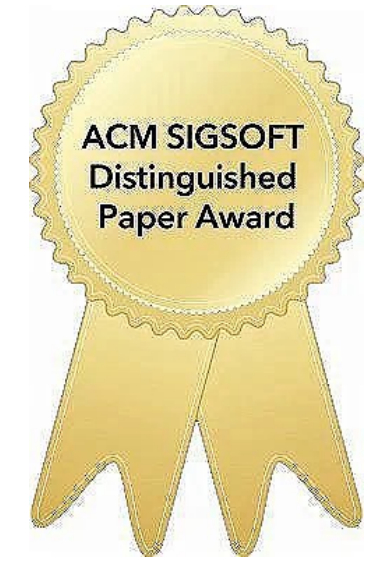




MiniPoly: Automatic Extraction of Efficient JavaScript Polyfill from Language Specification

Jungwoong Kim*, Youngmin Cho*, and Jihyeok Park

2026.10.13



KOREA
UNIVERSITY



(* equally contributed)

Built-in APIs in JavaScript

```
const a = new Set([1, 2, 3]);  
const b = new Set([1, 2]);  
a.isSupersetOf(b); // true
```

Built-in APIs in JavaScript

```
const a = new Set([1, 2, 3]);  
const b = new Set([1, 2]);  
a.isSupersetOf(b); // true
```



≥ 122



Built-in APIs in JavaScript

ES2025

```
const a = new Set([1, 2, 3]);  
const b = new Set([1, 2]);  
a.isSupersetOf(b); // true
```



≥ 122



Built-in APIs in JavaScript

ES2025

```
const a = new Set([1, 2, 3]);  
const b = new Set([1, 2]);  
a.isSupersetOf(b); // true
```



≤ 126



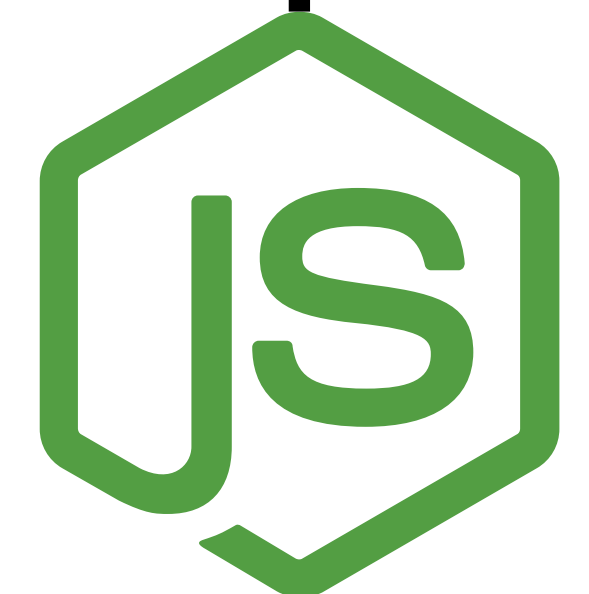
≤ 2025.09



≥ 122



≤ 16.7



≤ 21



Built-in APIs in JavaScript

ES2025

```
const a = new Set([1, 2, 3]);  
const b = new Set([1, 2]);  
a.isSupersetOf(b); // true
```

Native Code
(e.g., C++, Java)

```
Welcome to Node.js v24.14.1.  
Type ".help" for more information.  
> Set.prototype.isSupersetOf.toString()  
function isSupersetOf() { [native code] }  
>
```



≤ 126



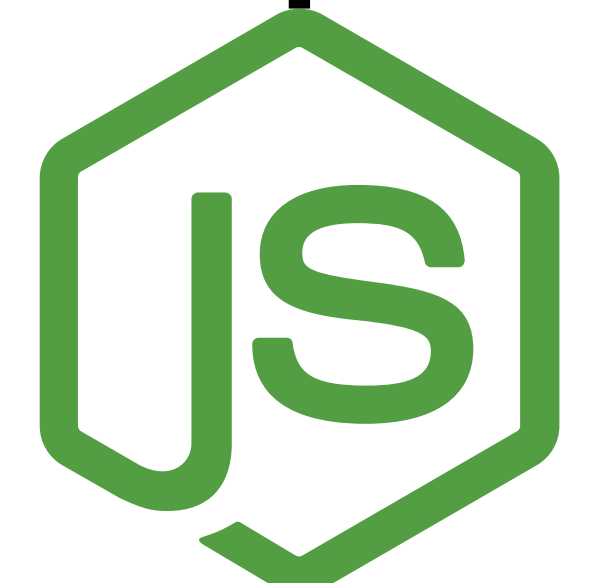
≤ 2025.09



≥ 122



≤ 16.7



≤ 21



Polyfill

- **Polyfill** is a **JavaScript** implementation of built-in APIs
- **core-js** is used by **41% of websites** and has **$\geq 46\text{M}$ downloads per week**

```
module.exports = function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator, function (e) {  
    if (!has(0, e)) {  
      return iteratorClose(iterator, 'normal', false);  
    }  
  }) !== false;  
};
```

core-js

Polyfill

```
import 'core-js/actual/set/is-superset-of';  
const a = new Set([1, 2, 3]);  
const b = new Set([1, 2]);  
a.isSupersetOf(b); // true
```

```
module.exports = function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator, function (e) {  
    if (!has(0, e)) {  
      return iteratorClose(iterator, 'normal', false);  
    }  
  }) !== false;  
};
```

core-js



≤ 126



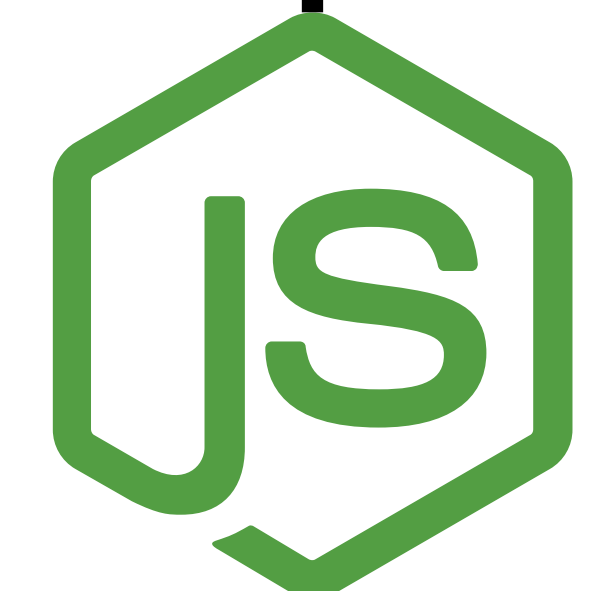
≤ 2025.09



≥ 122



≤ 16.7



≤ 21



ECMA-262 - JavaScript Language Specification

tc39/**ecma262**

Status, process, and documents for ECMA-262



ECMA-262 - JavaScript Language Specification

tc39/ecma262

Status, process, and documents for ECMA-262



Syntax

```
AdditiveExpression[Yield, Await] :  
  MultiplicativeExpression[?Yield, ?Await]  
  AdditiveExpression[?Yield, ?Await] + MultiplicativeExpression[?Yield, ?Await]  
  AdditiveExpression[?Yield, ?Await] - MultiplicativeExpression[?Yield, ?Await]
```

ECMA-262 - JavaScript Language Specification

tc39/ecma262

Status, process, and documents for ECMA-262



Syntax

*AdditiveExpression*_[Yield, Await] :

*MultiplicativeExpression*_[?Yield, ?Await]

*AdditiveExpression*_[?Yield, ?Await] + *MultiplicativeExpression*_[?Yield, ?Await]

*AdditiveExpression*_[?Yield, ?Await] - *MultiplicativeExpression*_[?Yield, ?Await]

AdditiveExpression : *AdditiveExpression* + *MultiplicativeExpression*

1. Return ? *EvaluateStringOrNumericBinaryExpression*(
AdditiveExpression, +, *MultiplicativeExpression*).

Semantics

ECMA-262 - JavaScript Language Specification

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? **RequireInternalSlot**(*set*, [[SetData]]).
3. Let *otherRecord* be ? **GetSetRecord**(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

TC
39

tc39/ecma262

Status, process, and documents for ECMA-262

TC
39

Syntax

*AdditiveExpression*_[Yield, Await] :

*MultiplicativeExpression*_[?Yield, ?Await]

*AdditiveExpression*_[?Yield, ?Await] + *MultiplicativeExpression*_[?Yield, ?Await]

*AdditiveExpression*_[?Yield, ?Await] - *MultiplicativeExpression*_[?Yield, ?Await]

