Feature-Sensitive Coverage for Conformance Testing of Programming Language Implementations

Jihyeok Park [1], Dongjun Youn [2], Kanguk Lee [2], and Sukyoung Ryu [2]

[1] KOREA UNIVERSITY [2] KAIST

June 21, 2023

PLRG

PLDI Orlando 2023

Feature-Sensitive Coverage for Conformance Testing of Programming Language Implementations

Background

Jihyeok Park [1], Dongjun Youn [2], Kanguk Lee [2], and Sukyoung Ryu [2]

[1] KOREA UNIVERSITY

[2] KAIST

June 21, 2023

PLRG

PLDI Orlando 2023

Our Idea

Feature-Sensitive Coverage for Conformance Testing of Programming Language Implementations

Background

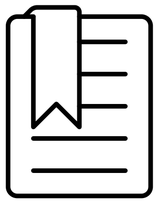
Jihyeok Park [1], Dongjun Youn [2], Kanguk Lee [2], and Sukyoung Ryu [2]

[1] KOREA UNIVERSITY

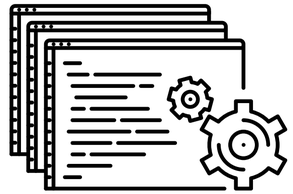
[2] KAIST

June 21, 2023

Conformance Testing of PL Implementations



Specification
of **L_1**



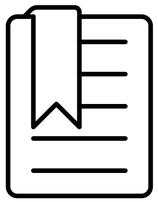
Implementations
of **L_1**

Programming Language **L_1**

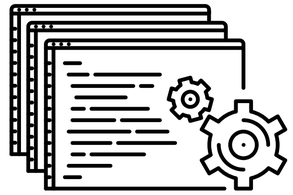
Conformance Testing of PL Implementations

$A ::= BC$
Syntax

$[\![-]\!]$
Semantics



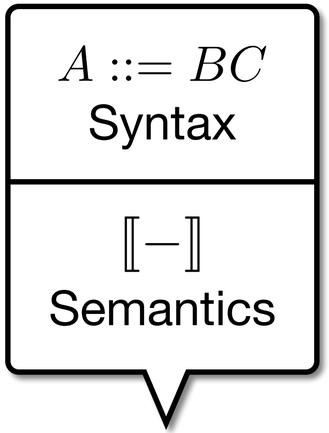
Specification
of **L_1**

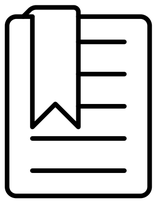


Implementations
of **L_1**

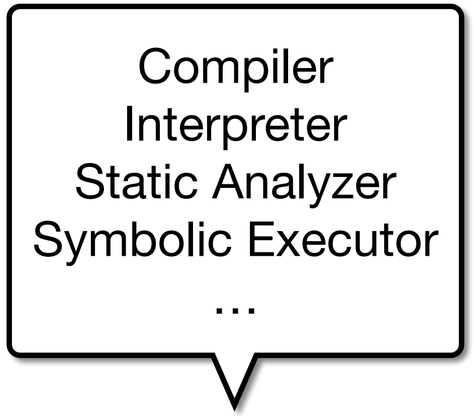
Programming Language **L_1**

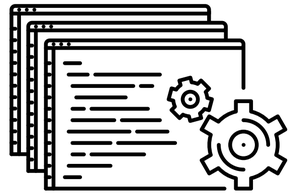
Conformance Testing of PL Implementations





Specification
of **L_1**

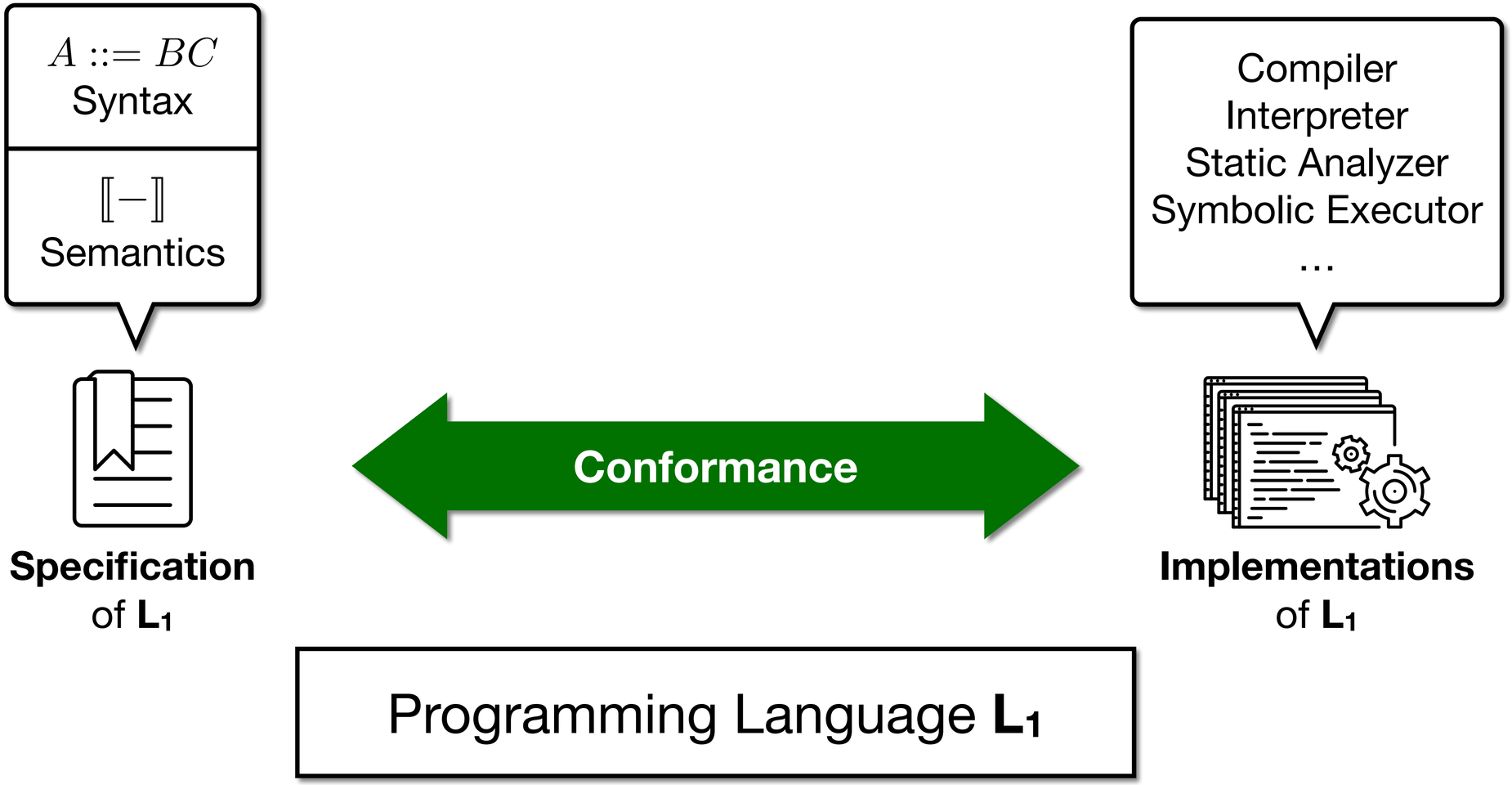




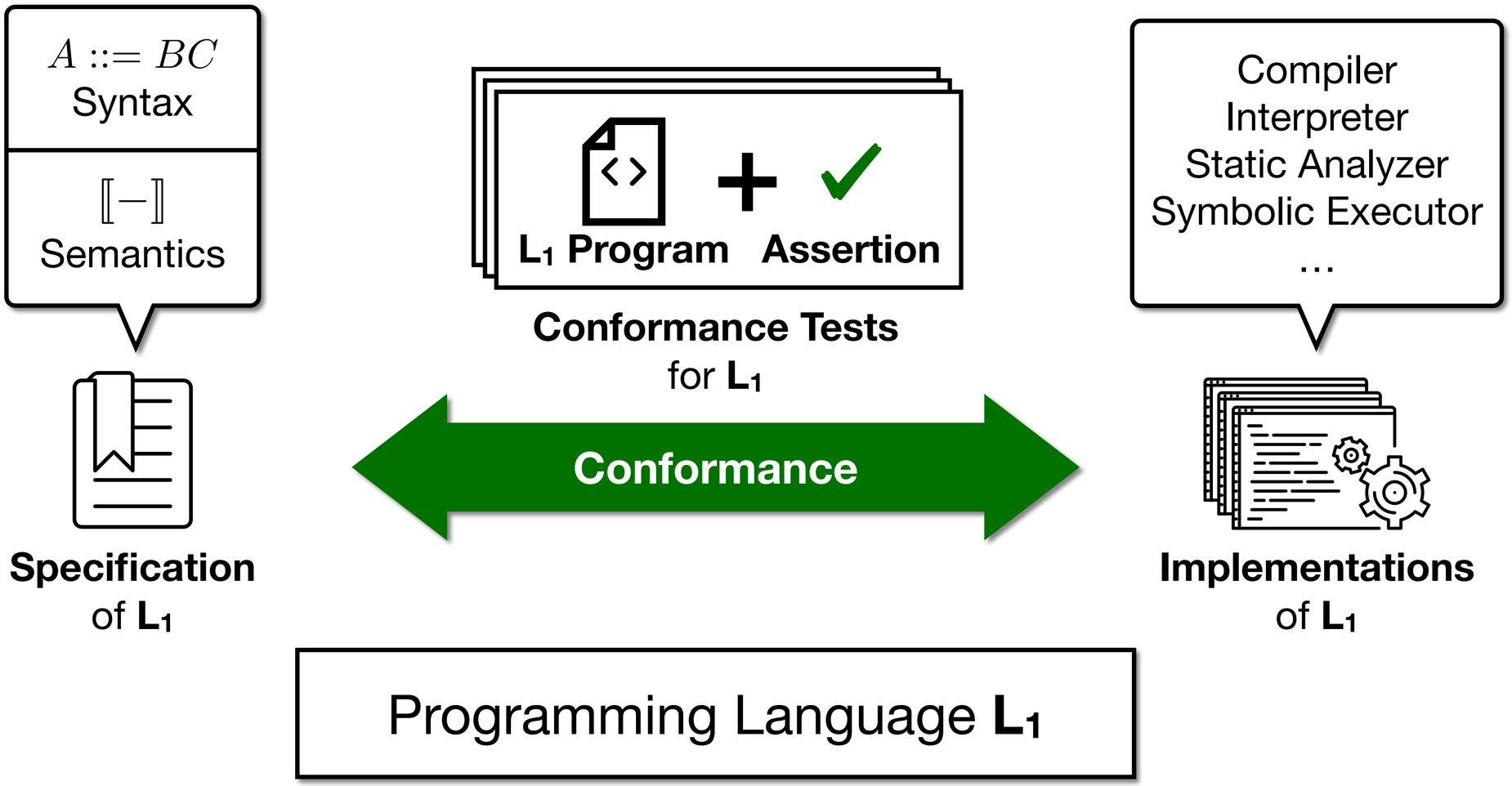
Implementations
of **L_1**

Programming Language **L_1**

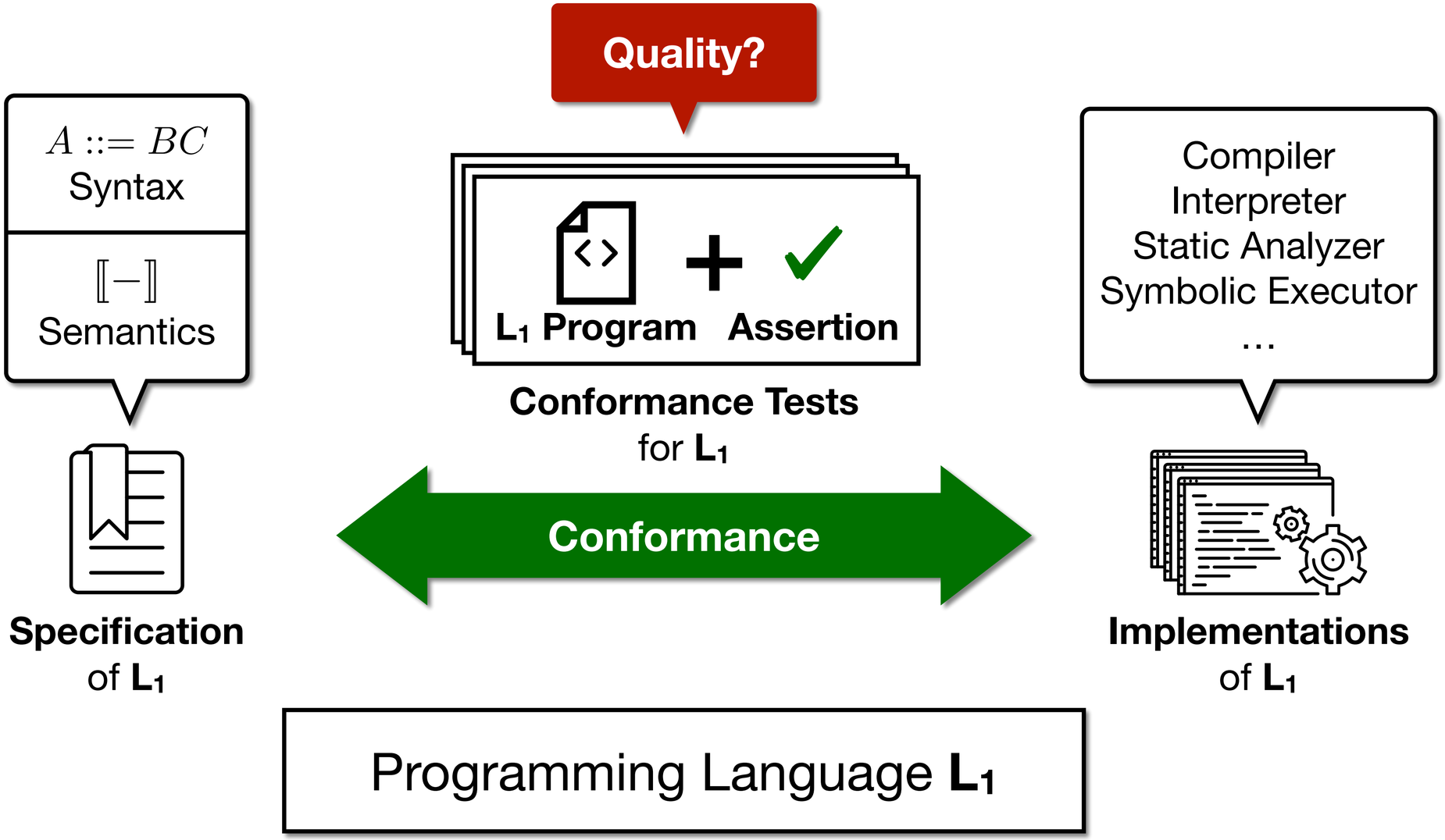
Conformance Testing of PL Implementations



