$notCallable(arr);
$notConstructable(arr);
$compareArray(Reflect.ownKeys(arr),
  ['length'], arr);
$verifyProperty(arr, "length", {
  value      : 1,      writable     : true,
  enumerable: false, configurable: false
});
```



JEST - Test Synthesis

JavaScript Engines and Specification Tester

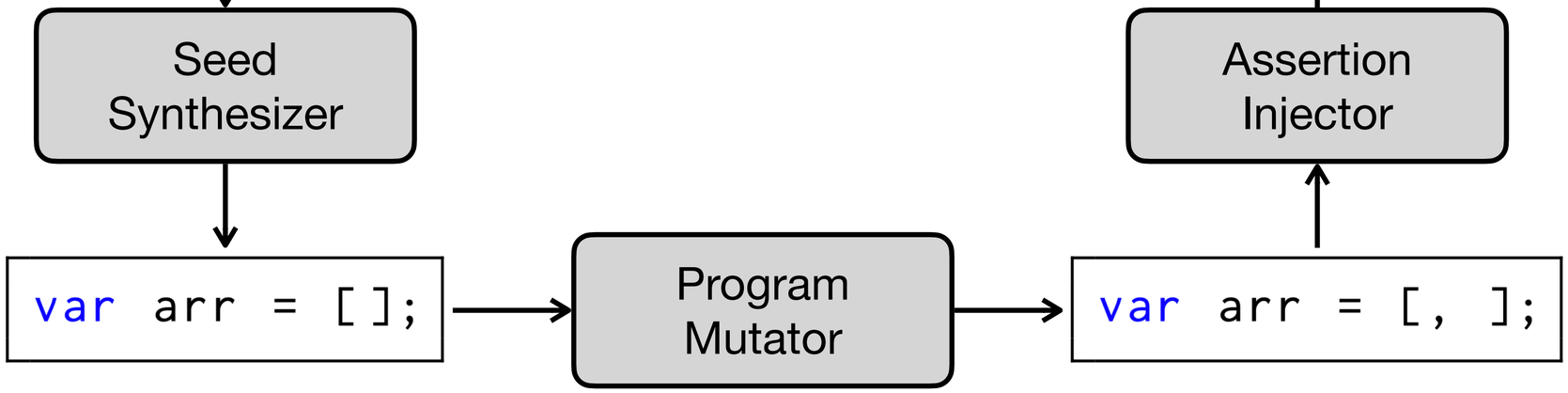
ArrayLiteral[Yield, Await] :

[*Elision*opt]

[*ElementList*[?Yield, ?Await]]

[*ElementList*[?Yield, ?Await] , *Elision*opt]