AdditiveExpression : *AdditiveExpression* + *MultiplicativeExpression*

1. Return ? **EvaluateStringOrNumericBinaryExpression**(
AdditiveExpression, +, *MultiplicativeExpression*).

Semantics

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? **RequireInternalSlot**(*set*, [[SetData]]).
3. Let *otherRecord* be ? **GetSetRecord**(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }  
  ) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? **RequireInternalSlot**(*set*, [[SetData]]).
3. Let *otherRecord* be ? **GetSetRecord**(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }  
  ) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? RequireInternalSlot(*set*, [[SetData]]).
3. Let *otherRecord* be ? GetSetRecord(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }  
  ) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? RequireInternalSlot(*set*, [[SetData]]).
3. Let *otherRecord* be ? GetSetRecord(*other*).
4. If SetDataSize(*set*.[[SetData]]) < *otherRecord*.[[Size]], return **false**.
5. Let *keysIterator* be ? GetIteratorFromMethod(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? IteratorStepValue(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If SetDataHas(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? IteratorClose(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? RequireInternalSlot(*set*, [[SetData]]).
3. Let *otherRecord* be ? GetSetRecord(*other*).
4. If SetDataSize(*set*.[[SetData]]) < *otherRecord*.[[Size]], return **false**.
5. Let *keysIterator* be ? GetIteratorFromMethod(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? IteratorStepValue(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If SetDataHas(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? IteratorClose(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? RequireInternalSlot(*set*, [[SetData]]).
3. Let *otherRecord* be ? GetSetRecord(*other*).
4. If SetDataSize(*set*.[[SetData]]) < *otherRecord*.[[Size]], return **false**.
5. Let *keysIterator* be ? GetIteratorFromMethod(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? IteratorStepValue(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If SetDataHas(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? IteratorClose(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? RequireInternalSlot(*set*, [[SetData]]).
3. Let *otherRecord* be ? GetSetRecord(*other*).
4. If SetDataSize(*set*.[[SetData]]) < *otherRecord*.[[Size]], return **false**.
5. Let *keysIterator* be ? GetIteratorFromMethod(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? IteratorStepValue(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If SetDataHas(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? IteratorClose(...).
 2. Return **false**.
8. Return **true**.

Manual

TC
39

```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }) !== false;  
};
```

core-js

Current Approach - Manual Translation

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? RequireInternalSlot(*set*, [[SetData]]).
3. Let *otherRecord* be ? GetSetRecord(*other*).
4. If SetDataSize(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? GetIteratorFromMethod(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? IteratorStepValue(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If SetDataHas(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? IteratorClose(...).
 2. Return **false**.
8. Return **true**.

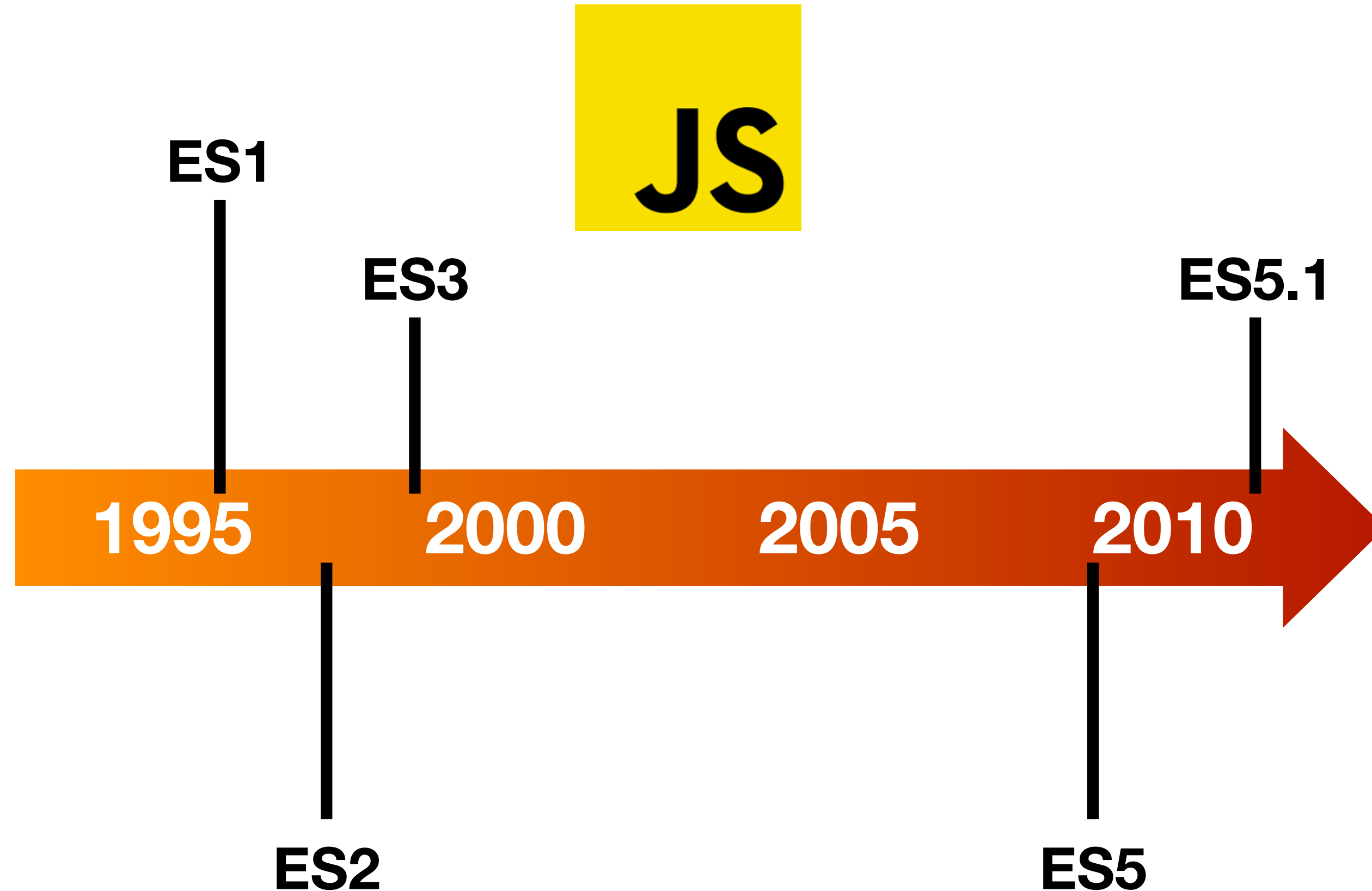
Manual

TC
39

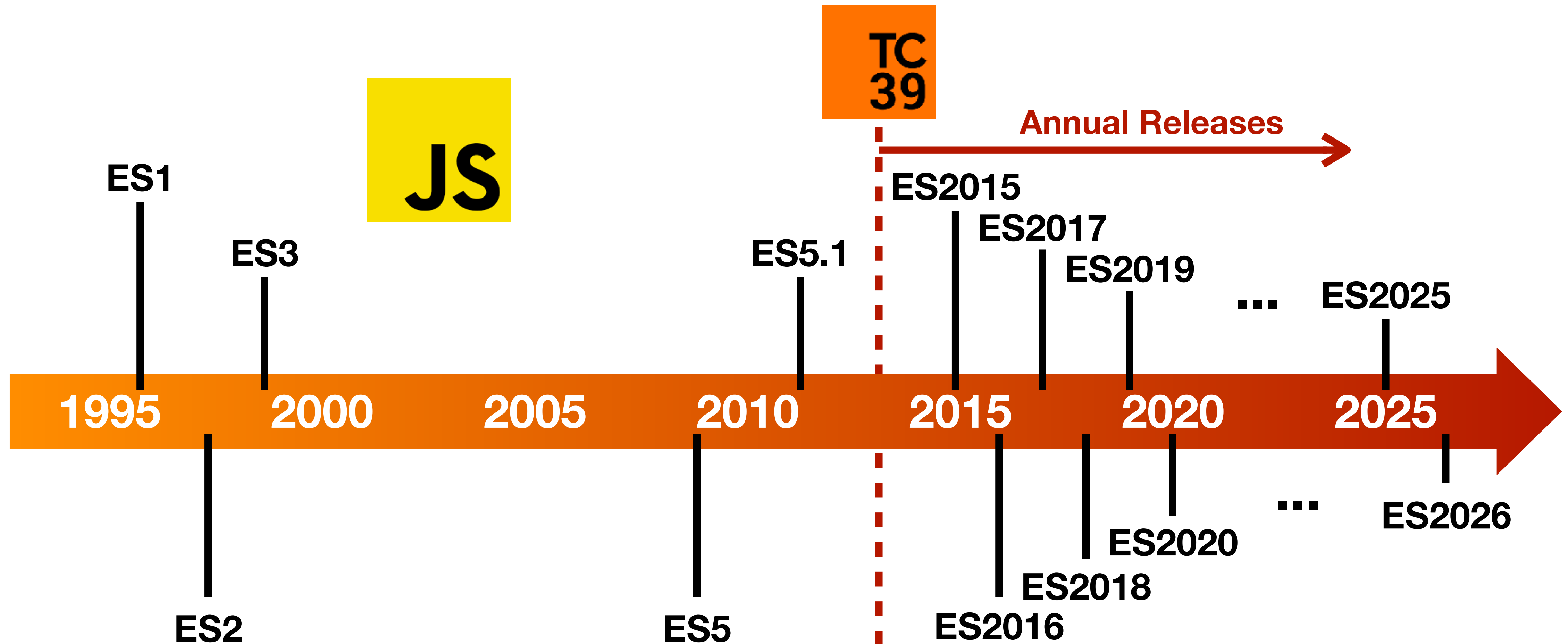
```
function isSupersetOf(other) {  
  var 0 = aSet(this);  
  var otherRec = getSetRecord(other);  
  if (size(0) < otherRec.size) return false;  
  var iterator = otherRec.getIterator();  
  return iterateSimple(iterator,  
    function (e) {  
      if (!has(0, e)) {  
        return iteratorClose(iterator,  
          'normal', false);  
      }  
    }) !== false;  
};
```