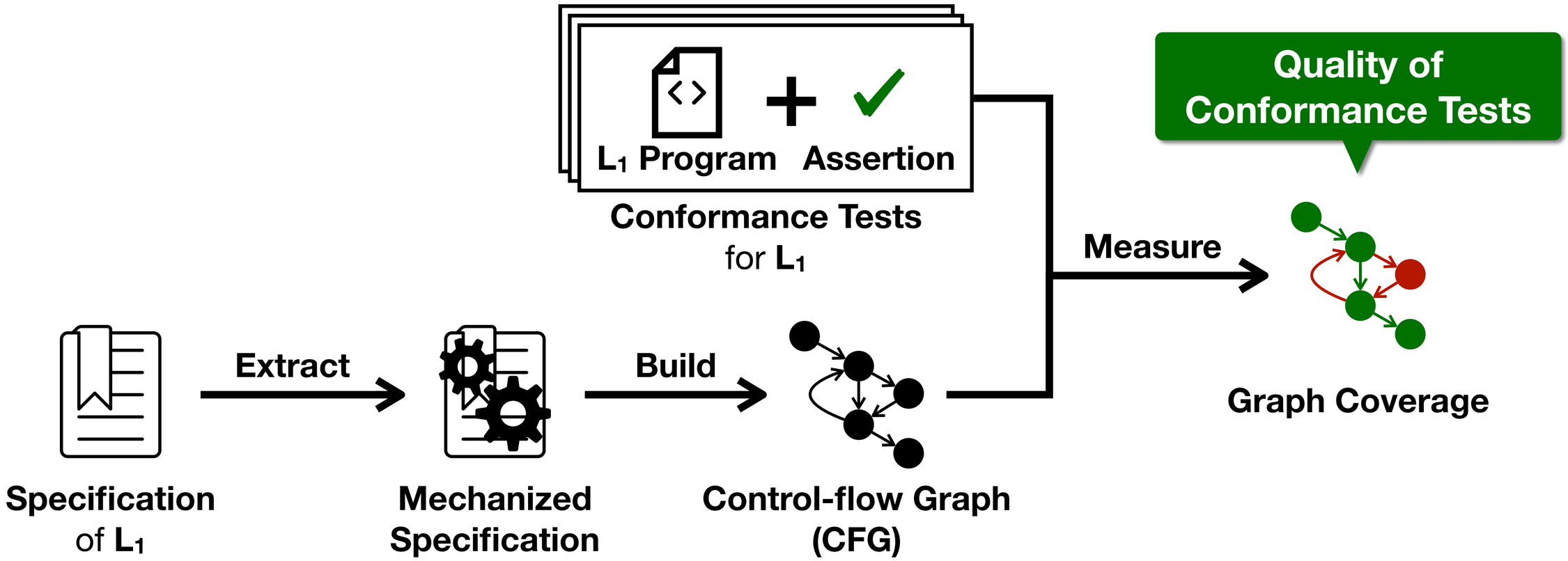
Conformance Testing of PL Implementations



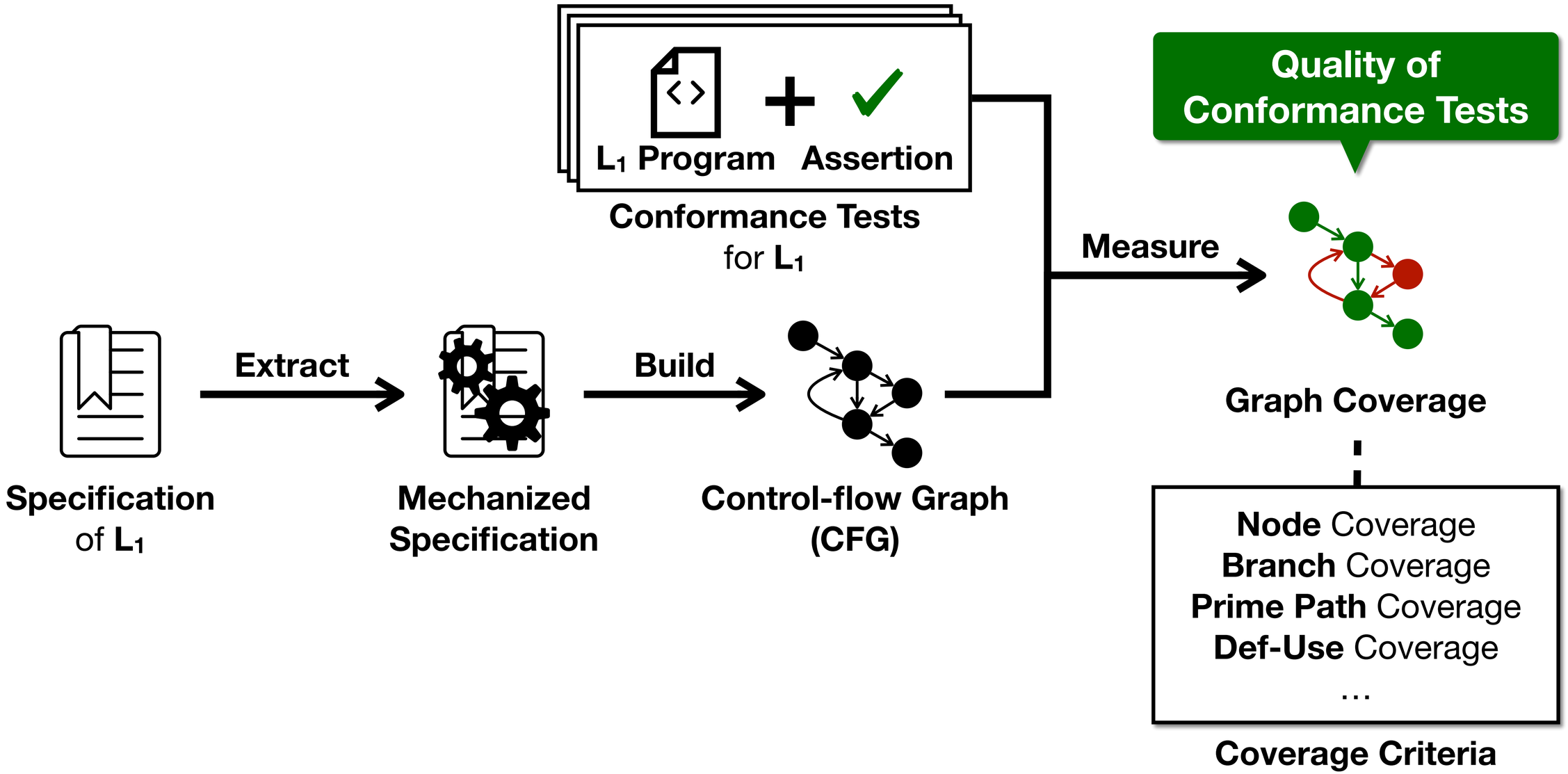
Conformance Testing of PL Implementations



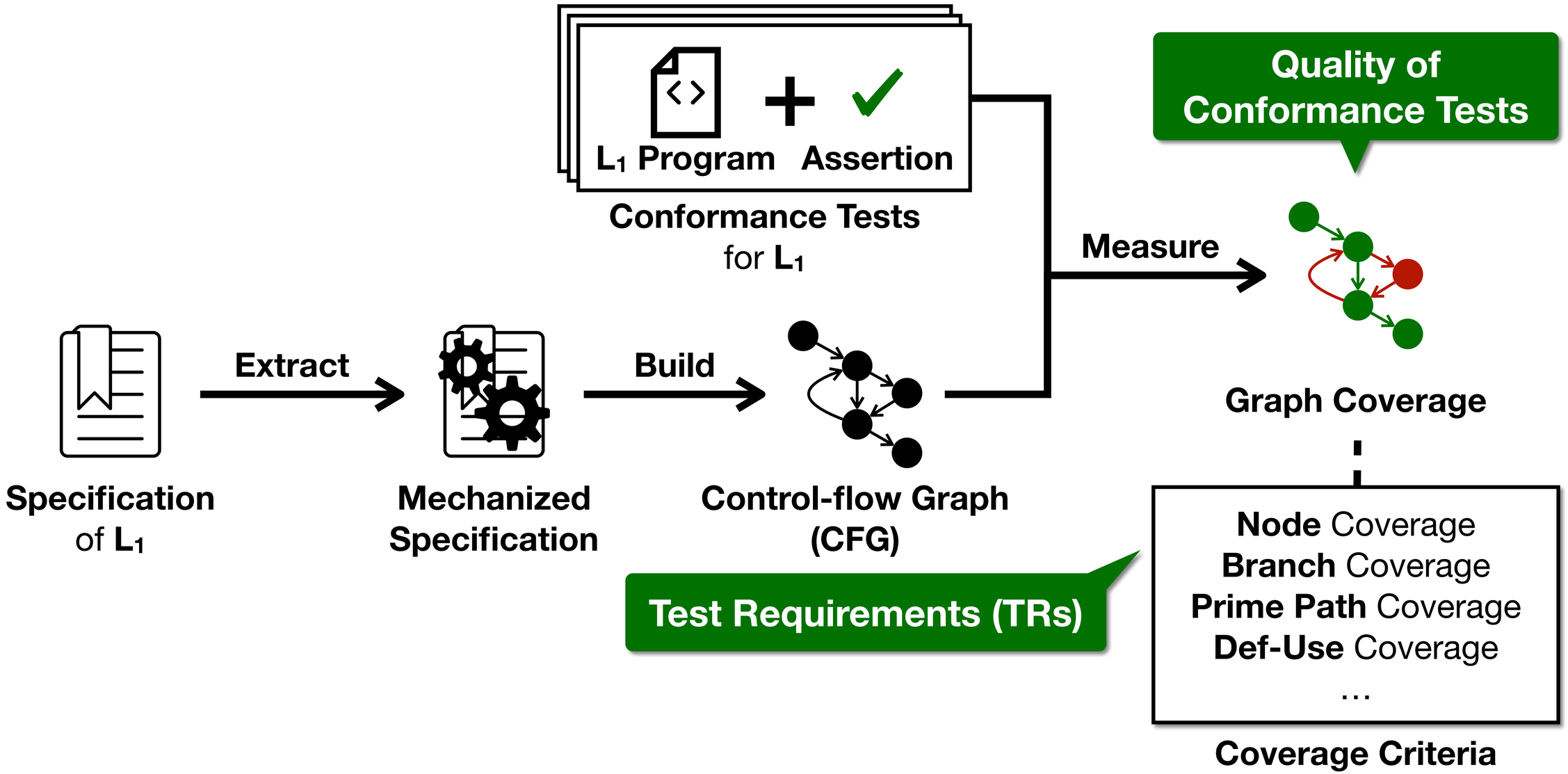
Graph Coverage for Language Specification



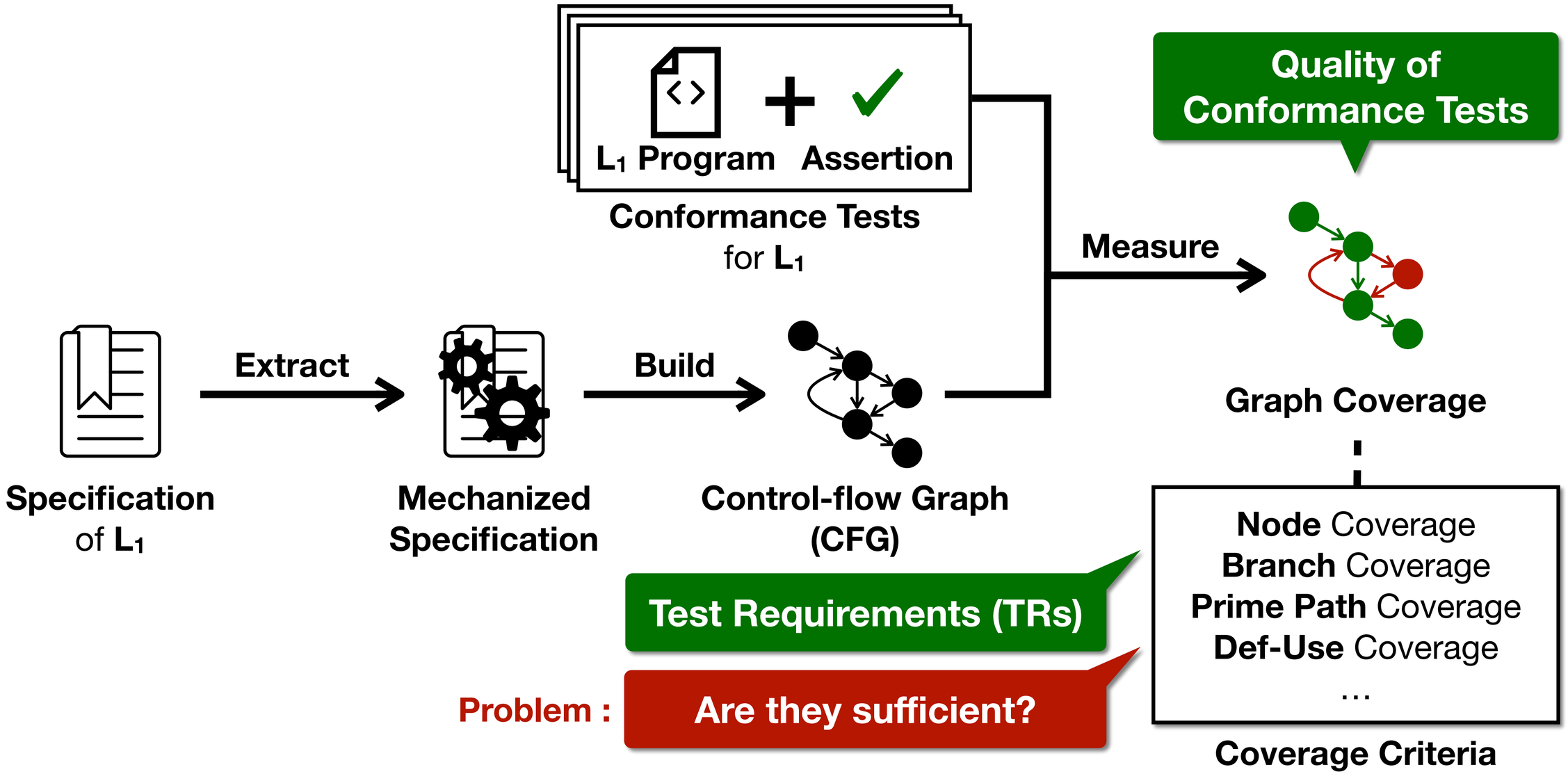
Graph Coverage for Language Specification



Graph Coverage for Language Specification

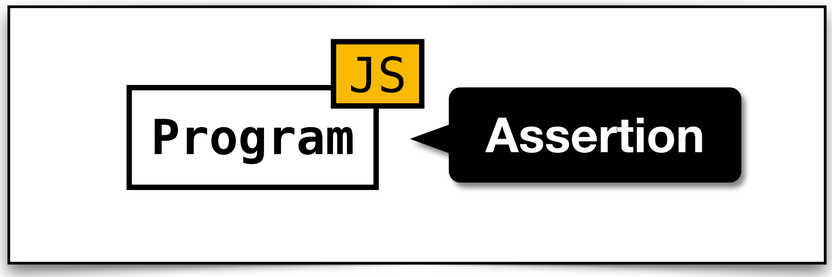


Graph Coverage for Language Specification



Motivating Example — JavaScript

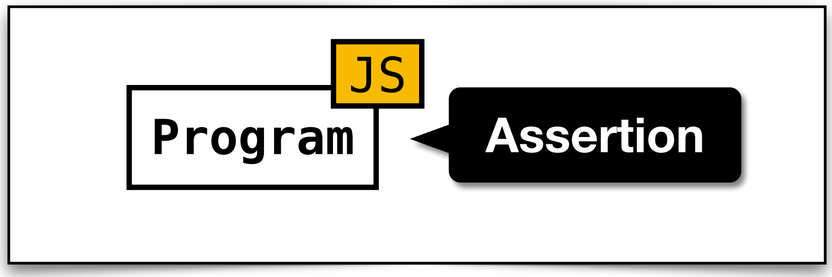
- JavaScript is a **dynamically-typed language** with **complex semantics**
- It is not easy to understand even simple **addition/subtraction operations**.



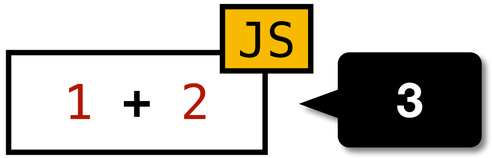
Conformance Test for **JS**

Motivating Example — JavaScript

- JavaScript is a **dynamically-typed language** with **complex semantics**
- It is not easy to understand even simple **addition/subtraction operations**.

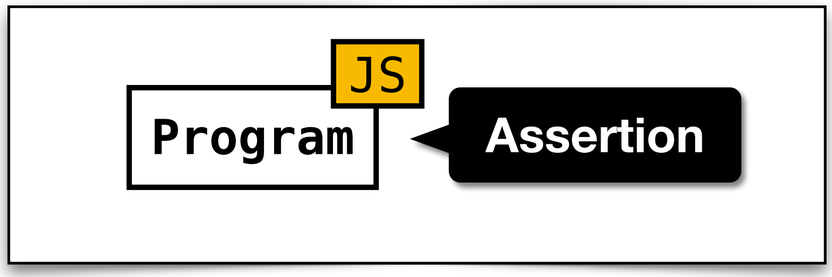


Conformance Test for **JS**

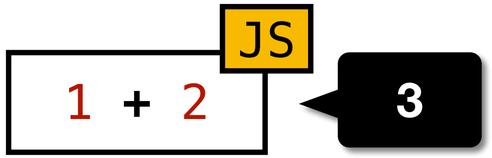


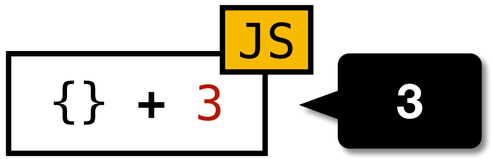
Motivating Example — JavaScript

- JavaScript is a **dynamically-typed language** with **complex semantics**
- It is not easy to understand even simple **addition/subtraction operations**.