```
var arr = [, ];
$equal(Object.getPrototypeOf(arr),
  Array.prototype);
$equal(Object.isExtensible(arr), true);
$notCallable(arr);
$notConstructable(arr);
$compareArray(Reflect.ownKeys(arr),
  ['length'], arr);
$verifyProperty(arr, "length", {
  value      : 1,      writable     : true,
  enumerable: false, configurable: false
});
```



JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester

ECMAScript

Synthesize

`Test`

test

test

test

test

GraalVM

`QuickJS`

moddable

JavaScript Engines

JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester

ECMAScript

Synthesize

`Test`

test

test

test

test

JavaScript Engines

JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester

ECMAScript

Synthesize

Test

test

test

test

test

JavaScript Engines

An engine bug in

JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester

ECMAScript

Synthesize

`Test`

test

test

test

test

JavaScript Engines

JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester

ECMAScript

Synthesize

Test

test

test

test

test

GraalVM

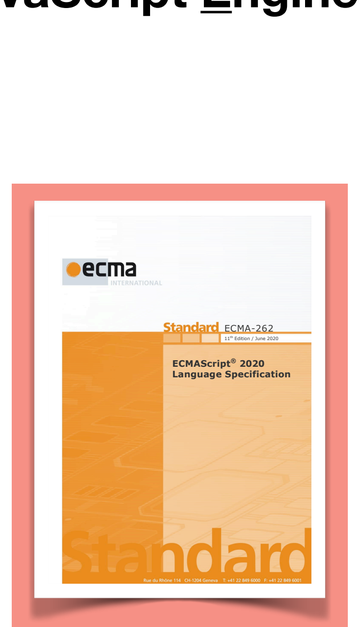
QuickJS

moddable

JavaScript Engines

JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester



ECMAScript

Synthesize

Test

test

test

test

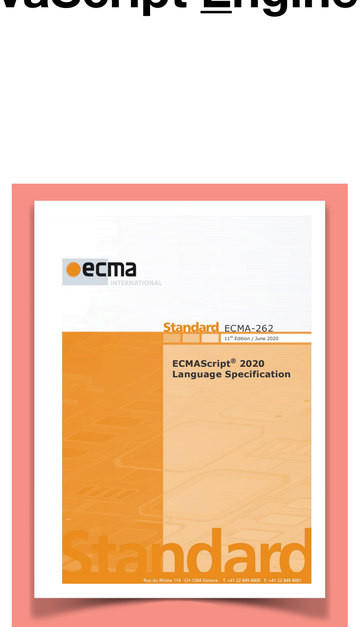
test

JavaScript Engines

A specification bug in ECMAScript

JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester



ECMAScript

Synthesize

Test

test

test

test

test

JavaScript Engines

A specification bug in ECMAScript

An engine bug in GraalVM

JEST - Evaluation

TABLE II: The number of engine bugs detected by JEST

Engines	Exc	Abort	Var	Obj	Desc	Key	In	Total
V8	0	0	0	0	0	2	0	2
GraalJS	6	0	0	0	2	8	0	16
QuickJS	3	0	1	0	0	2	0	6
Moddable XS	12	0	0	0	3	5	0	20
Total	21	0	1	0	5	17	0	44

TABLE III: Specification bugs in ECMAScript 2020 (ES11) detected by JEST

Name	Feature	#	Assertion	Known	Created	Resolved	Existed
ES11-1	Function	12	Key	O	2019-02-07	2020-04-11	429 days