core-js

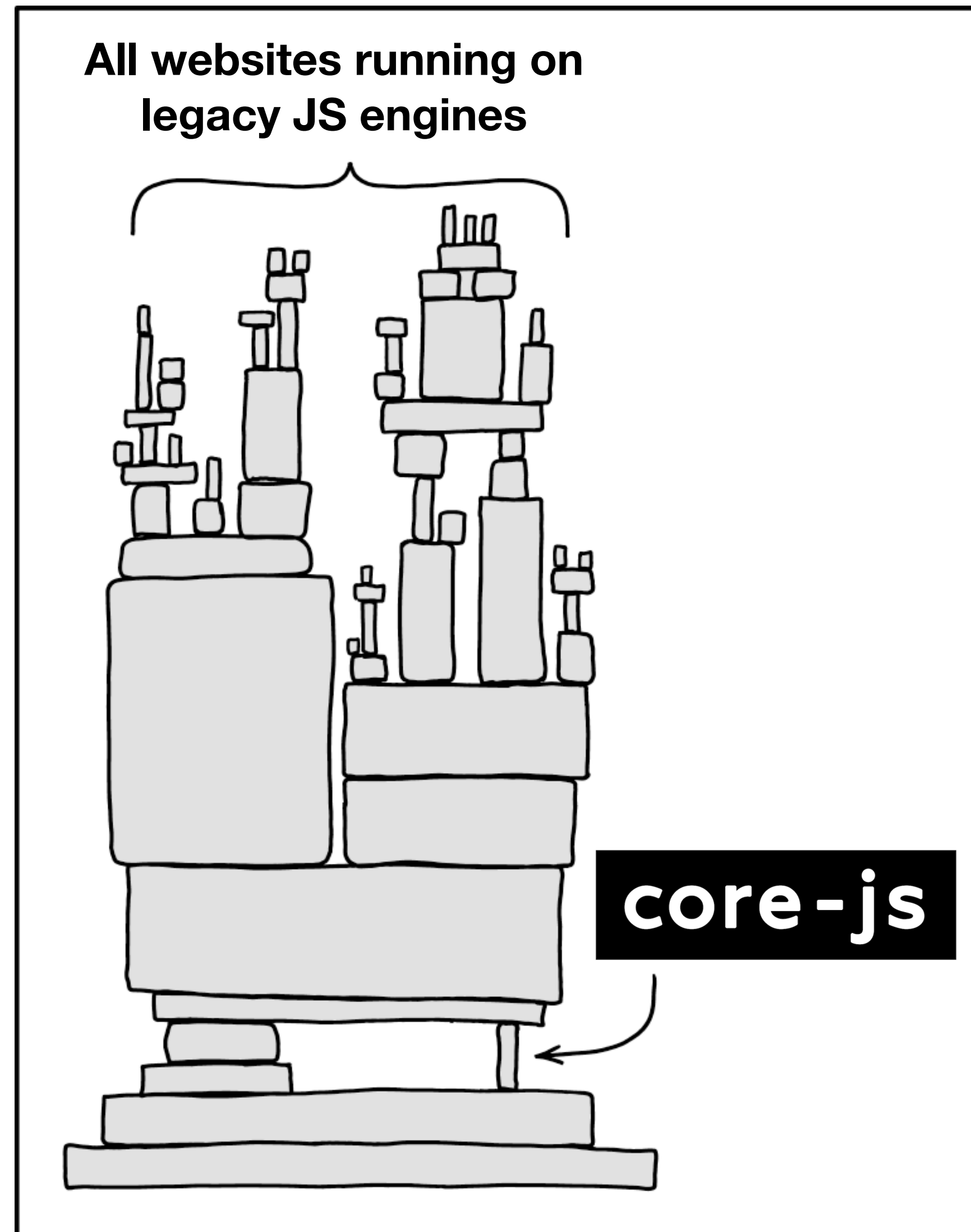
Problem - **Fast Evolving** JavaScript



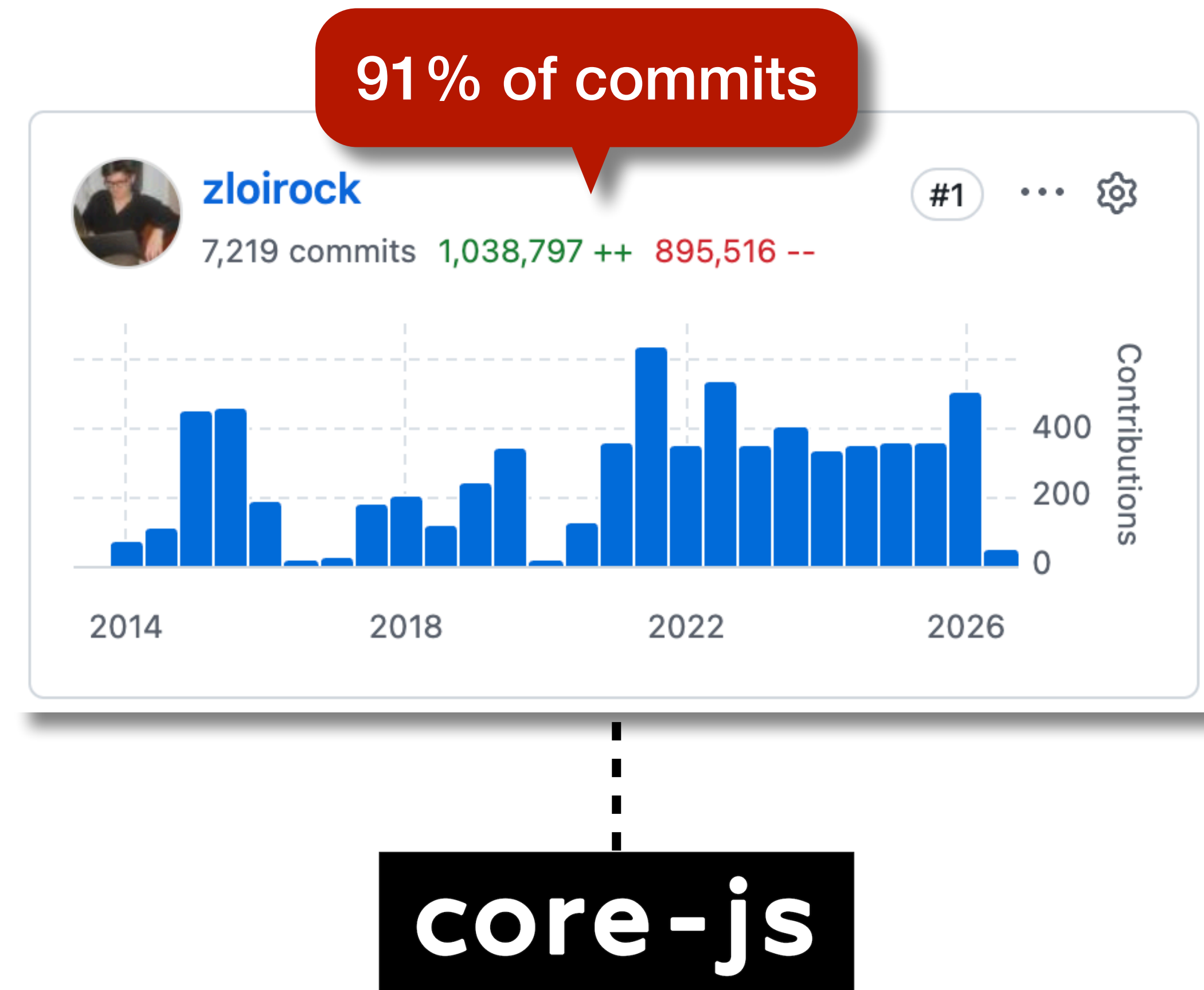
Problem - **Fast Evolving** JavaScript



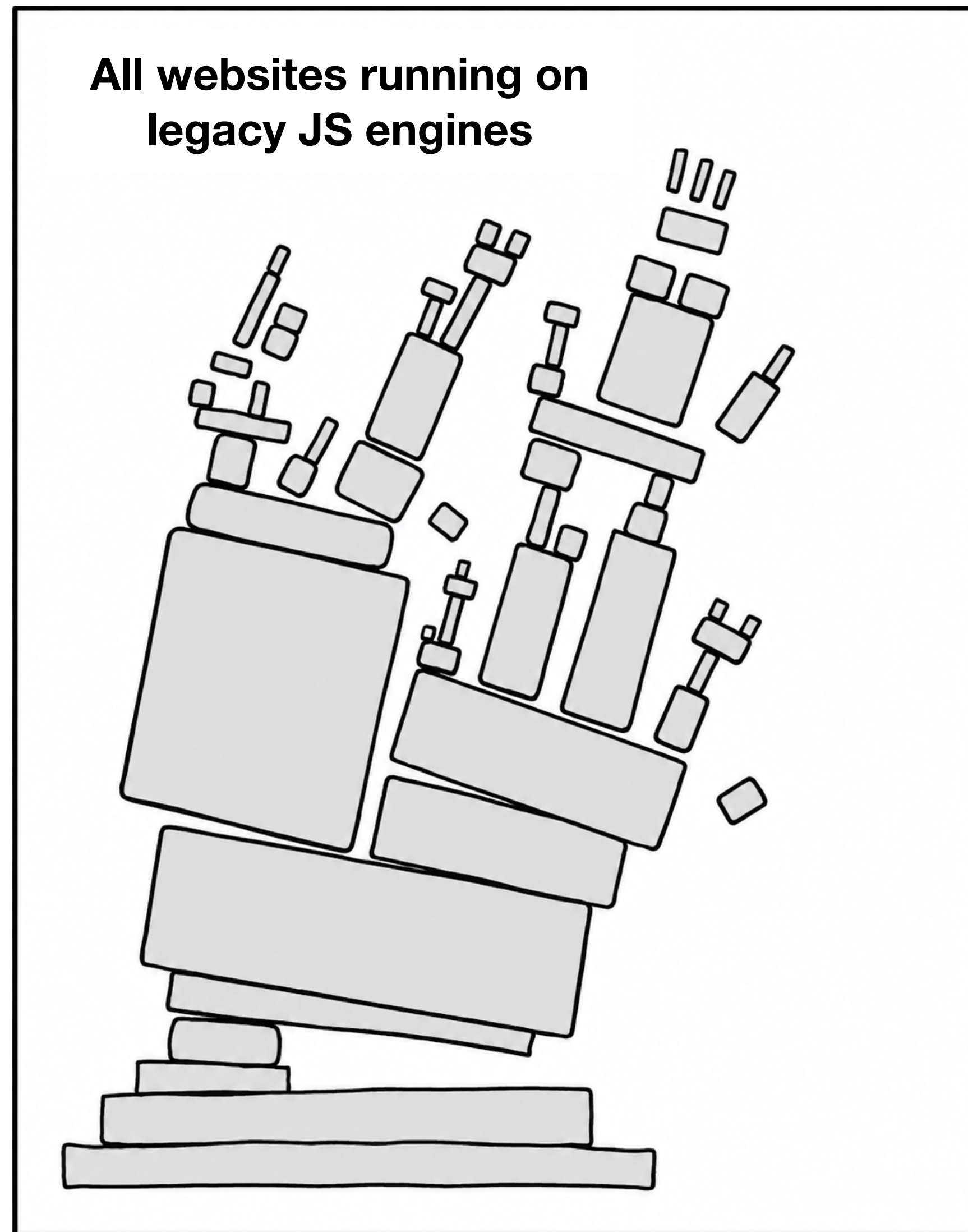
Problem - Burden of **Manual** Implementation



<https://xkcd.com/2347/> (Original, modified with AI)



Problem - Burden of **Manual** Implementation



<https://xkcd.com/2347/> (Original, modified with AI)



core-js

Our Approach - **Automatic Extraction** from ECMA-262

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? **RequireInternalSlot**(*set*, [[SetData]]).
3. Let *otherRecord* be ? **GetSetRecord**(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

Extract

TC
39

Polyfill

in **JavaScript**

Challenges - **Correctness** and **Efficiency**

Built-in

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? **RequireInternalSlot**(*set*, [[SetData]]).
3. Let *otherRecord* be ? **GetSetRecord**(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

Extract

TC
39

Polyfill

in **JavaScript**

Challenges - **Correctness** and **Efficiency**

Built-in

1) How to **correctly** handle *internal repr. of control flows*?

Set.prototype.isSupersetOf (*other*)

1. Let *set* be the **this** value.
2. Perform ? **RequireInternalSlot**(*set*, [[SetData]]).
3. Let *otherRecord* be ? **GetSetRecord**(*other*).
4. If **SetDataSize**(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? **GetIteratorFromMethod**(...).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? **IteratorStepValue**(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If **SetDataHas**(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? **IteratorClose**(...).
 2. Return **false**.
8. Return **true**.

Extract

TC
39

Polyfill

in **JavaScript**

Challenges - **Correctness** and **Efficiency**

Built-in

`Set.prototype.isSupersetOf ( other )`

1. Let *set* be the **this** value.
2. Perform ? `RequireInternalSlot`(*set*, [[SetData]]).
3. Let *otherRecord* be ? `GetSetRecord`(*other*).