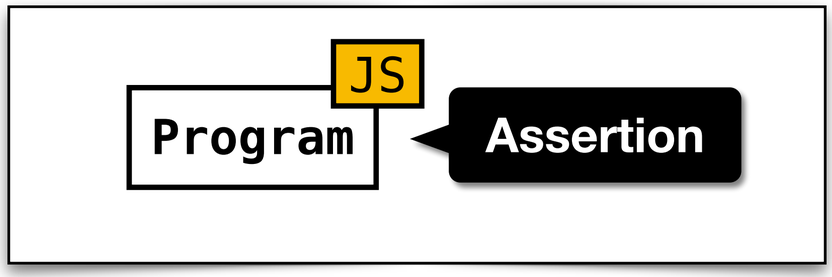
Conformance Test for **JS**



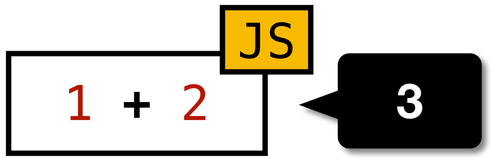


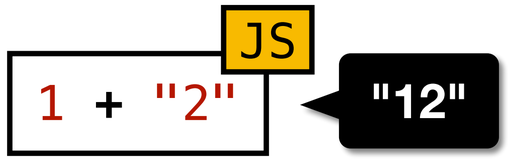
Motivating Example — JavaScript

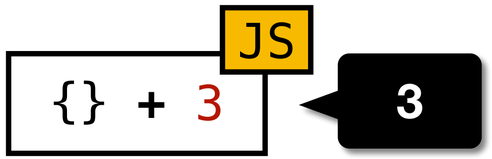
- JavaScript is a **dynamically-typed language** with **complex semantics**
- It is not easy to understand even simple **addition/subtraction operations**.

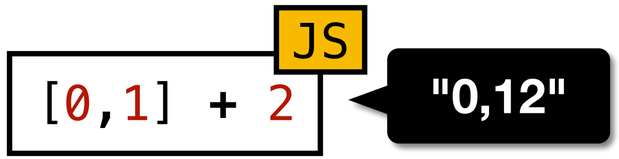


Conformance Test for **JS**



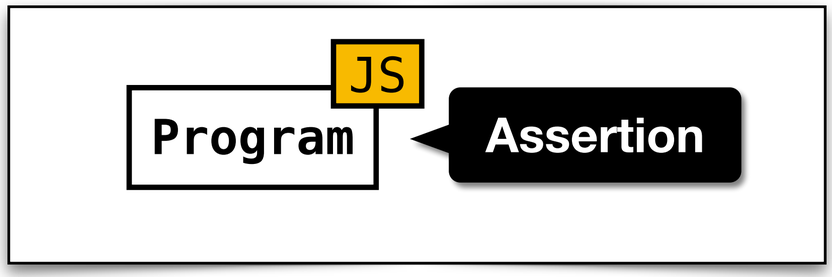




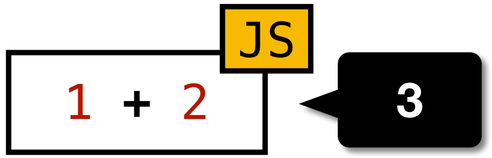


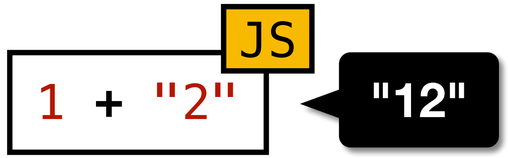
Motivating Example — JavaScript

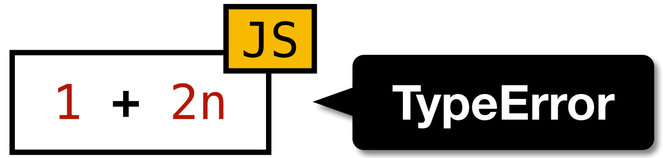
- JavaScript is a **dynamically-typed language** with **complex semantics**
- It is not easy to understand even simple **addition/subtraction operations**.

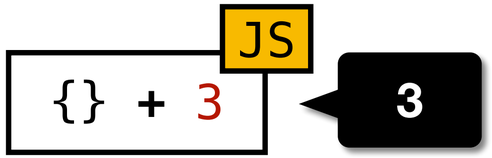


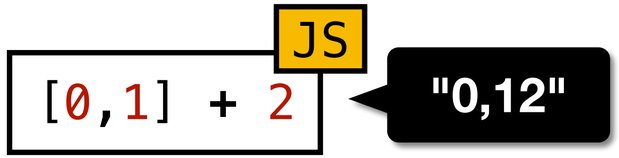
Conformance Test for **JS**

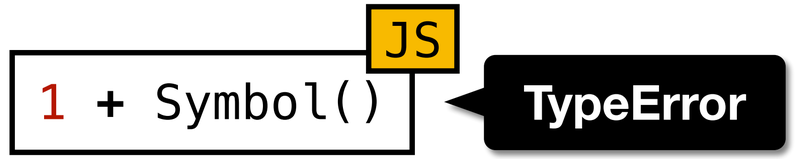






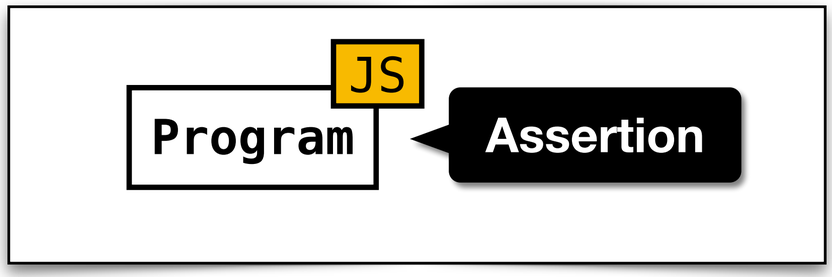




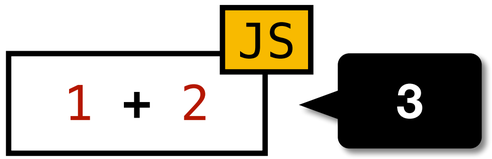


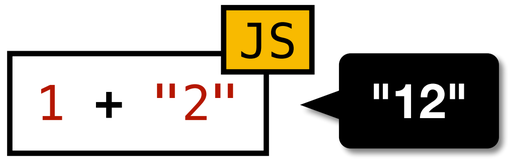
Motivating Example — JavaScript

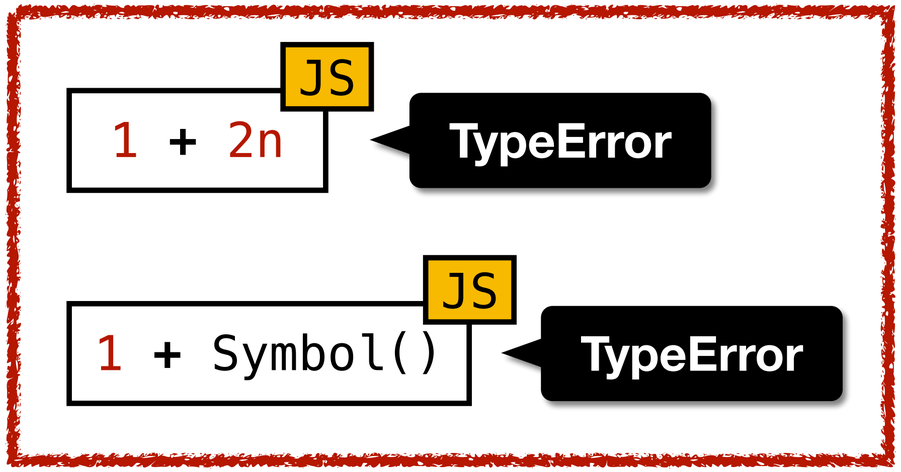
- JavaScript is a **dynamically-typed language** with **complex semantics**
- It is not easy to understand even simple **addition/subtraction operations**.

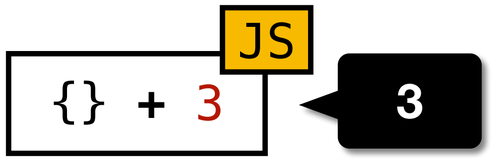


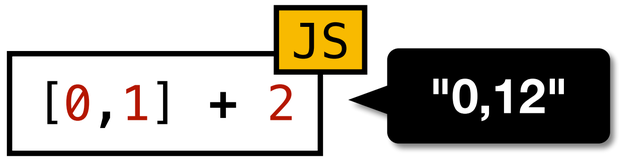
Conformance Test for **JS**



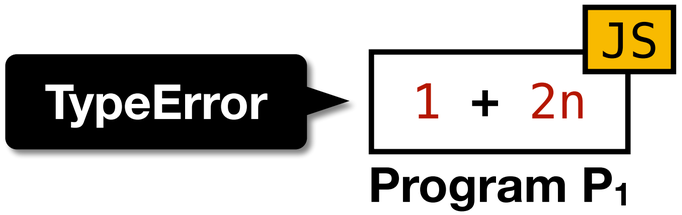




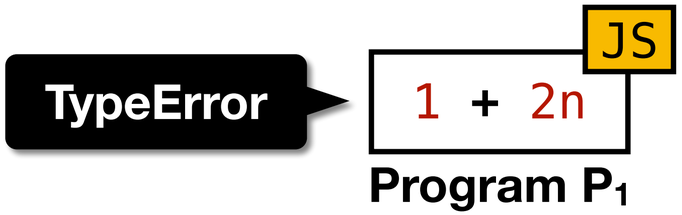




Motivating Example 1



Motivating Example 1



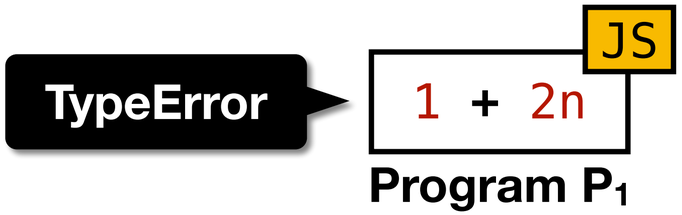
feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1



feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

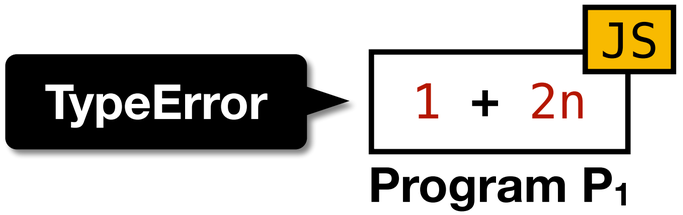
EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

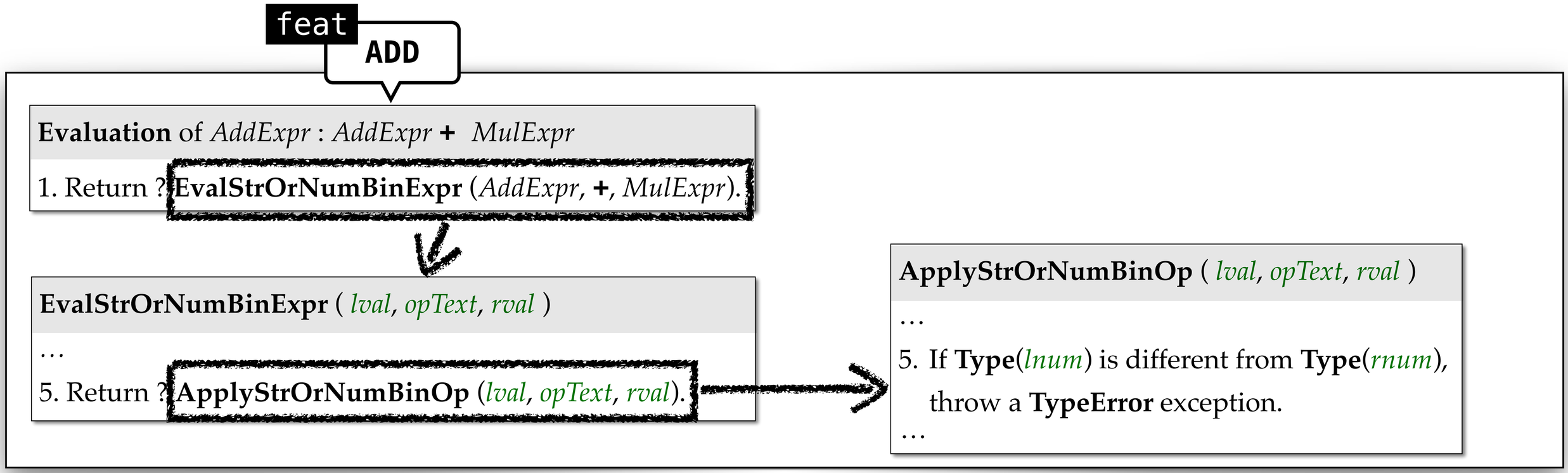
…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