ES11-2	Function	8	Key	O	2015-06-01	2020-04-11	1,776 days
ES11-3	Loop	1	Exc	O	2017-10-17	2020-04-30	926 days
ES11-4	Expression	4	Abort	O	2019-09-27	2020-04-23	209 days
ES11-5	Expression	1	Exc	O	2015-06-01	2020-04-28	1,793 days
ES11-6	Object	1	Exc	X	2019-02-07	2020-11-05	637 days

JEST - Evaluation

44 Bugs in Engines

TABLE II: The number of engine bugs detected by JEST

Engines	Exc	Abort	Var	Obj	Desc	Key	In	Total
V8	0	0	0	0	0	2	0	2
GraalJS	6	0	0	0	2	8	0	16
QuickJS	3	0	1	0	0	2	0	6
Moddable XS	12	0	0	0	3	5	0	20
Total	21	0	1	0	5	17	0	44

TABLE III: Specification bugs in ECMAScript 2020 (ES11) detected by JEST

Name	Feature	#	Assertion	Known	Created	Resolved	Existed
ES11-1	Function	12	Key	O	2019-02-07	2020-04-11	429 days
ES11-2	Function	8	Key	O	2015-06-01	2020-04-11	1,776 days
ES11-3	Loop	1	Exc	O	2017-10-17	2020-04-30	926 days
ES11-4	Expression	4	Abort	O	2019-09-27	2020-04-23	209 days
ES11-5	Expression	1	Exc	O	2015-06-01	2020-04-28	1,793 days
ES11-6	Object	1	Exc	X	2019-02-07	2020-11-05	637 days

JEST - Evaluation

44 Bugs in Engines

TABLE II: The number of engine bugs detected by JEST

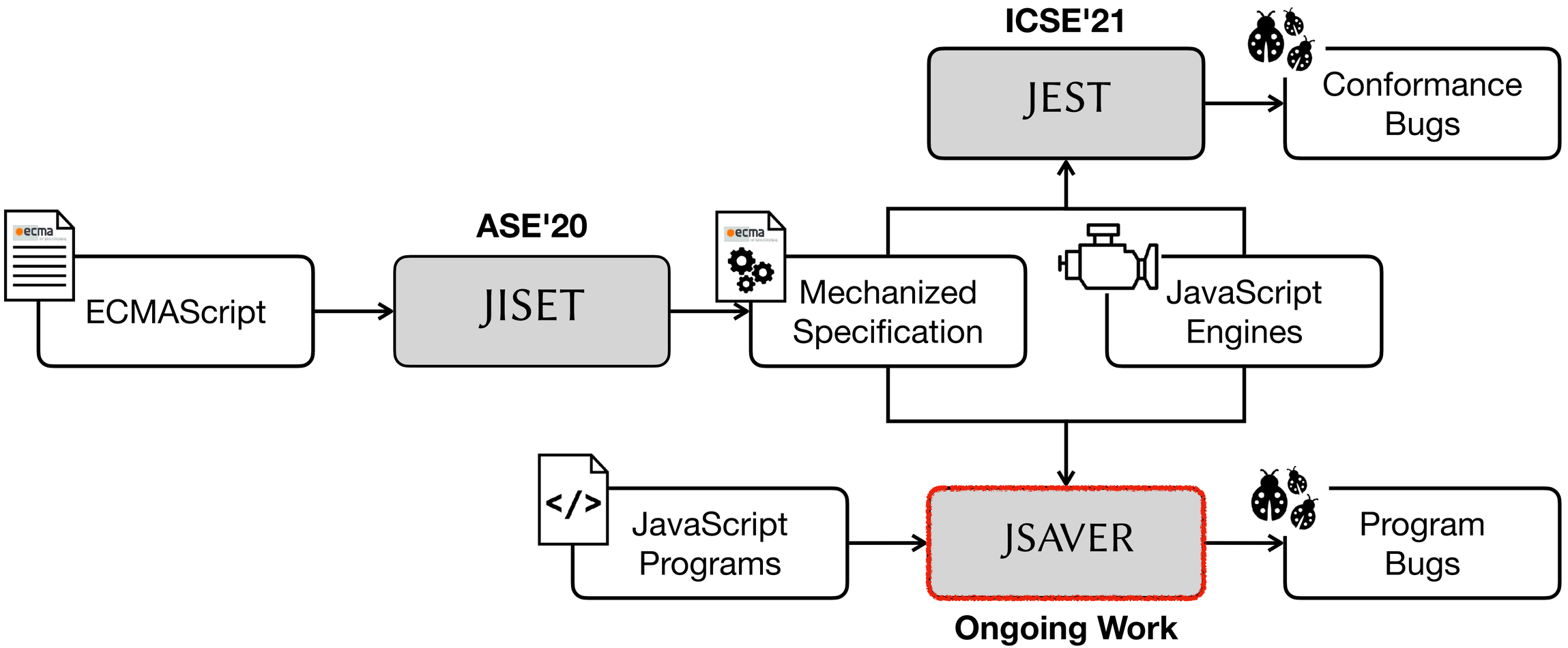
Engines	Exc	Abort	Var	Obj	Desc	Key	In	Total
V8	0	0	0	0	0	2	0	2
GraalJS	6	0	0	0	2	8	0	16
QuickJS	3	0	1	0	0	2	0	6
Moddable XS	12	0	0	0	3	5	0	20
Total	21	0	1	0	5	17	0	44

27 Bugs in Spec.

TABLE III: Specification bugs in ECMAScript 2020 (ES11) detected by JEST

Name	Feature	#	Assertion	Known	Created	Resolved	Existed
ES11-1	Function	12	Key	O	2019-02-07	2020-04-11	429 days
ES11-2	Function	8	Key	O	2015-06-01	2020-04-11	1,776 days
ES11-3	Loop	1	Exc	O	2017-10-17	2020-04-30	926 days
ES11-4	Expression	4	Abort	O	2019-09-27	2020-04-23	209 days
ES11-5	Expression	1	Exc	O	2015-06-01	2020-04-28	1,793 days
ES11-6	Object	1	Exc	X	2019-02-07	2020-11-05	637 days

Overall Structure



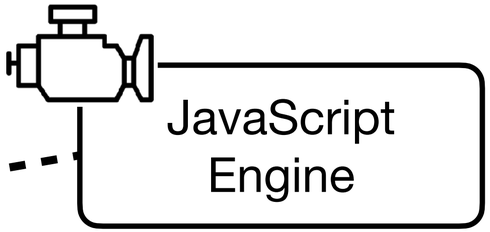
JSAVER - Basic Idea

JavaScript Static Analysis via ECMAScript Representation