4. If `SetDataSize`(*set*.[[SetData]]) < *otherRecord*.[[Size]],
return **false**.
5. Let *keysIterator* be ? `GetIteratorFromMethod`(*set*, `[[SetData]].keys`).
6. Let *next* be **NOT-STARTED**.
7. Repeat, while *next* is not **DONE**,
 - a. Set *next* to ? `IteratorStepValue`(*keysIterator*).
 - b. If *next* is not **DONE**, then
 - i. If `SetDataHas`(*set*.[[SetData]], *next*) is **false**, then
 1. Perform ? `IteratorClose`(...).
 2. Return **false**.
8. Return **true**.

1) How to **correctly** handle *internal repr. of control flows*?

2) How to extract *algorithmically efficient* polyfill?

Extract

TC
39

Polyfill

in **JavaScript**

Challenge 1 - **Correctness** (Internal Repr. of Control Flow)

A computation **X** $\left\{ \begin{array}{ll} \text{is **normally** terminated with a **value**} & \text{if **X** is a **normal completion**} \\ \text{throws an **exception**} & \text{if **X** is a **throw completion**} \end{array} \right.$

IteratorClose (*iteratorRecord*, *completion*)

...

3. Let *innerResult* be **GetMethod**(...).

4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*.

c. Set *innerResult* to **Call**(...).

5. If *completion* is a **throw completion**, return ? *completion*.

6. If *innerResult* is a **throw completion**, return ? *innerResult*.

...

8. Return ? *completion*.

**TC
39**

Challenge 1 - **Correctness** (Internal Repr. of Control Flow)

A computation **X** $\left\{ \begin{array}{ll} \text{is **normally** terminated with a **value**} & \text{if **X** is a **normal completion**} \\ \text{throws an **exception**} & \text{if **X** is a **throw completion**} \end{array} \right.$

IteratorClose (*iteratorRecord*, *completion*)

...

3. Let *innerResult* be **GetMethod**(...).

4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*.

c. Set *innerResult* to **Call**(...).

5. If *completion* is a **throw completion**, return ? *completion*.

6. If *innerResult* is a **throw completion**, return ? *innerResult*.

...