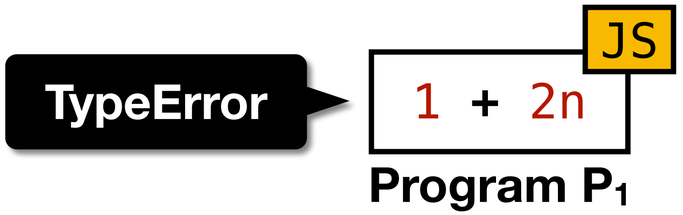
Motivating Example 1





Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1



feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

5. If **Type**(*lnum*) is different from **Type**(*rnum*),
   throw a **TypeError** exception.

…

1

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

PLRG

Motivating Example 1

Node Coverage

TR = Node

TypeError

JS

```
1 + 2n
```

Program P_1

feat

ADD

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

5. If **Type**(*lnum*) is different from **Type**(*rnum*),
   throw a **TypeError** exception.

…

1

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1

Node Coverage

TR = Node

TypeError

JS

```
1 + 2n
```

Program P_1

1

cover

feat

ADD

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

5. If **Type**(*lnum*) is different from **Type**(*rnum*),
   throw a **TypeError** exception.

…

1

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1

Node Coverage

TR = Node

TypeError JS `1 + 2n` — **Program P_1** — 1 — **cover**

TypeError JS `1 - 2n` — **Program P_2**

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

5. If **Type**(*lnum*) is different from **Type**(*rnum*),
   throw a **TypeError** exception. — 1

…

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1

Node Coverage

TR = Node

TypeError

JS

```
1 + 2n
```

1

Program P_1 cover

TypeError

JS

```
1 - 2n
```

Program P_2

feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

feat **SUB**

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

5. If **Type**(*lnum*) is different from **Type**(*rnum*),
   throw a **TypeError** exception.

…

1

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1

Node Coverage

TR = Node

TypeError `JS` `1 + 2n` **1**

Program P_1 **cover**

TypeError `JS` `1 - 2n` **1**

Program P_2 **cover**

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

`feat` **SUB**

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

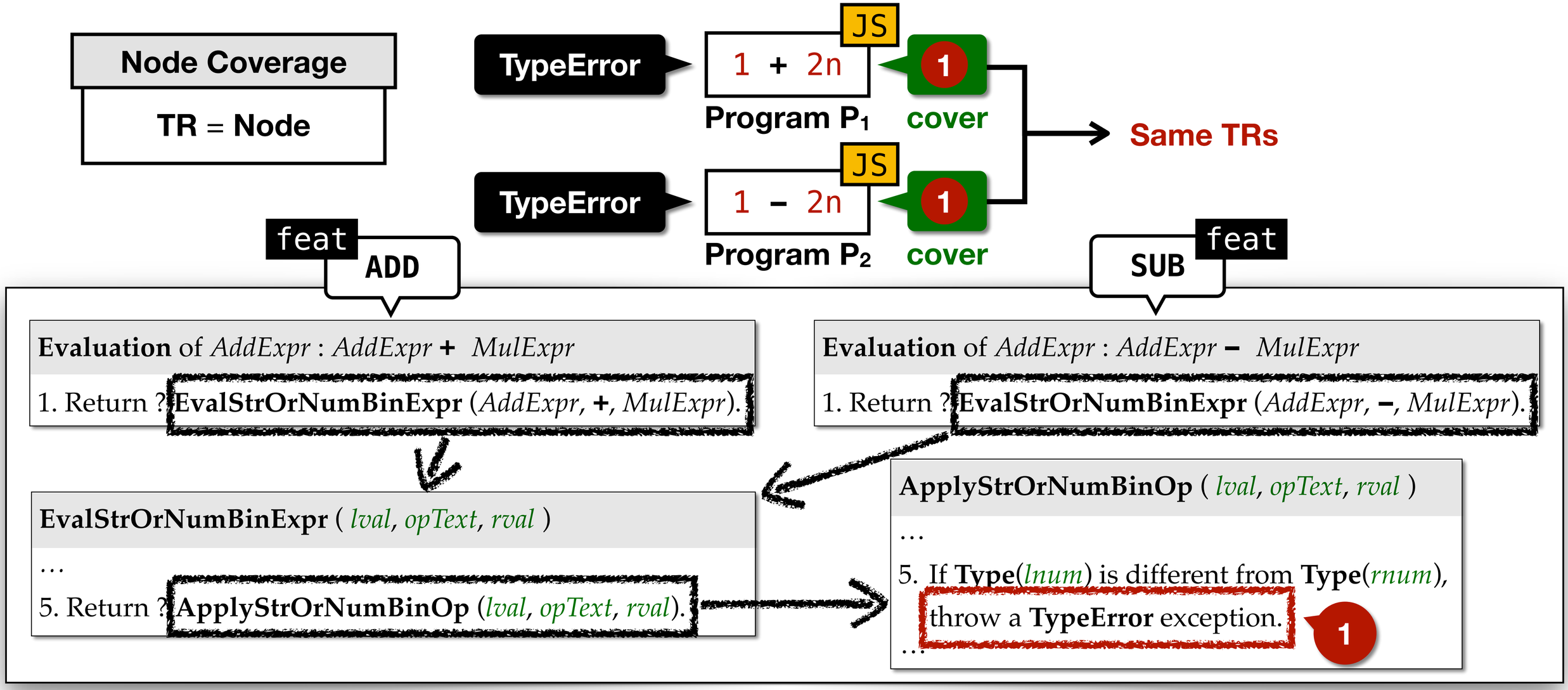
…

5. If **Type**(*lnum*) is different from **Type**(*rnum*),
   throw a **TypeError** exception. **1**

…

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1



Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Motivating Example 1

Node Coverage

TR = Node

TypeError — JS `1 + 2n` — **Program P_1** — 1 — cover

TypeError — JS `1 - 2n` — **Program P_2** — 1 — cover

→ **Same TRs** — **Cannot Distinguish P_1 and P_2**

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

…

5. Return ? **ApplyStrOrNumBinOp** (*lval*, *opText*, *rval*).

`feat` **SUB**

Evaluation of *AddExpr* : *AddExpr* **−** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **−**, *MulExpr*).

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

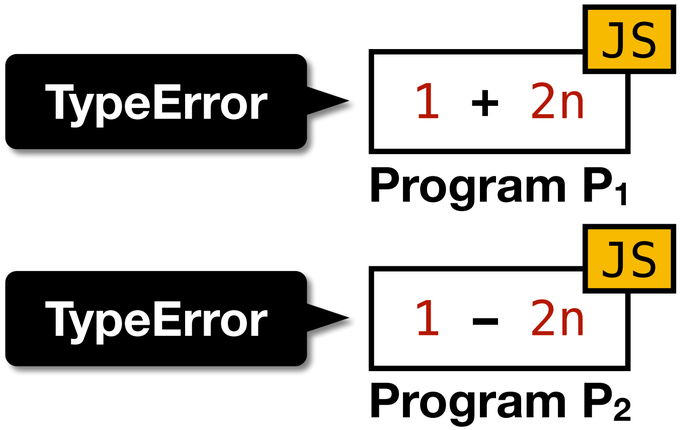
…

5. If **Type**(*lnum*) is different from **Type**(*rnum*), throw a **TypeError** exception. — 1

…

Abstract Algorithms in **ECMA-262** (ES13, 2022), **JavaScript Language Specification**

Feature-Sensitive (FS) Coverage

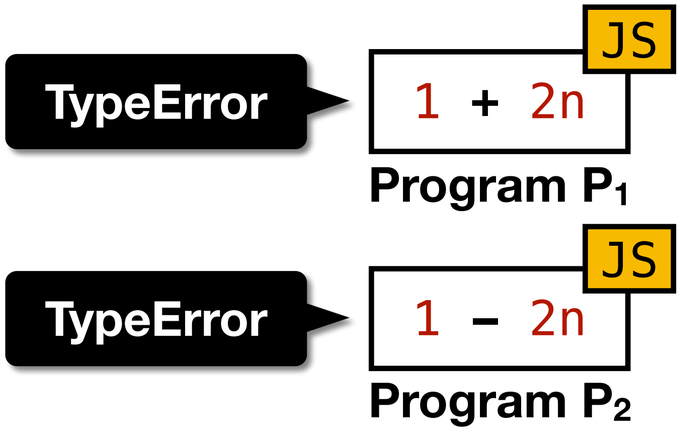


- Feature-Sensitive **(FS)** coverage criterion **divides**

  the given TRs with the **innermost enclosing** language **features**

FS Coverage
TR = (Feature, given **TR)**

Feature-Sensitive (FS) Coverage

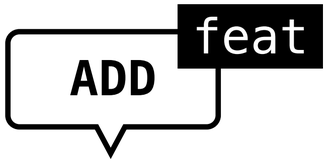


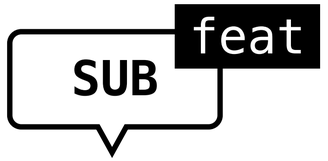
- Feature-Sensitive **(FS)** coverage criterion **divides**

  the given TRs with the **innermost enclosing** language **features**

FS Coverage

TR = (Feature, given **TR)**





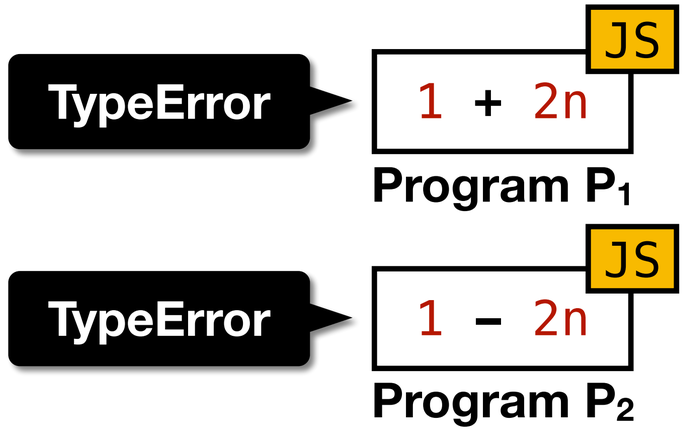
Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

Feature-Sensitive (FS) Coverage

FS Node Coverage

TR = (Feature, Node)

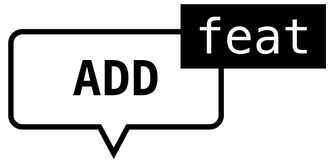


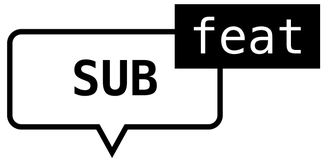
- **F**eature-**S**ensitive **(FS)** coverage criterion **divides**

  the given TRs with the **innermost enclosing** language **features**

FS Coverage

TR = (Feature, given **TR)**





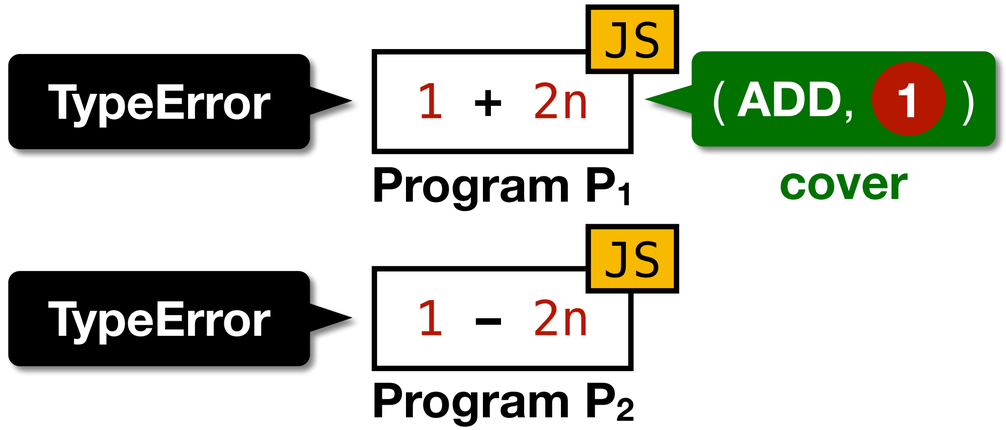
Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

Feature-Sensitive (FS) Coverage

FS Node Coverage

TR = (Feature, Node)

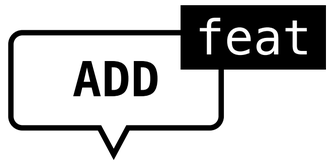


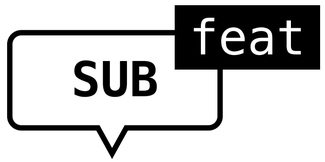
- **F**eature-**S**ensitive **(FS)** coverage criterion **divides**

  the given TRs with the **innermost enclosing** language **features**

FS Coverage

TR = (Feature, given **TR)**





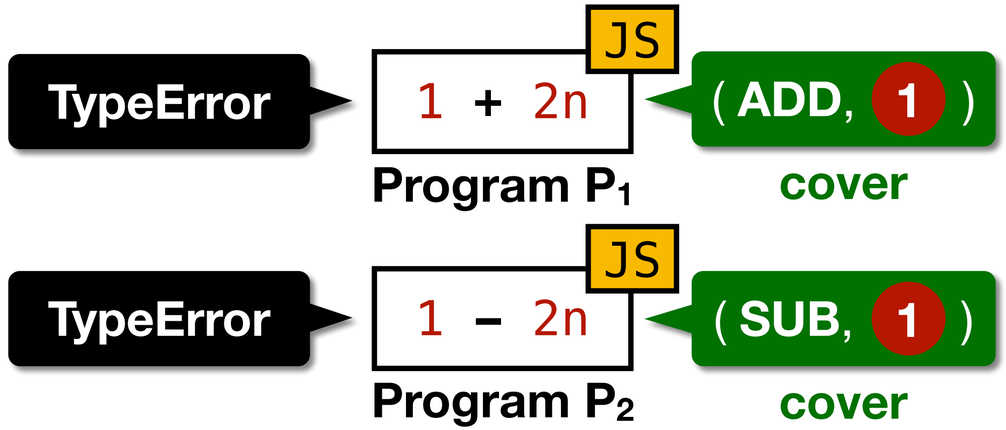
Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

Feature-Sensitive (FS) Coverage

FS Node Coverage

TR = (Feature, Node)

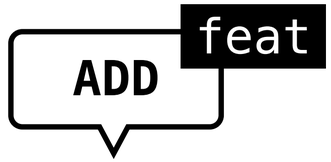


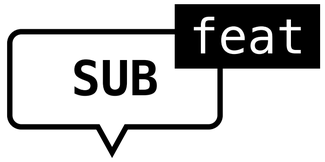
- **F**eature-**S**ensitive **(FS)** coverage criterion **divides**

  the given TRs with the **innermost enclosing** language **features**

FS Coverage

TR = (Feature, given **TR)**





Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

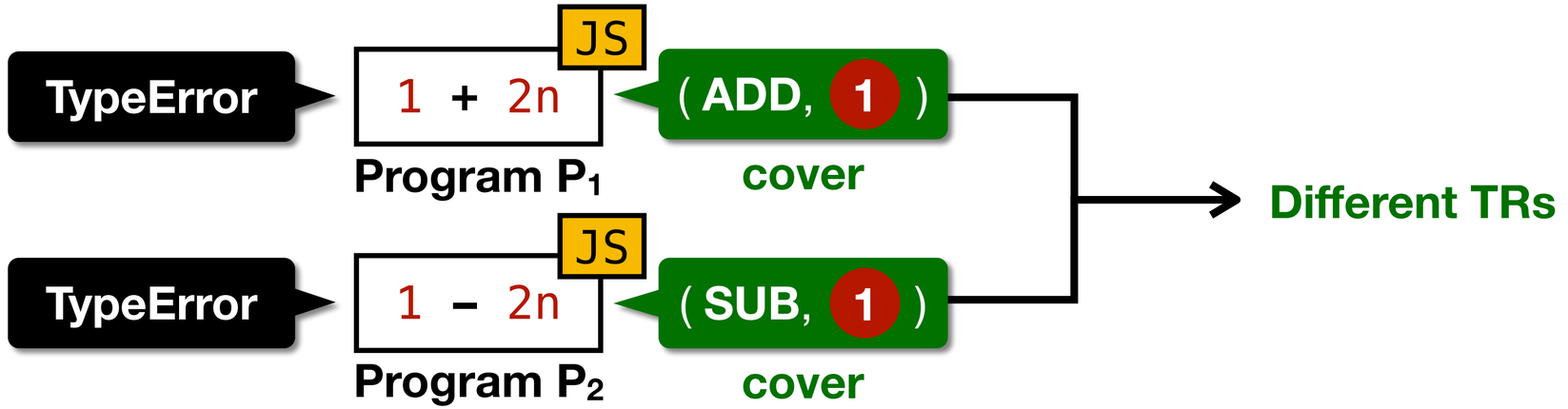
Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*

1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

Feature-Sensitive (FS) Coverage

FS Node Coverage

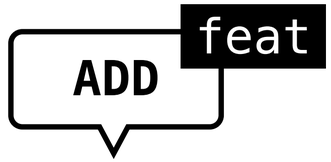
TR = (Feature, Node)

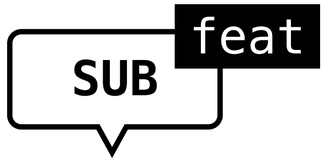


- **F**eature-**S**ensitive **(FS)** coverage criterion **divides** the given TRs with the **innermost enclosing** language **features**

FS Coverage

TR = (Feature, given **TR)**

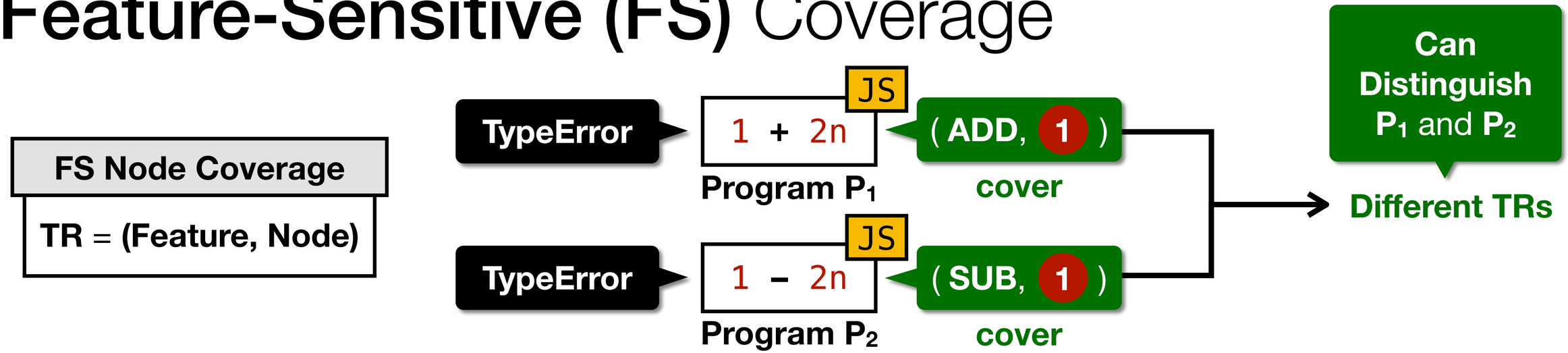




Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

Feature-Sensitive (FS) Coverage

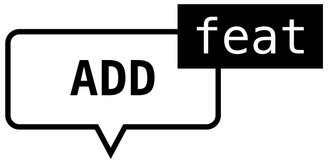


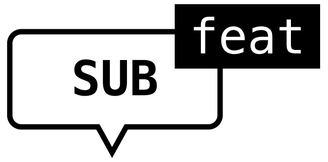
- Feature-Sensitive **(FS)** coverage criterion **divides**

  the given TRs with the **innermost enclosing** language **features**

FS Coverage

TR = (Feature, given **TR)**

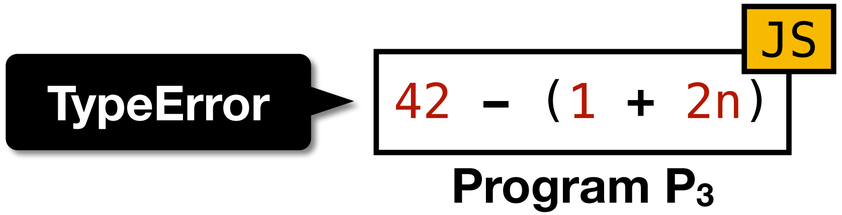




Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

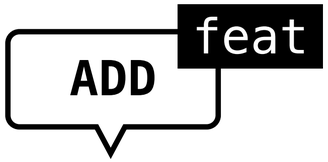
Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

k-Feature-Sensitive (k-FS) Coverage

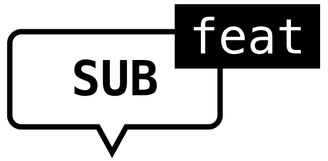


- k-**F**eature-**S**ensitive (**k-FS**) coverage criterion **divides** the given TRs with at most **k-innermost enclosing** language **features**

k-FS Coverage
TR = (Feature$^{\leq k}$, given **TR)**



Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).



Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

k-Feature-Sensitive (k-FS) Coverage

2-FS Node Coverage

TR = (Feature$^{\leq 2}$, Node)

TypeError

JS

```
42 - (1 + 2n)
```

Program P_3

- k-Feature-Sensitive (**k-FS**) coverage criterion **divides** the given TRs with at most **k-innermost enclosing** language **features**

k-FS Coverage

TR = (Feature$^{\leq k}$, given TR)

feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

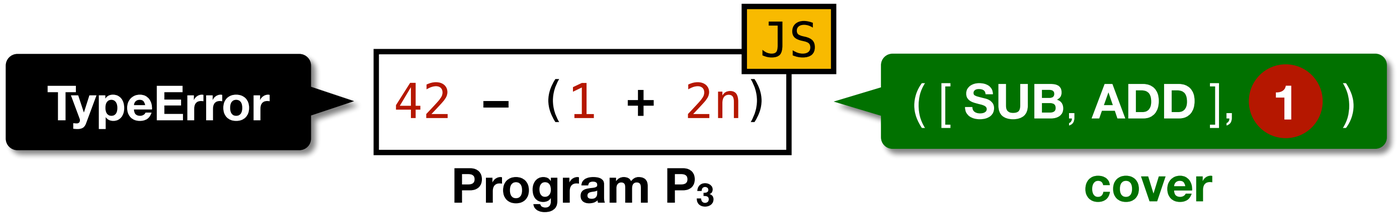
feat **SUB**

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

k-Feature-Sensitive (k-FS) Coverage

2-FS Node Coverage

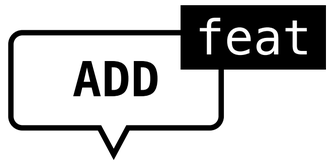
TR = (Feature$^{\leq 2}$, Node)

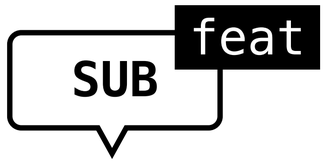


- k-**F**eature-**S**ensitive (**k-FS**) coverage criterion **divides** the given TRs with at most **k-innermost enclosing** language **features**

k-FS Coverage

TR = (Feature$^{\leq k}$, given **TR)**

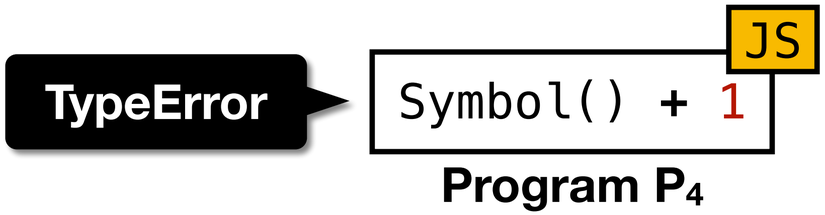




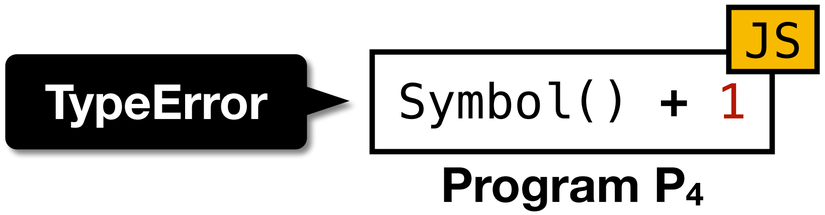
Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **+**, *MulExpr*).

Evaluation of *AddExpr* : *AddExpr* **–** *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, **–**, *MulExpr*).

Motivating Example 2



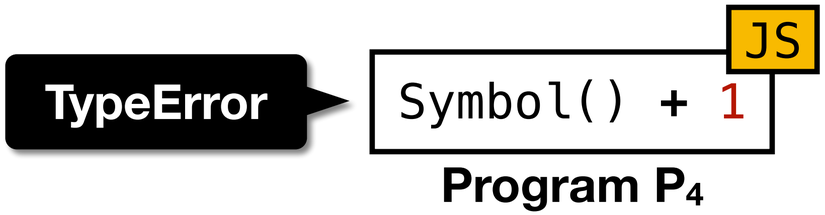
Motivating Example 2

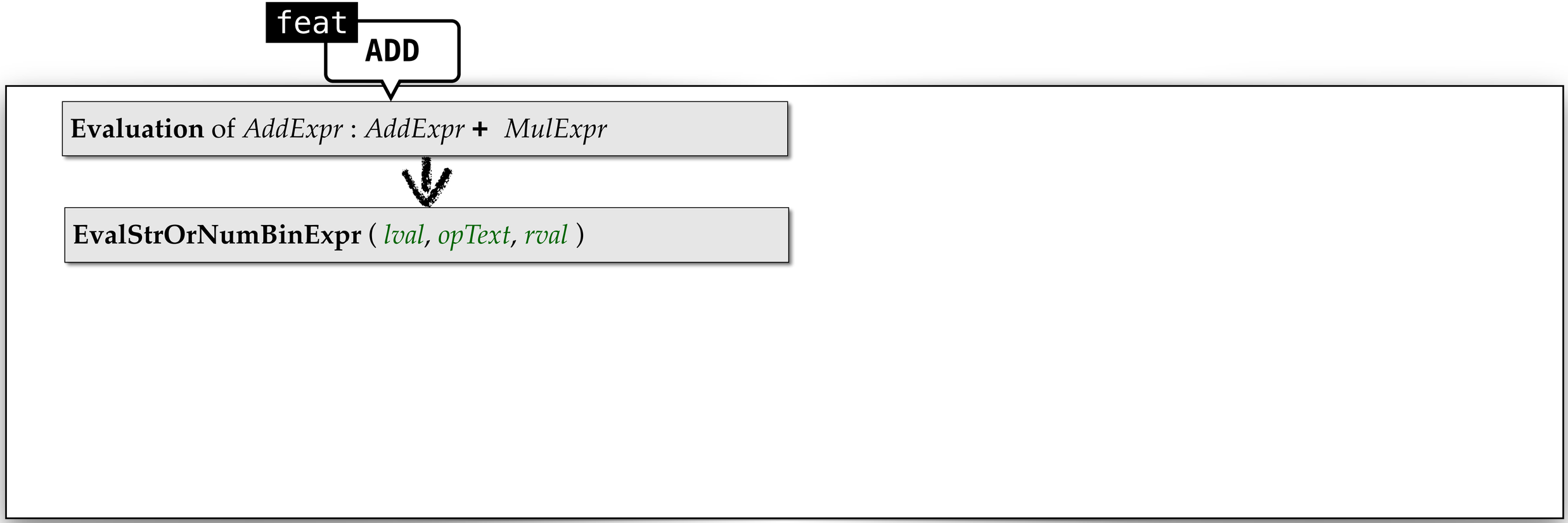


feat **ADD**

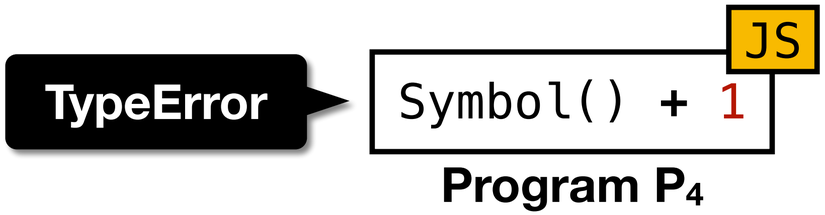
Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

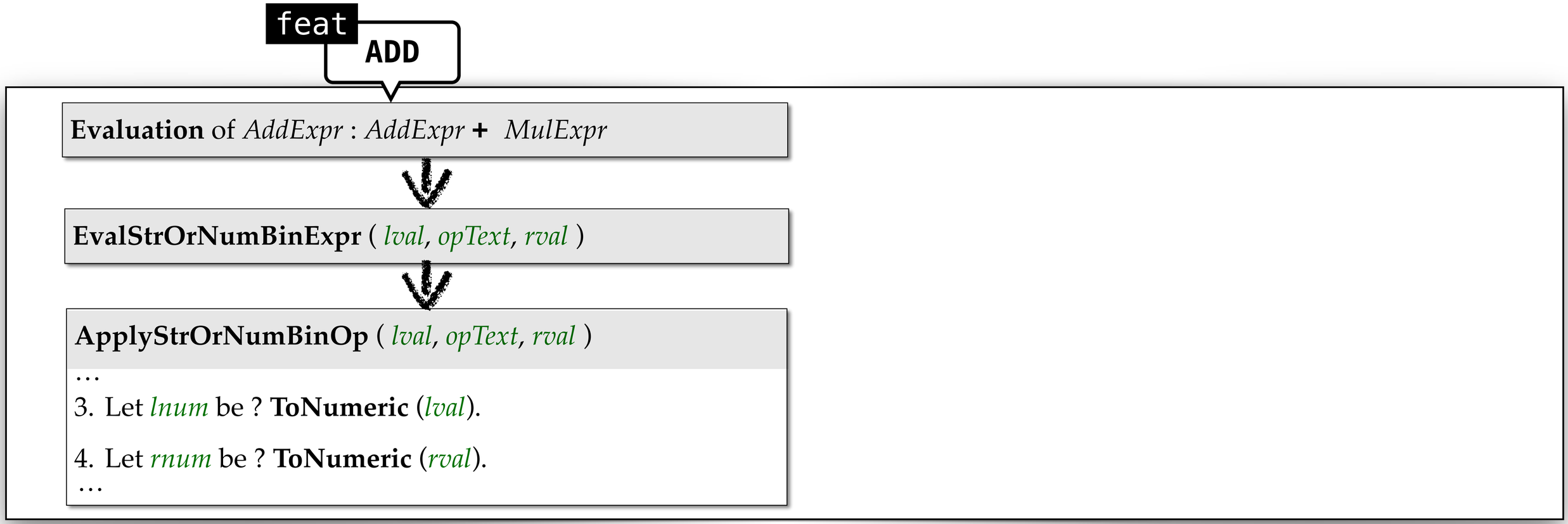
Motivating Example 2



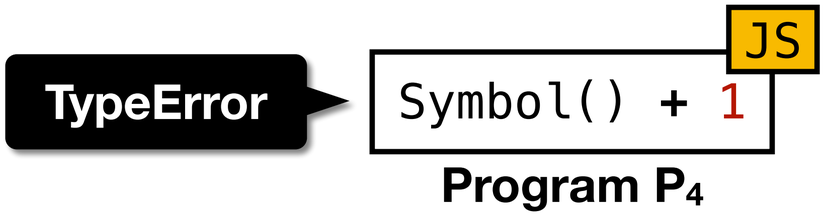


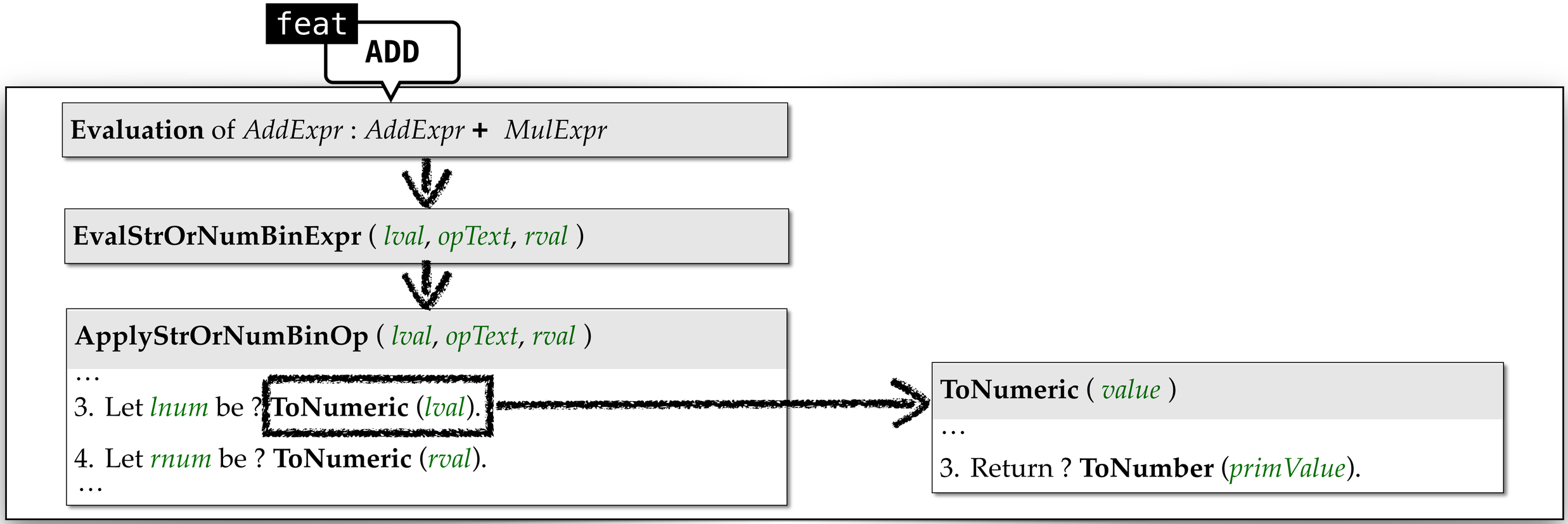
Motivating Example 2



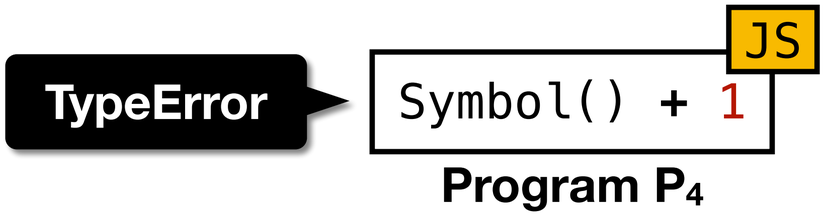


Motivating Example 2





Motivating Example 2



feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

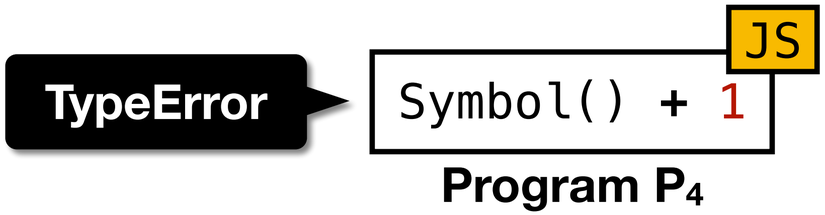
ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,
   throw a **TypeError** exception.