```
(1) class A extends (2) foo.Component {
      constructor(...args) { /* ... */ }
      $AccountRecoveryModal1() { /* ... */ }
      render() {
    (3) const { optionsList: list, title: title } =
            this.props.options;  (4)
        /* ...*/
      }
}
```

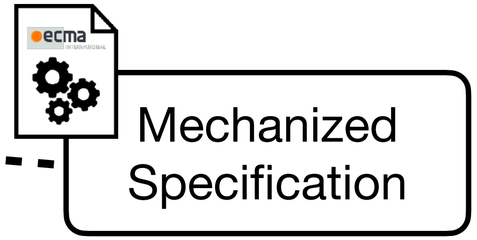
LandingPage.js file in the Instagram Website.

New language features in >= ES6




```
// a function call with concrete values
var a1 = new A(1, 2, 3);

// a function call with abstract values
var x = Math.random();
var a2 = new A(x);
```

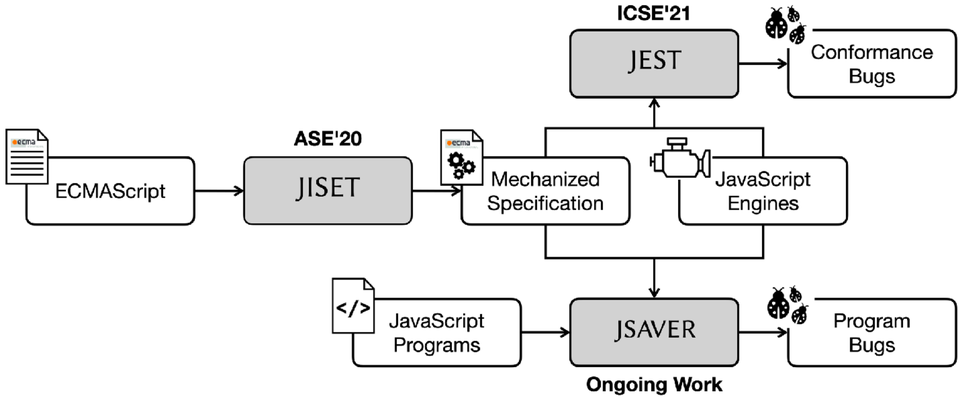


JSAVER - Evaluation Plan

JavaScript Static Analysis via ECMAScript Representation

- **Static analysis for real-world applications**
  - Web applications using new language features >= ES6
- **Research Questions**
  - Analysis Scope
  - Performance
  - Precision of Bug Detection

Overall Structure



JISET - Algorithm Compiler (Semantics)

12.2.5.3 Runtime Semantics: Evaluation
ArrayLiteral : [*Elision*]

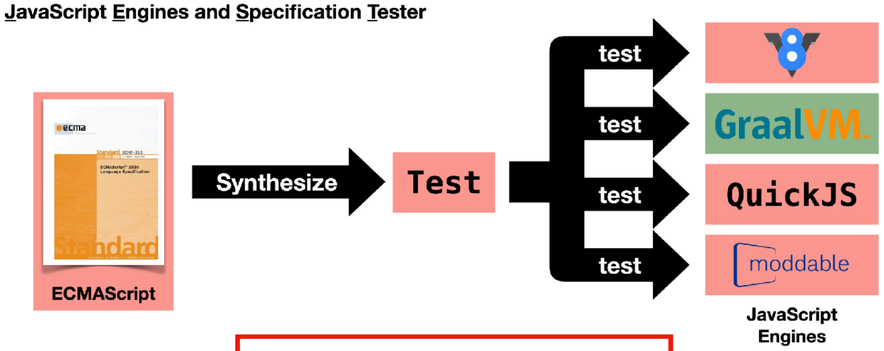
1. Let *array* be ! ArrayCreate(0).
2. Let *pad* be the ElisionWidth of *Elision*; if *Elision* is not present, use the numeric value zero.
3. Perform Set(*array*, "length", ToUint32(*pad*), **false**).
4. NOTE: The above Set cannot fail because of the nature of the object returned by ArrayCreate.
5. Return *array*.

Compile Rules for Steps in Abstract Algorithms

```
ArrayLiteral[0].Evaluation (Elision) => {
  let array = [! (ArrayCreate 0)]
  if (! (= Elision absent)) {
    let len = (Elision.ArrayAccumulation array 0)
    [? len]
  }
  return array
}
```


JEST - N+1-version Differential Testing

JavaScript Engines and Specification Tester



A specification bug in ECMAScript
An engine bug in GraalVM

JSAVER - Basic Idea

JavaScript Static Analysis via ECMAScript Representation

```
(1) class A extends foo.Component {                    (2)
      constructor(...args) { /* ... */ }
      $AccountRecoveryModal1() { /* ... */ }
      render() {
(3)     const { optionsList: list, title: title } =
          this.props.options;                          (4)
        /* ...*/
      }
    }
```

LandingPage.js file in the Instagram Website.

New language features in >= ES6

```
// a function call with concrete values
var a1 = new A(1, 2, 3);

// a function call with abstract values
var x = Math.random();
var a2 = new A(x);
```

JavaScript Engine

Mechanized Specification