8. Return ? *completion*.

return *completion*
OR
throw *completion*

TC
39

Challenge 1 - **Correctness** (Internal Repr. of Control Flow)

A computation **X** $\left\{ \begin{array}{ll} \text{is **normally** terminated with a **value**} & \text{if } \mathbf{X} \text{ is a **normal completion**} \\ \text{throws an **exception**} & \text{if } \mathbf{X} \text{ is a **throw completion**} \end{array} \right.$

IteratorClose (*iteratorRecord*, *completion*)

...

3. Let *innerResult* be **GetMethod**(...).

4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*.

c. Set *innerResult* to **Call**(...).

5. If *completion* is a **throw completion**, return ? *completion*.

6. If *innerResult* is a **throw completion**, return ? *innerResult*.

...

8. Return ? *completion*.

return *completion*

OR

throw *completion*

throw *completion*

throw *innerResult*

**TC
39**

Challenge 1 - **Correctness** (Internal Repr. of Control Flow)

A computation **X** $\left\{ \begin{array}{ll} \text{is **normally** terminated with a **value**} & \text{if } \mathbf{X} \text{ is a **normal completion**} \\ \text{throws an **exception**} & \text{if } \mathbf{X} \text{ is a **throw completion**} \end{array} \right.$

IteratorClose (*iteratorRecord*, *completion*)

...

3. Let *innerResult* be **GetMethod**(...).

4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*.

c. Set *innerResult* to **Call**(...).

5. If *completion* is a **throw completion**, return ? *completion*.

6. If *innerResult* is a **throw completion**, return ? *innerResult*.

...

8. Return ? *completion*.

return *completion*

OR

throw *completion*

throw *completion*

throw *innerResult*

return *completion*

TC
39

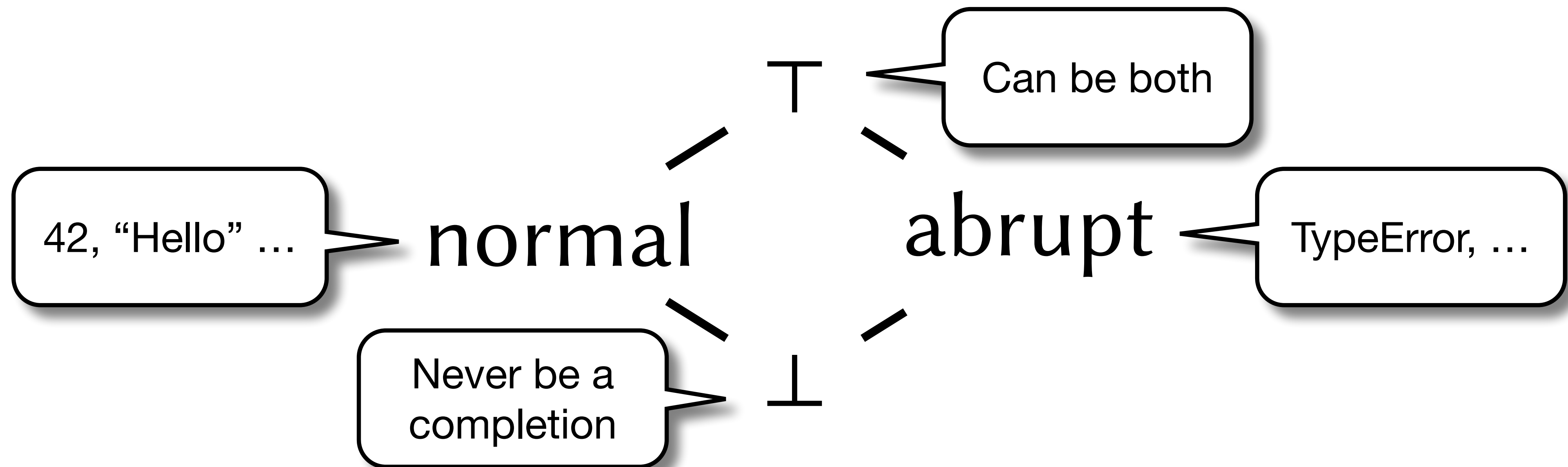
Solution 1 - Completion Type Analysis

IteratorClose (*iteratorRecord*, *completion*)

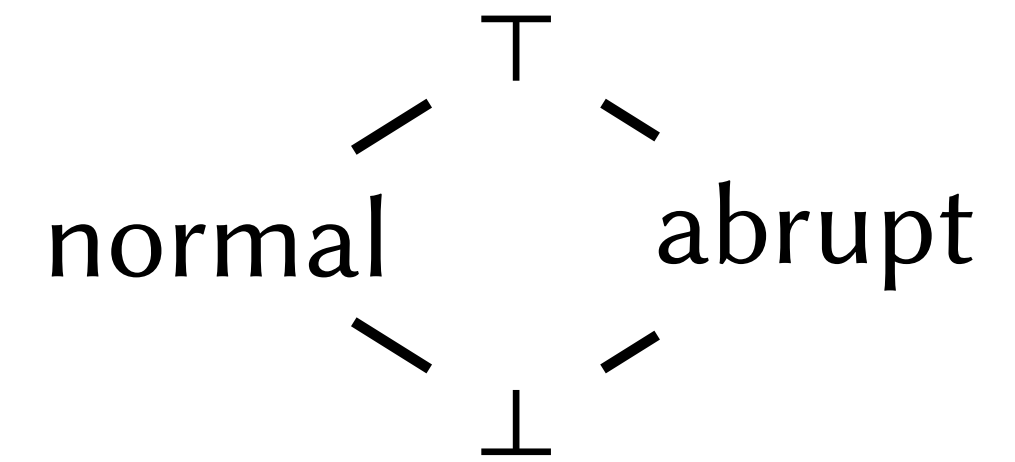
...

3. Let *innerResult* be **GetMethod**(...).

TC
39



Solution 1 - Completion Type Analysis



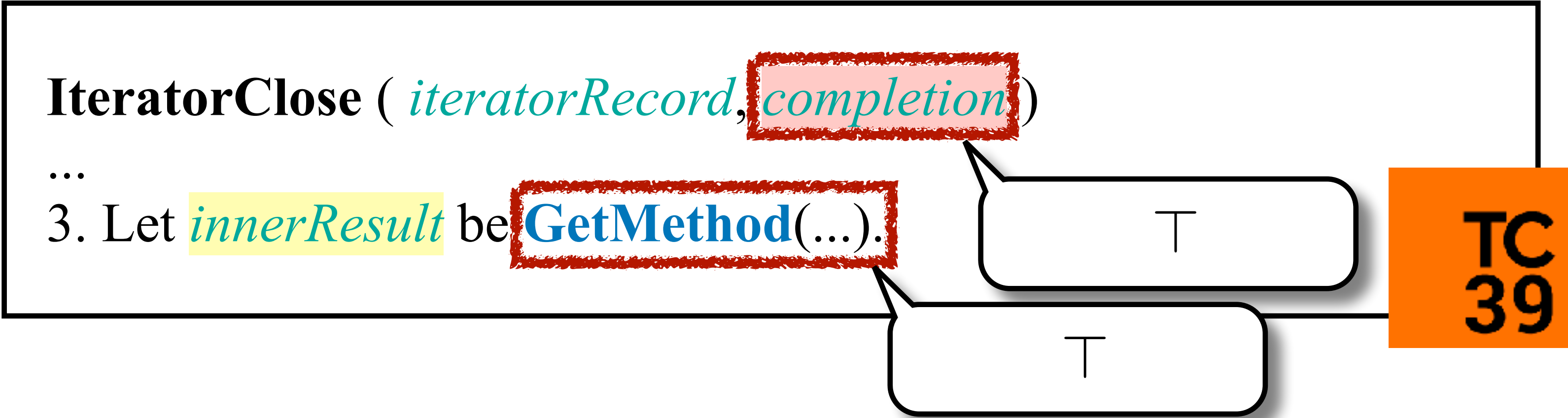
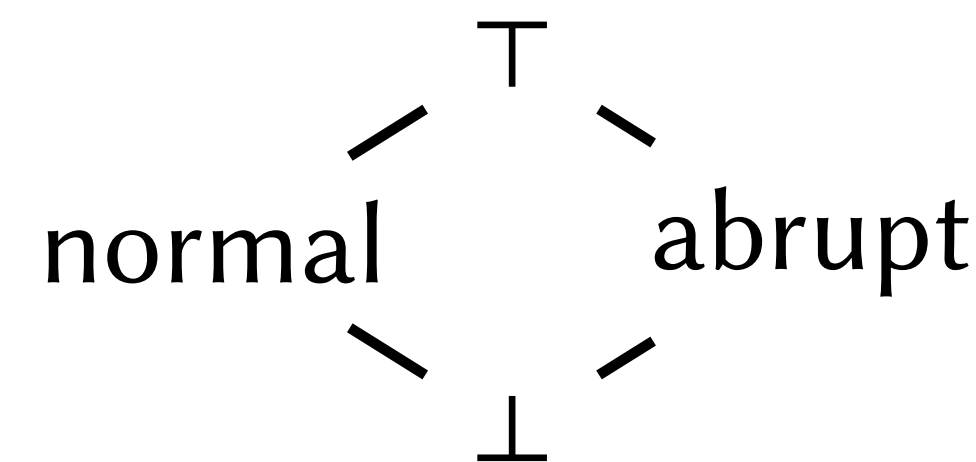
IteratorClose (*iteratorRecord*, *completion*)

...