…

Motivating Example 2



feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,
   throw a **TypeError** exception.

…

2

Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError

JS

```
Symbol() + 1
```

Program P_4

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,
   throw a **TypeError** exception.

…

2

Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError

JS

```
Symbol() + 1
```

Program P_4

(ADD, 2)

cover

feat **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,

throw a **TypeError** exception.

…

2

Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError `Symbol() + 1` JS

Program P_4

(**ADD**, **2**) **cover**

TypeError `1 + Symbol()` JS

Program P_5

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,

throw a **TypeError** exception. **2**

…

Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError `Symbol() + 1` JS

Program P_4

(**ADD**, **2**) **cover**

TypeError `1 + Symbol()` JS

Program P_5

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,

throw a **TypeError** exception.

…

2

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError `Symbol() + 1` JS — **Program P_4** — **(ADD, 2)** **cover**

TypeError `1 + Symbol()` JS — **Program P_5** — **(ADD, 2)** **cover**

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

ToNumber (*argument*)

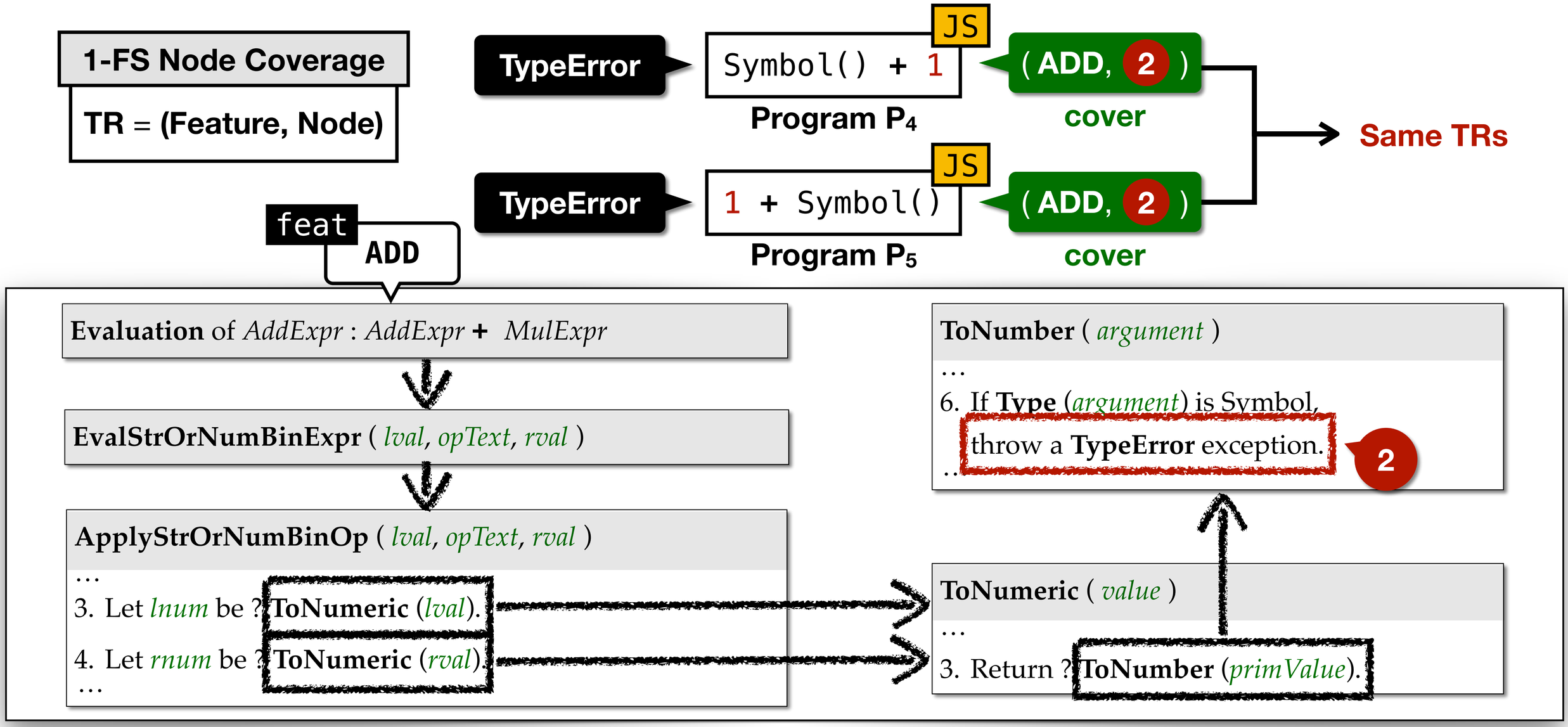
…

6. If **Type** (*argument*) is Symbol,

throw a **TypeError** exception. 2

…

Motivating Example 2



Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError `Symbol() + 1` (JS) — **Program P_4** — **(ADD, 2)** **cover**

TypeError `1 + Symbol()` (JS) — **Program P_5** — **(ADD, 2)** **cover**

Cannot Distinguish P_4 and P_5

Same TRs

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*).

4. Let *rnum* be ? **ToNumeric** (*rval*).

…

ToNumeric (*value*)

…

3. Return ? **ToNumber** (*primValue*).

ToNumber (*argument*)

…

6. If **Type** (*argument*) is Symbol,

throw a **TypeError** exception. **2**

…

Motivating Example 2

1-FS Node Coverage

TR = (Feature, Node)

TypeError `Symbol() + 1` JS — **Program P_4** — **(ADD, 2)** **cover**

TypeError `1 + Symbol()` JS — **Program P_5** — **(ADD, 2)** **cover**

Cannot Distinguish P_4 and P_5

Same TRs

`feat` **ADD**

Evaluation of *AddExpr* : *AddExpr* **+** *MulExpr*

3 call

EvalStrOrNumBinExpr (*lval*, *opText*, *rval*)

4 call

ApplyStrOrNumBinOp (*lval*, *opText*, *rval*)

…

3. Let *lnum* be ? **ToNumeric** (*lval*). — 5 call

4. Let *rnum* be ? **ToNumeric** (*rval*). — 6 call

…

ToNumeric (*value*) — 7 call

…

3. Return ? **ToNumber** (*primValue*).

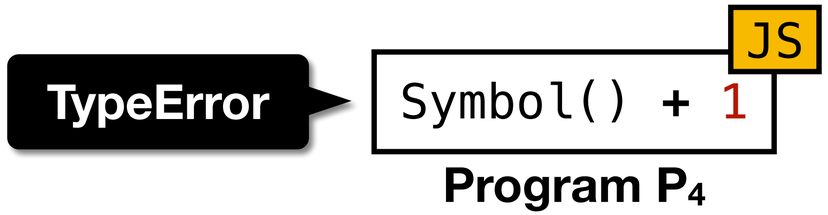
ToNumber (*argument*)

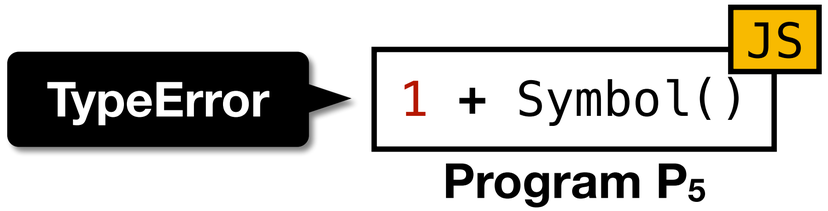
…

6. If **Type** (*argument*) is Symbol, throw a **TypeError** exception. — 2

…

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



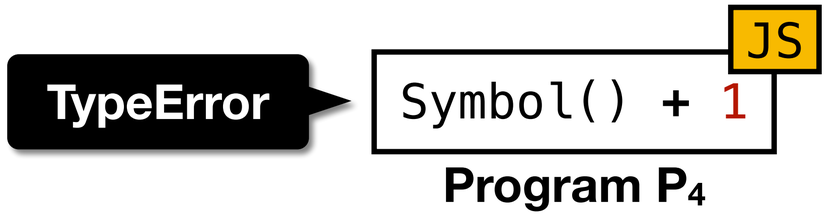


- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

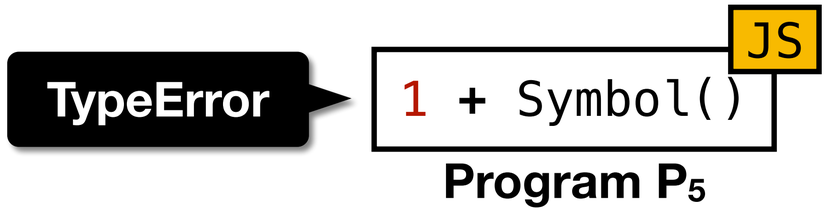
TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



1-FCPS Node Coverage

TR = (Feature, Call-Path, Node)

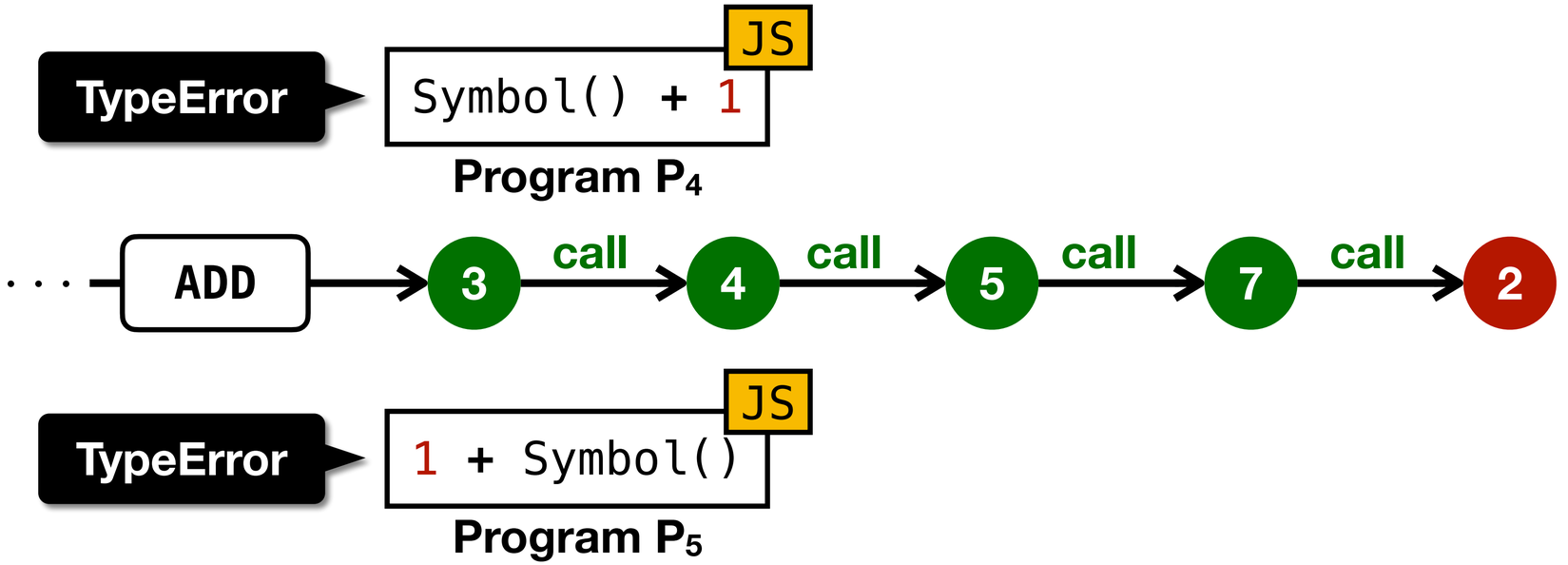


- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



1-FCPS Node Coverage

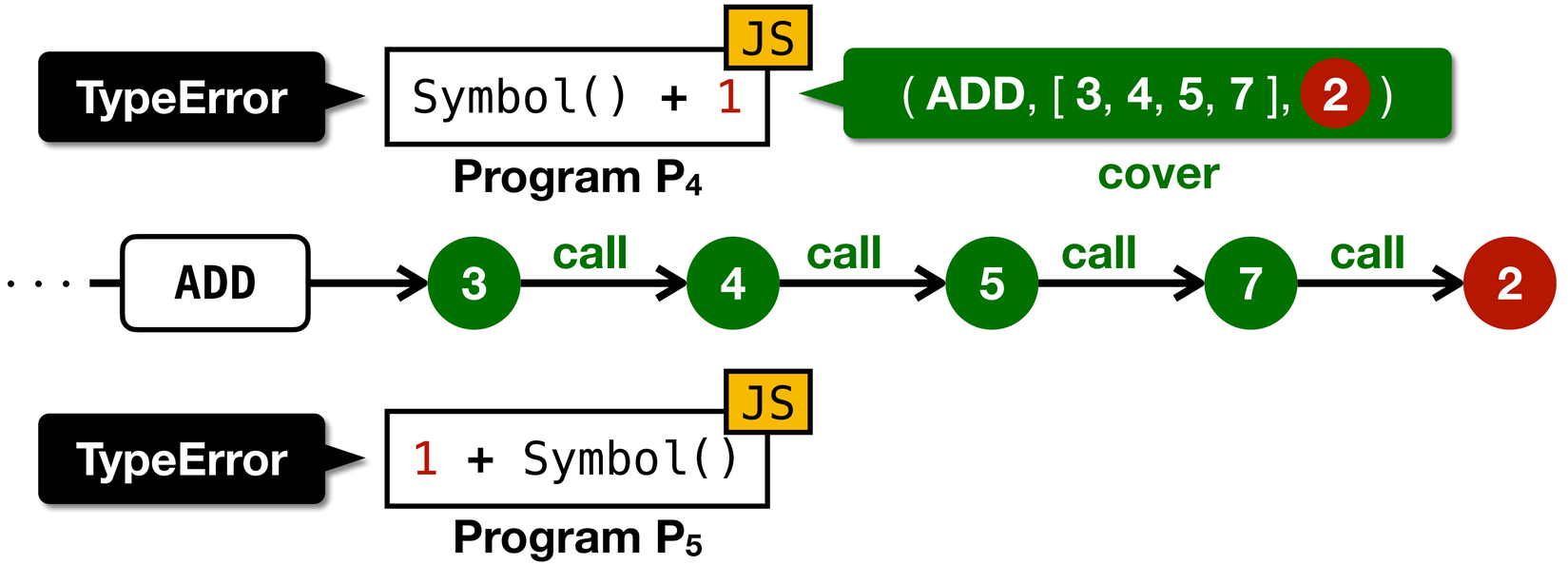
TR = (Feature, Call-Path, Node)

- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



1-FCPS Node Coverage

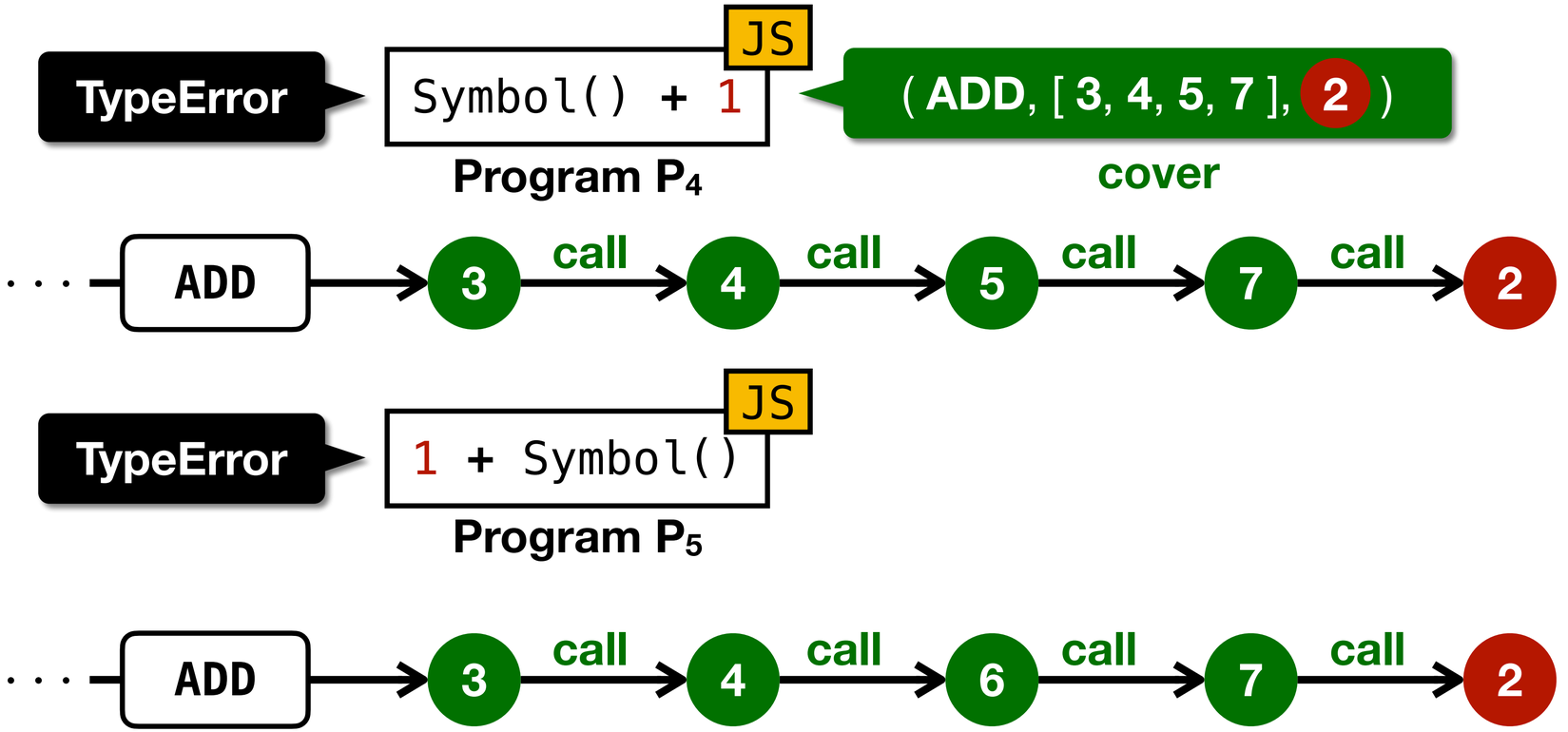
TR = (Feature, Call-Path, Node)

- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



1-FCPS Node Coverage

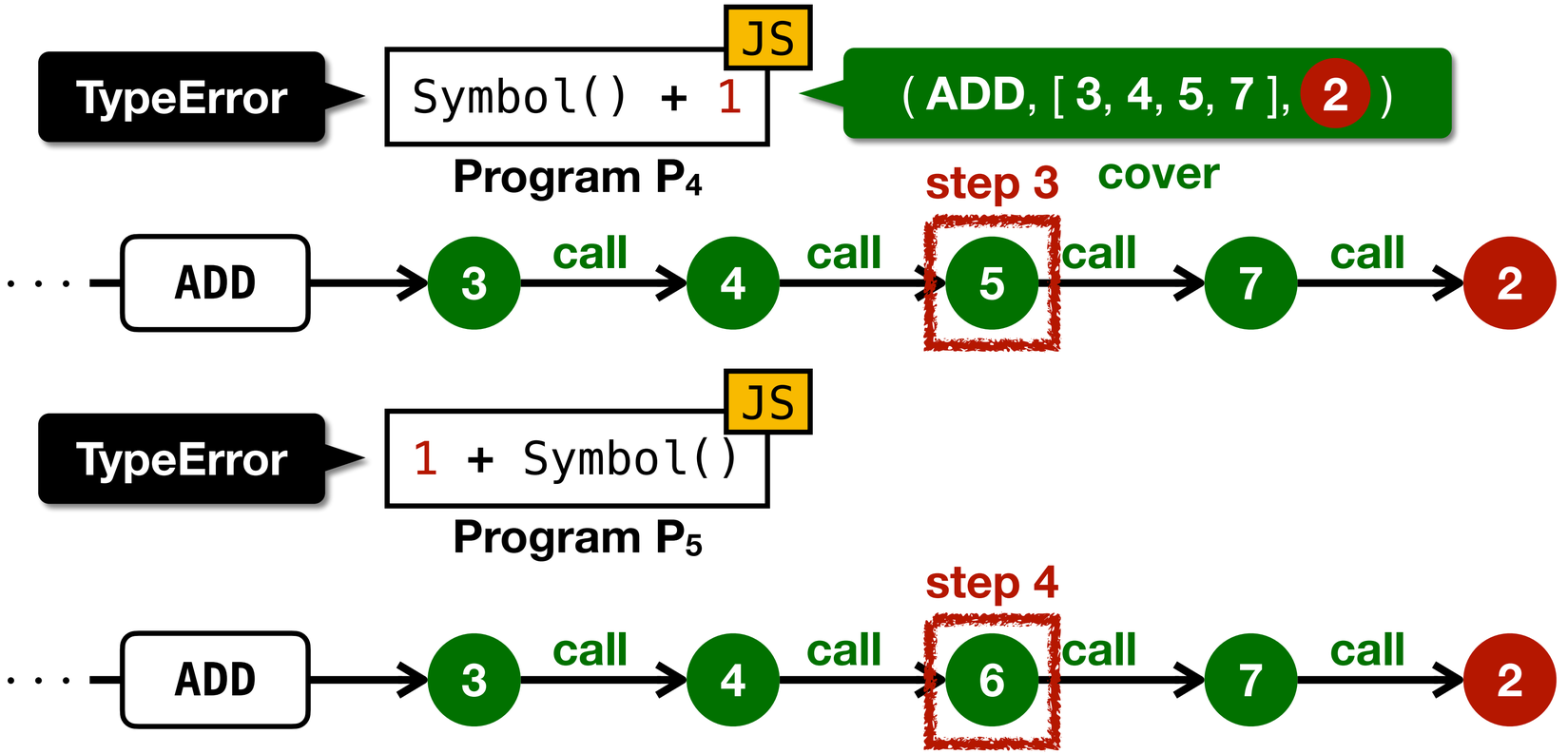
TR = (Feature, Call-Path, Node)

- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



1-FCPS Node Coverage

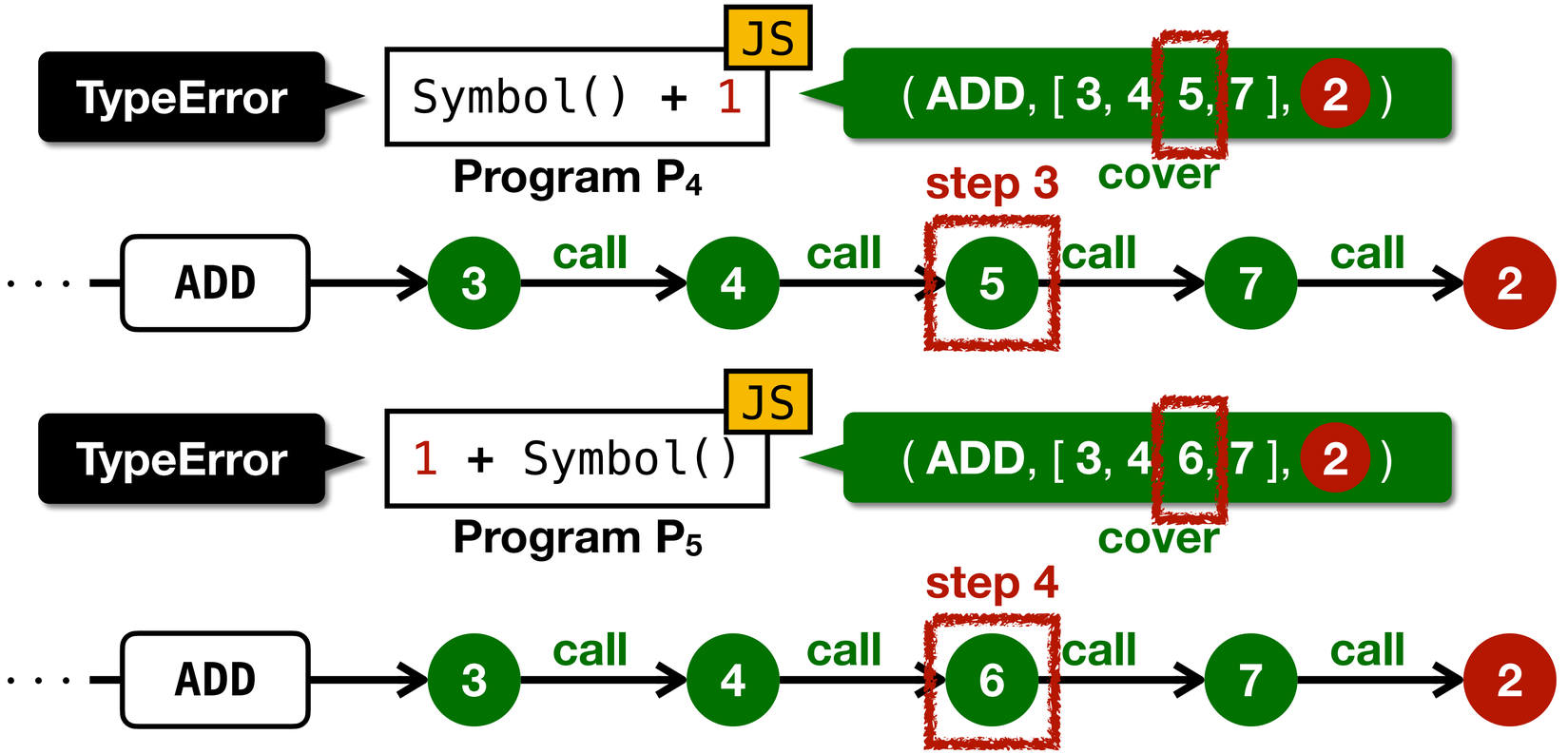
TR = (Feature, Call-Path, Node)

- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



1-FCPS Node Coverage

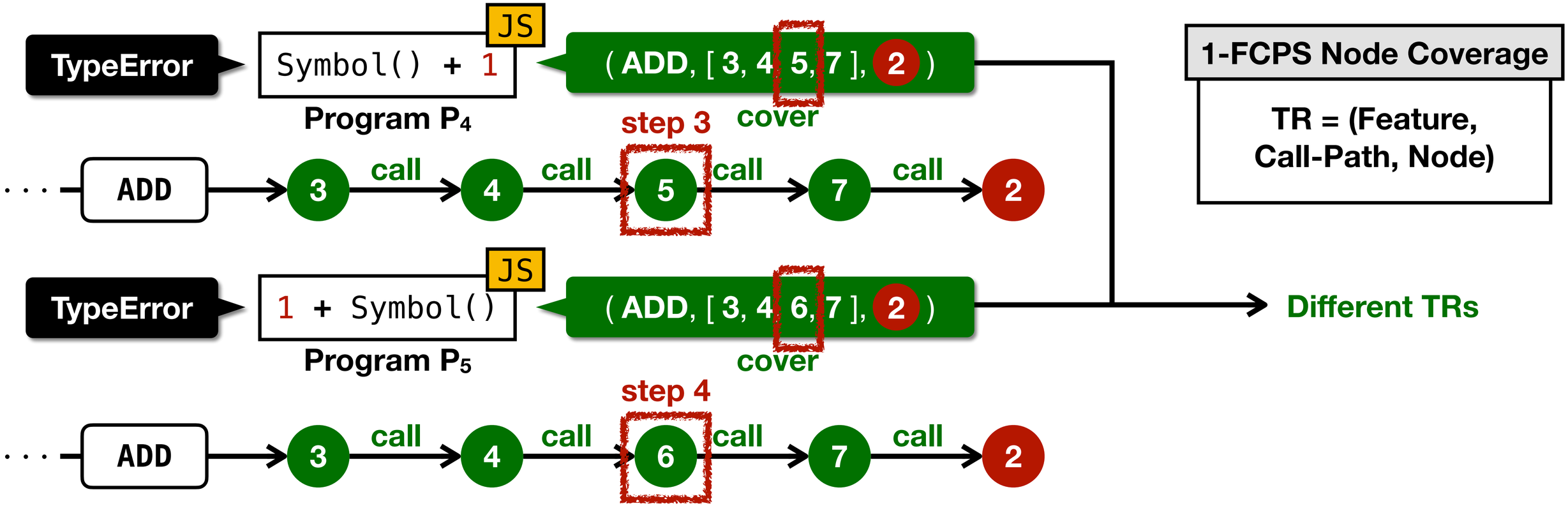
TR = (Feature, Call-Path, Node)

- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage

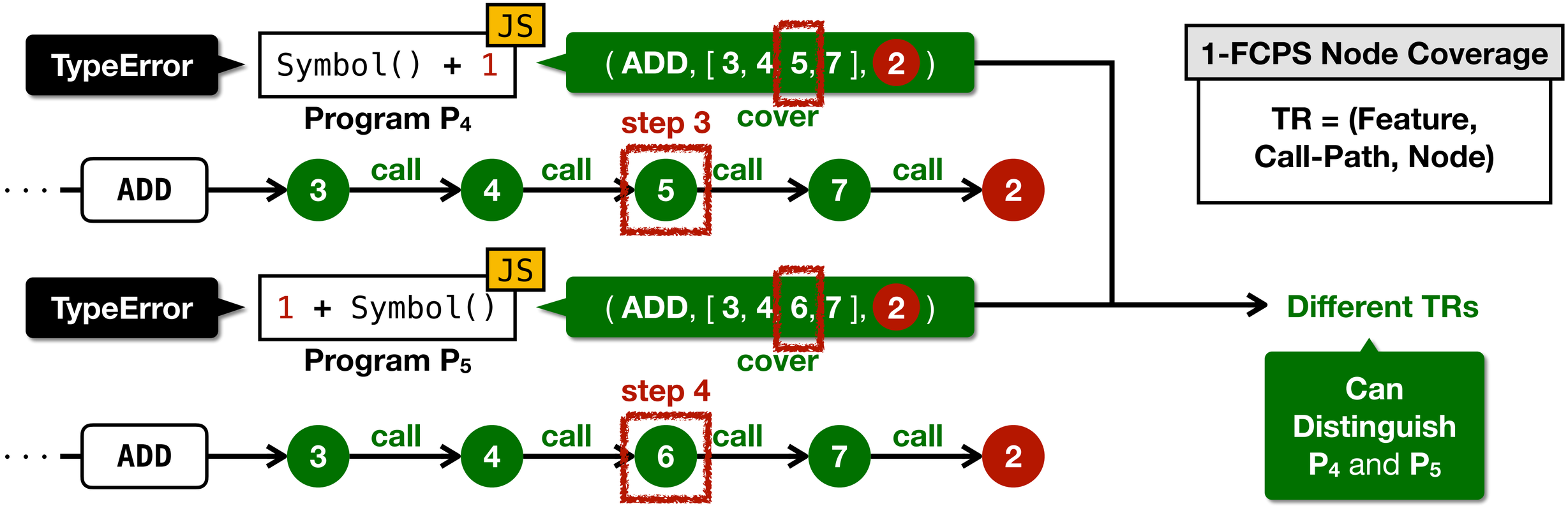


- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage



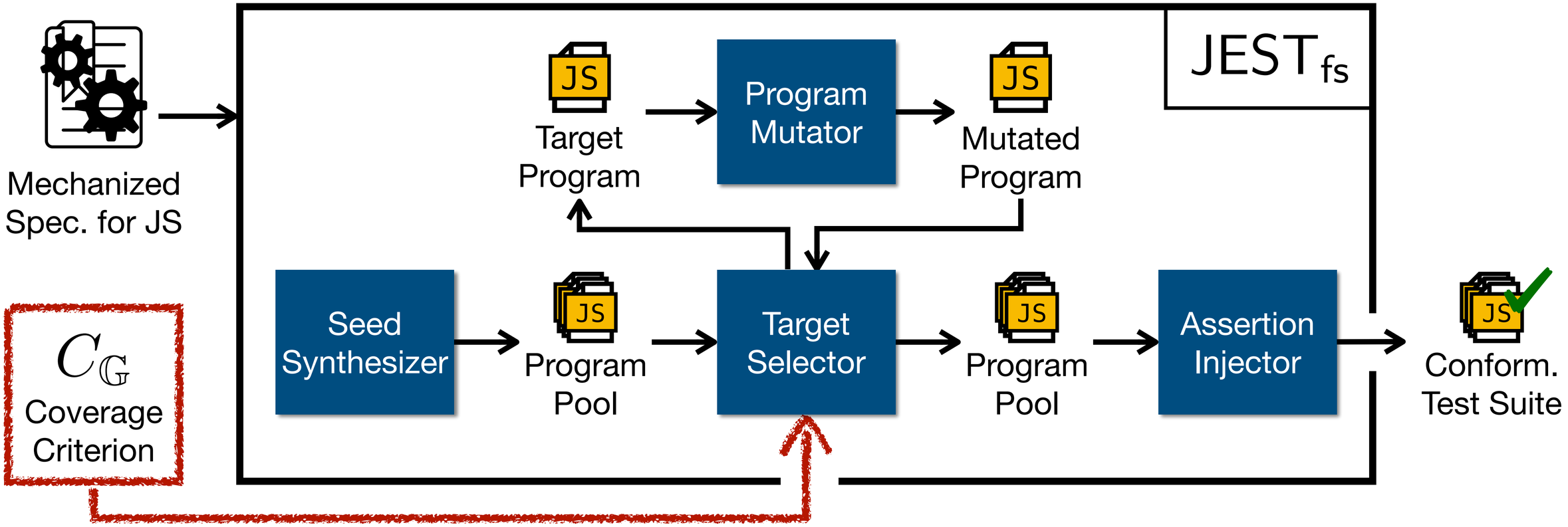
- k-**F**eature-**C**all-**P**ath-**S**ensitive (k-**FCPS**) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

k-FCPS Coverage

TR = (Feature$^{\leq k}$, Call-Path, given **TR)**

Implementation — JEST$_{fs}$

- **JEST** [1] is a **JavaScript conformance test generator** using **Coverage-Guided Fuzzing**
- We implemented **JEST$_{fs}$** as an extension of **JEST** with k**-FS** and k**-FCPS** coverage criteria



[1] Park et al., "JEST: N+1-version Differential Testing of Both JavaScript Engines and Specification", ICSE 2021

Evaluation

5 different k-FS and k-FCPS coverage criteria

- **JEST$_{fs}$** in 50 hours with **0-FS** / **1-FS** / **2-FS** / **1-FCPS** / **2-FCPS**
- **JavaScript Specification** — ECMA-262 for **ES13 (2022)**
- **JavaScript Implementations** — **4 Engines** and **4 Transpilers**

Kind	Name	Version	Release
Engine	**V8**	v10.8.121	2022.10.06
	JSC	v615.1.10	2022.10.26
	GraalJS	v22.2.0	2022.07.26
	SpiderMonkey	v107.0b4	2022.10.24
Transpiler	**Babel**	v7.19.1	2022.09.15
	SWC	v1.3.10	2022.10.21
	Terser	v5.15.1	2022.10.05
	Obfuscator	v4.0.0	2022.02.15

RQ1) Conformance Bug Detection

Kind	Name	Version	Release	# Detected Unique Bugs		
				# New	# Confirmed	# Reported
Engine	V8	v10.8.121	2022.10.06	0	0	4
	JSC	v615.1.10	2022.10.26	15	15	24
	GraalJS	v22.2.0	2022.07.26	9	9	10
	SpiderMonkey	v107.0b4	2022.10.24	1	3	4
	Total			**25**	**27**	**42**
Transpiler	Babel	v7.19.1	2022.09.15	30	30	35
	SWC	v1.3.10	2022.10.21	27	27	41
	Terser	v5.15.1	2022.10.05	1	1	18
	Obfuscator	v4.0.0	2022.02.15	0	0	7
	Total			**58**	**58**	**101**
Total				**83**	**85**	**143**

RQ1) Conformance Bug Detection

Kind	Name	Version	Release	# Detected Unique Bugs		
				# New	# Confirmed	# Reported
Engine	V8	v10.8.121	2022.10.06	0	0	4
	JSC	v615.1.10	2022.10.26	15	15	24
	GraalJS	v22.2.0	2022.07.26	9	9	10
	SpiderMonkey	v107.0b4	2022.10.24	1	3	4
	Total			**25**	**27**	**42**
Transpiler	Babel	v7.19.1	2022.09.15	30	30	35
	SWC	v1.3.10	2022.10.21	27	27	41
	Terser	v5.15.1	2022.10.05	1	1	18
	Obfuscator	v4.0.0	2022.02.15	0	0	7
	Total			**58**	**58**	**101**
Total				**83**	**85**	**143**

RQ1) Conformance Bug Detection

Kind	Name	Version	Release	# Detected Unique Bugs		
				# New	# Confirmed	# Reported
Engine	V8	v10.8.121	2022.10.06	0	0	4
	JSC	v615.1.10	2022.10.26	15	15	24
	GraalJS	v22.2.0	2022.07.26	9	9	10
	SpiderMonkey	v107.0b4	2022.10.24	1	3	4
	Total			**25**	**27**	**42**
Transpiler	Babel	v7.19.1	2022.09.15	30	30	35
	SWC	v1.3.10	2022.10.21	27	27	41
	Terser	v5.15.1	2022.10.05	1	1	18
	Obfuscator	v4.0.0	2022.02.15	0	0	7
	Total			**58**	**58**	**101**
Total				**83**	**85**	**143**

RQ1) Conformance Bug Detection

Kind	Name	Version	Release	# Detected Unique Bugs		
				# New	# Confirmed	# Reported
Engine	V8	v10.8.121	2022.10.06	0	0	4
	JSC	v615.1.10	2022.10.26	15	15	24
	GraalJS	v22.2.0	2022.07.26	9	9	10
	SpiderMonkey	v107.0b4	2022.10.24	1	3	4
	Total			**25**	**27**	**42**
Transpiler	Babel	v7.19.1	2022.09.15	30	30	35
	SWC	v1.3.10	2022.10.21	27	27	41
	Terser	v5.15.1	2022.10.05	1	1	18
	Obfuscator	v4.0.0	2022.02.15	0	0	7
	Total			**58**	**58**	**101**
Total				**83**	**85**	**143**

RQ2) Effectiveness of k-FS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

RQ2) Effectiveness of k-FS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

RQ2) Effectiveness of k-FS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+28

Spec.

Expected

Terminated

```
for (let {} = 0; 0; ) ;
```

Wrong Result

Crash

Babel

Synthesized with **1-FS** but not with **0-FS**

RQ2) Effectiveness of k-FS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+28

+19

Spec.

Expected

Terminated

```
for (let {} = 0; 0; ) ;
```

Wrong Result

Crash

Babel

Synthesized with **1-FS** but not with **0-FS**

RQ2) Effectiveness of k-FS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+28

+19

Spec.

Expected

Terminated

```
for (let {} = 0; 0; ) ;
```

Wrong Result

Crash

Babel

Synthesized with **1-FS** but not with **0-FS**

Spec.

Expected

"f"

```
class C { async ["f"](){} }
C.prototype.f.name
```

Wrong Result

"async"

JSC

Synthesized with **2-FS** but not with **1-FS**

RQ3) Effectiveness of k-FCPS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

RQ3) Effectiveness of k-FCPS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+4

RQ3) Effectiveness of k-FCPS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+4

+9

RQ3) Effectiveness of k-FCPS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+4

+9

Spec.

Expected

RangeError

```
String.prototype
      .normalize
      .call(0, "");
```

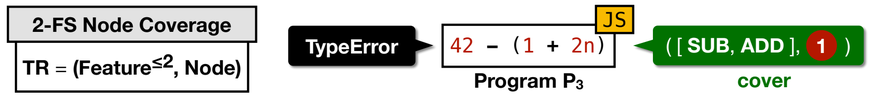
Wrong Result

Terminated

GraalJS

Synthesized with **1-FCPS** or **2-FCPS** but not with **1-FS** or **2-FS**

k-Feature-Sensitive (k-FS) Coverage



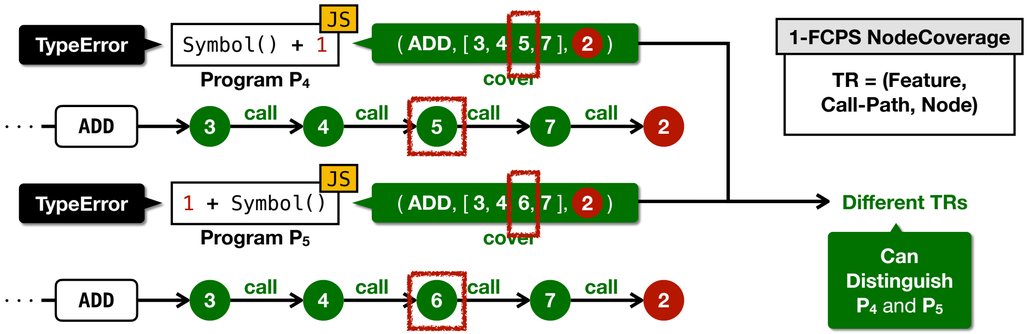
- k-Feature-Sensitive (k-FS) coverage criterion **divides** the given TRs with the at most k-**innermost enclosing** language **features**

FS Coverage
TR = (Feature≤k, given TR)

Evaluation of *AddExpr* : *AddExpr* + *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, +, *MulExpr*).

Evaluation of *AddExpr* : *AddExpr* − *MulExpr*
1. Return ? **EvalStrOrNumBinExpr** (*AddExpr*, −, *MulExpr*).

k-Feature-Call-Path-Sensitive (k-FCPS) Coverage

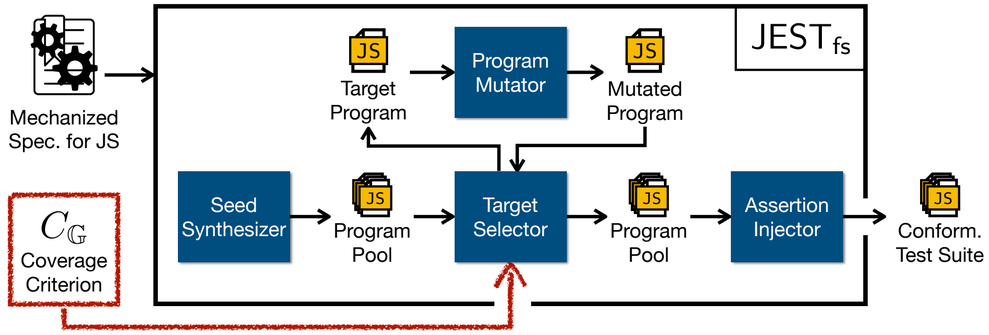


- k-Feature-Call-Path-Sensitive (k-FCPS) coverage criterion divides the k-FS TRs with the **call-paths from** the innermost enclosing language feature

FS Coverage
TR = (Feature≤k, Call-Path, given TR)

Implementation — JEST$_{fs}$

- **JEST** [1] is a **JavaScript conformance test generator** using **Coverage-Guided Fuzzing**
- We implemented **JEST$_{fs}$** as an extension of **JEST** with k-**FS** and k-**FCPS** coverage criteria



[1] Park et al., "JEST: N+1-version Differential Testing of Both JavaScript Engines and Specification", ICSE 2021

RQ2) Effectiveness of k-FS Coverage Criteria

Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

+28
+19

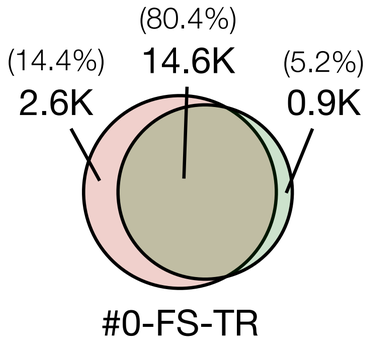
RQ3) Effectiveness of k-FCPS Coverage Criteria

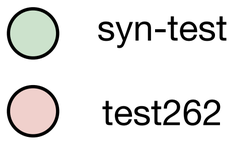
Coverage Criteria $C_{\mathbb{G}}$	# Covered k-F(CP)S-TR (k)			# Syn. Test	# Bug
	# Node	# Branch	# Total		
0-FS node-or-branch (0-fs)	10.0	5.6	15.6	2,111	55
1-FS node-or-branch (1-fs)	79.3	45.7	125.0	6,766	83
1-FCPS node-or-branch (1-fcps)	179.7	97.6	277.3	9,092	87
2-FS node-or-branch (2-fs)	1,199.8	696.3	1,896.1	97,423	102
2-FCPS node-or-branch (2-fcps)	2,323.1	1,297.6	3,620.7	122,589	111

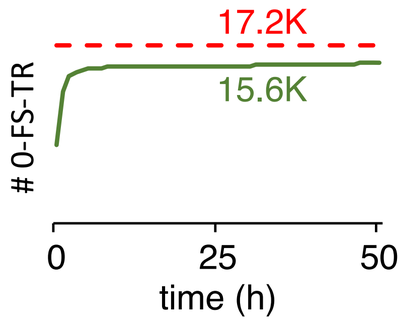
+4
+9

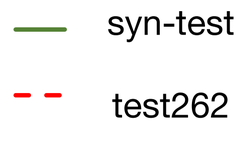
Backup Slides

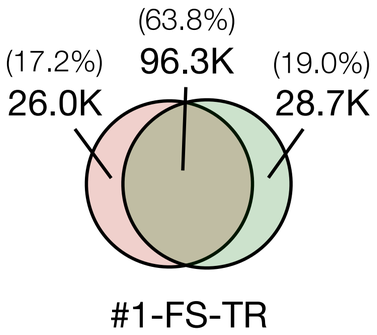
RQ4) Comparison with Test262

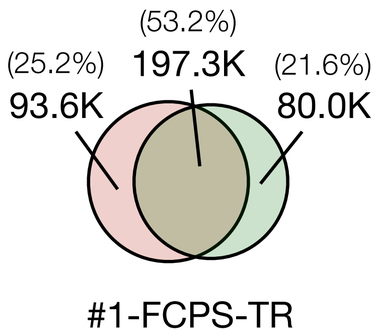


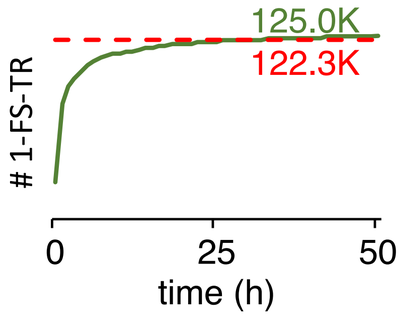


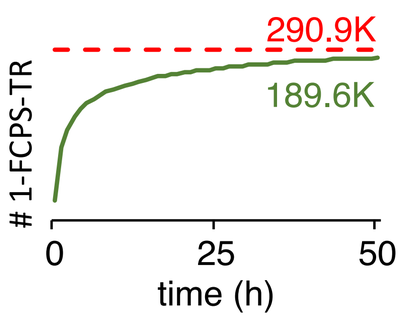


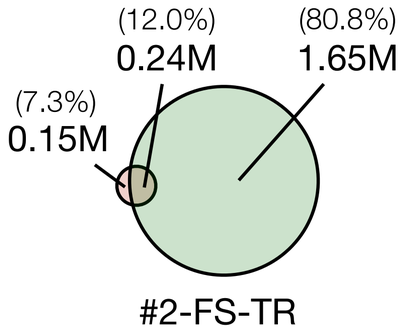


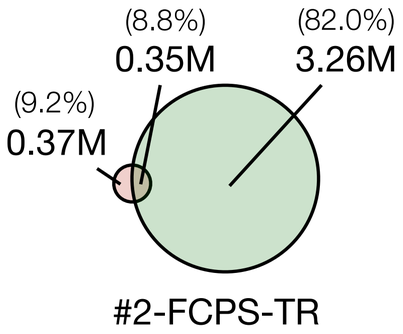


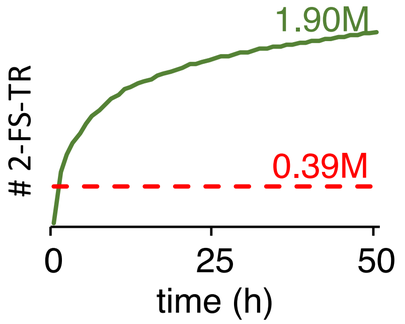


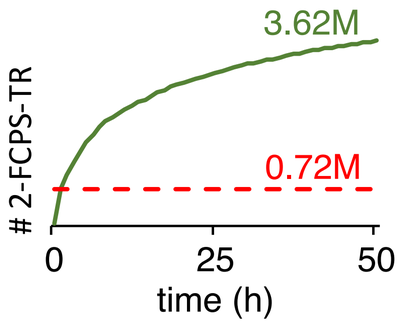












PLRG