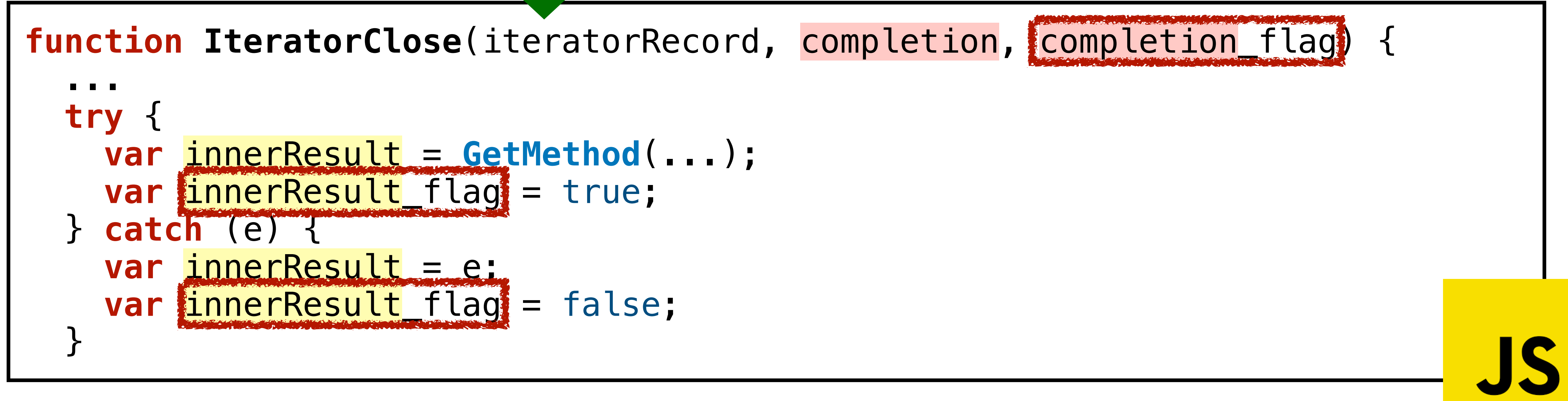
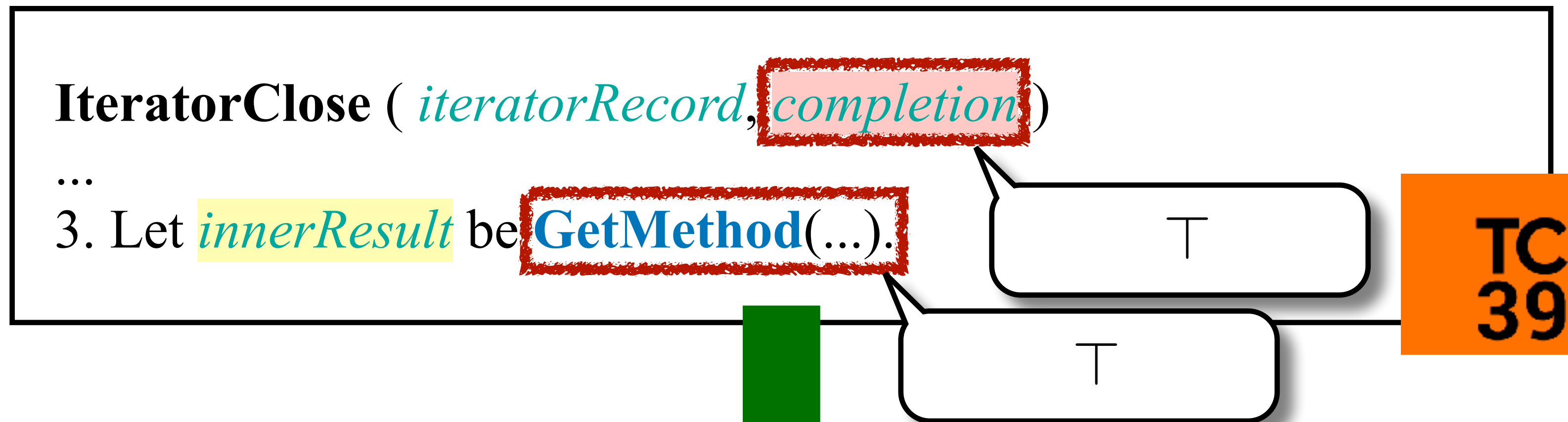
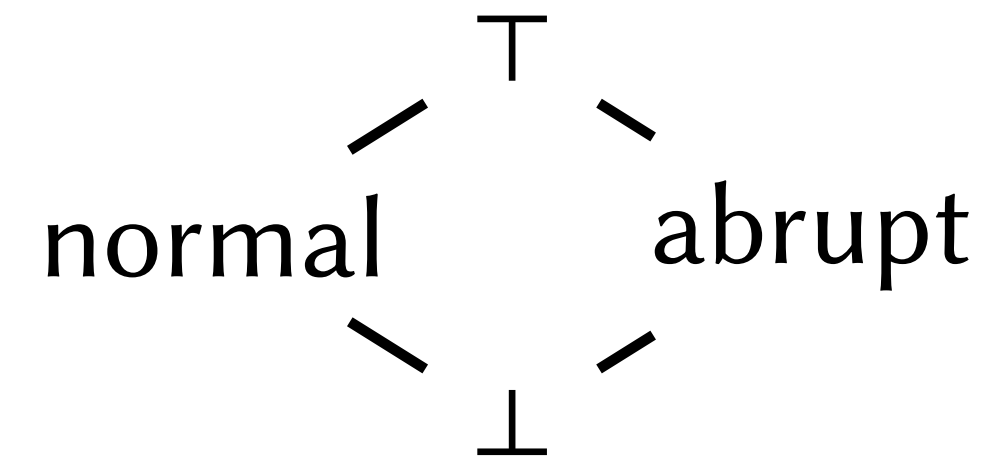
3. Let *innerResult* be **GetMethod**(...).

TC
39

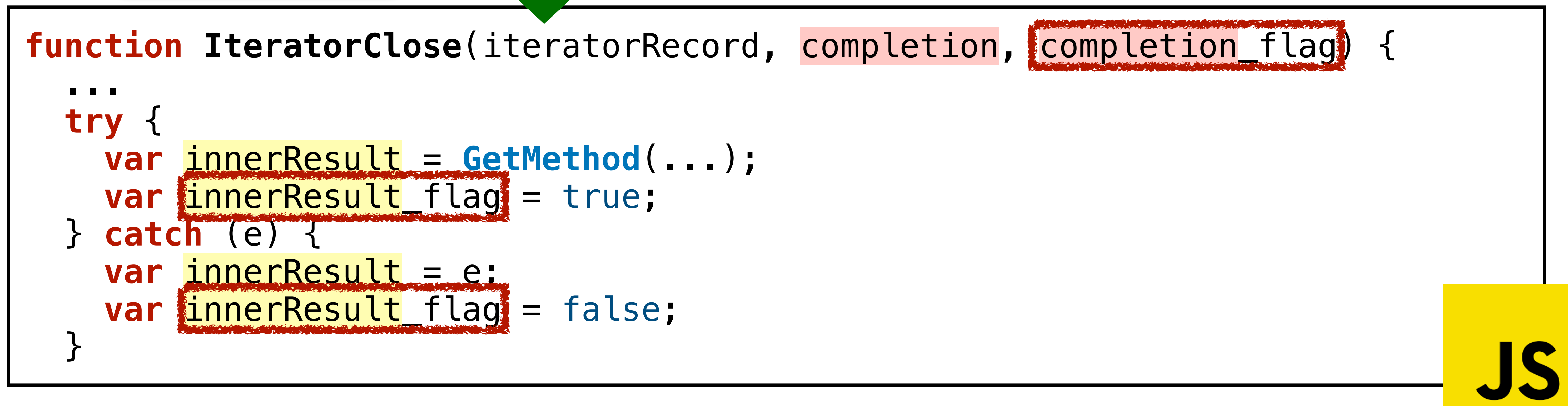
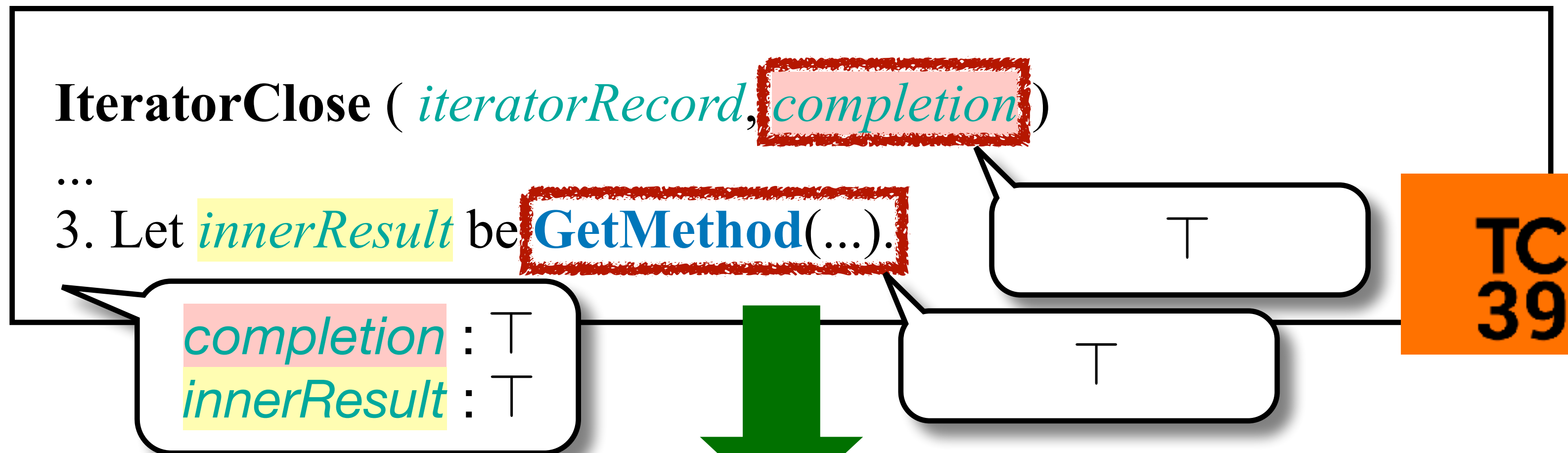
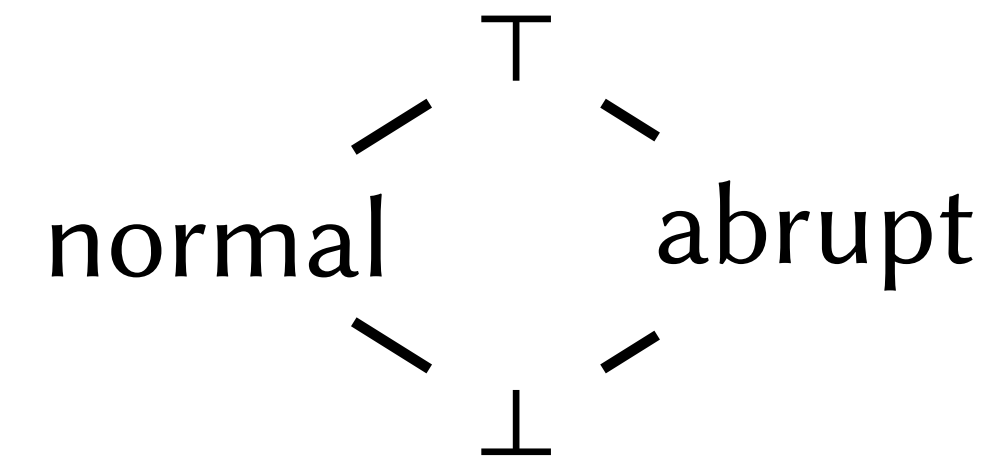
Solution 1 - Completion Type Analysis



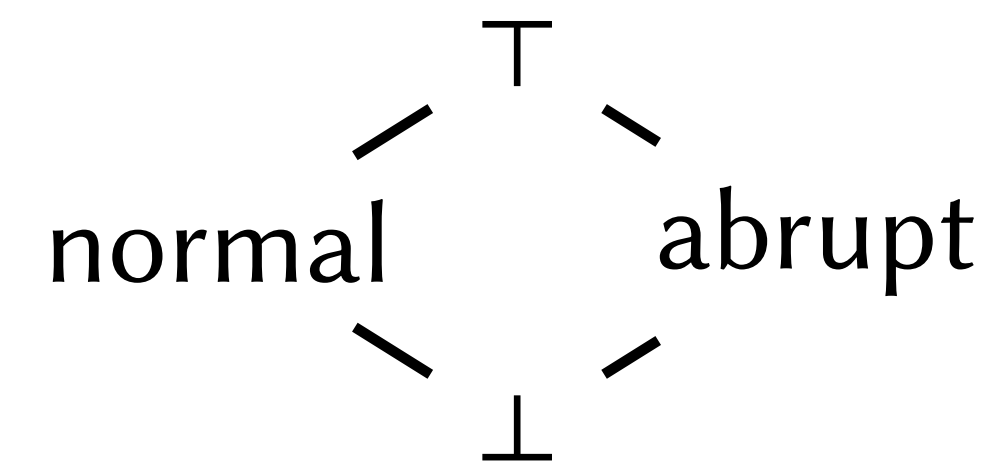
Solution 1 - Completion Type Analysis



Solution 1 - Completion Type Analysis



Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

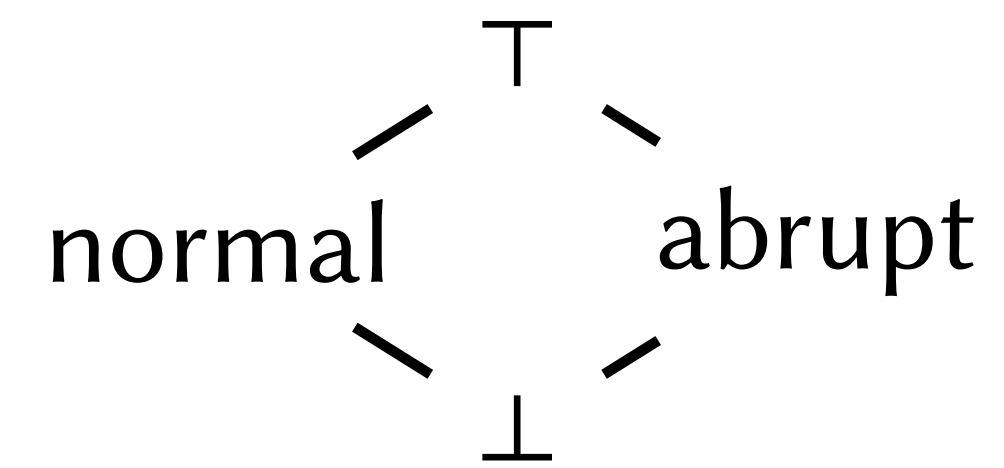
...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call**(...).

TC
39

Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

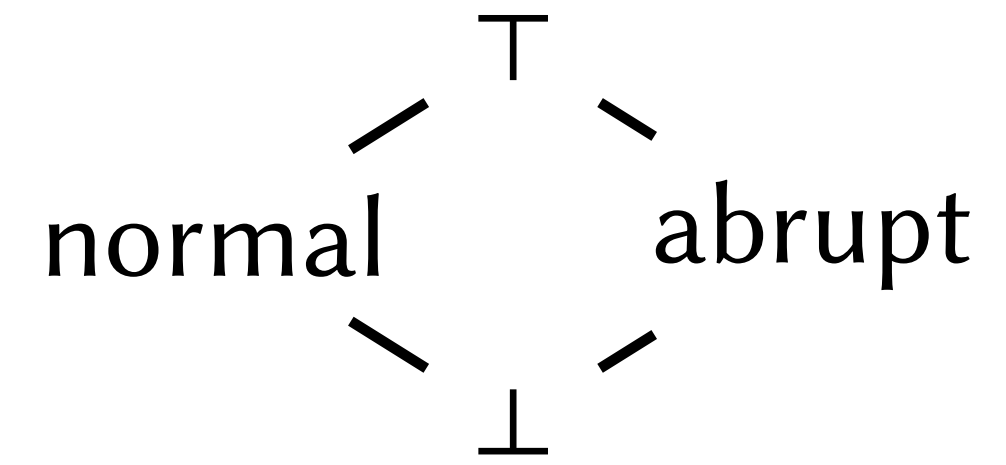
...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call**(...).

TC
39

Solution 1 - Completion Type Analysis



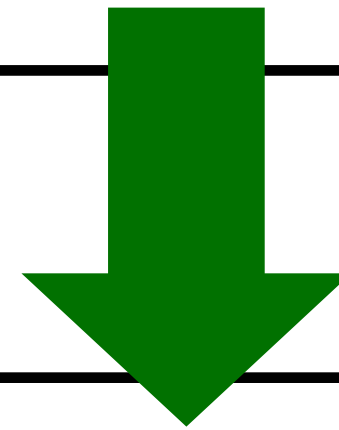
4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call**(...).

TC
39



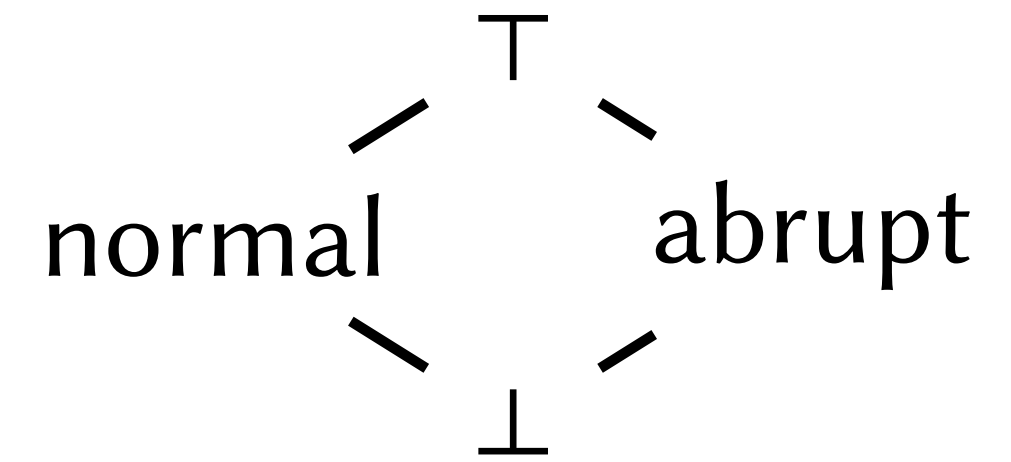
```
if (innerResult_flag) {
```

```
...
```

```
}
```

JS

Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call**(...).

completion : T
innerResult : normal

TC
39

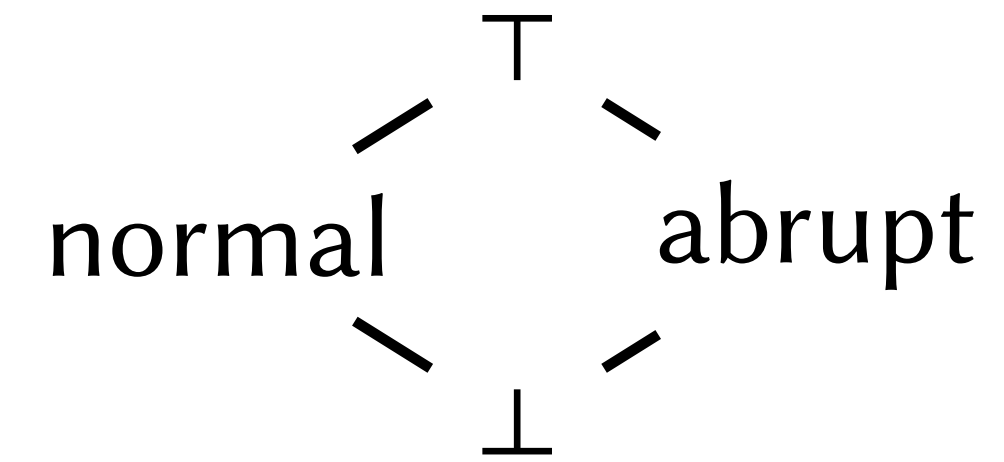
```
if (innerResult_flag) {
```

```
...
```

```
}
```

JS

Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

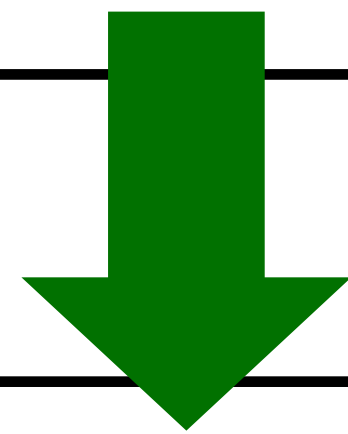
...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call**(...).

completion : T
innerResult : normal

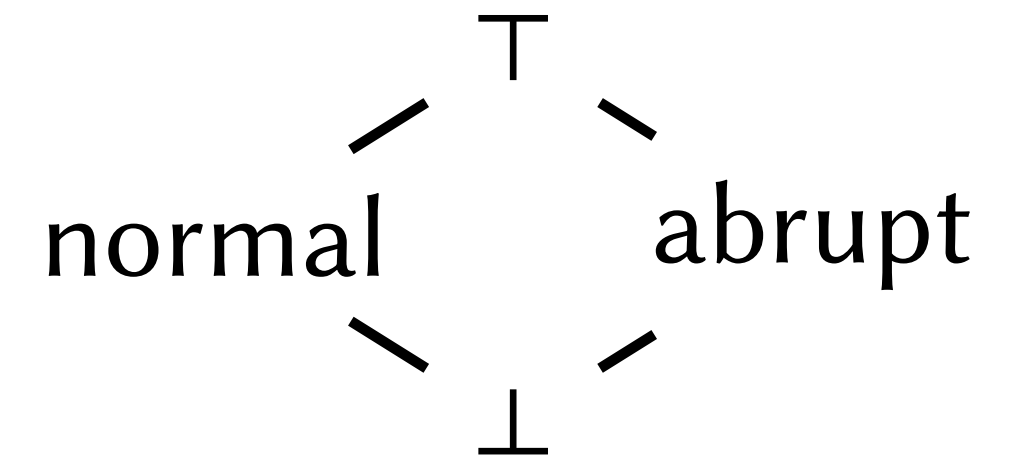
TC
39



```
if (innerResult_flag) {  
  ...  
  if ( return === undefined) {  
    if (completion_flag) return completion;  
    else throw completion;  
  }  
}
```

JS

Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call(...)**

completion : T
innerResult : normal

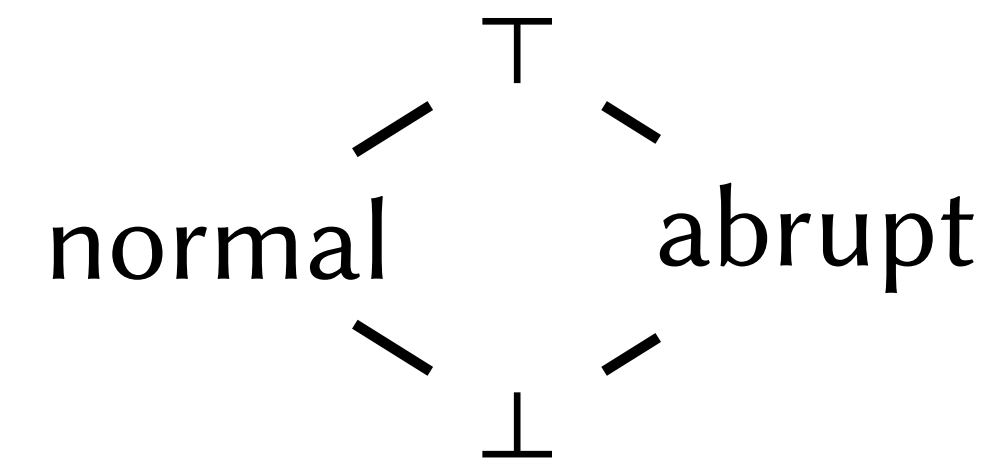
TC
39

T

```
if (innerResult_flag) {  
  ...  
  if ( return === undefined) {  
    if (completion_flag) return completion;  
    else throw completion;  
  }  
}
```

JS

Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call(...)**

completion : T
innerResult : normal

TC
39

T

```
if (innerResult_flag) {
```

```
...
```

```
if ( return === undefined) {
```

```
  if (completion_flag) return completion;
```

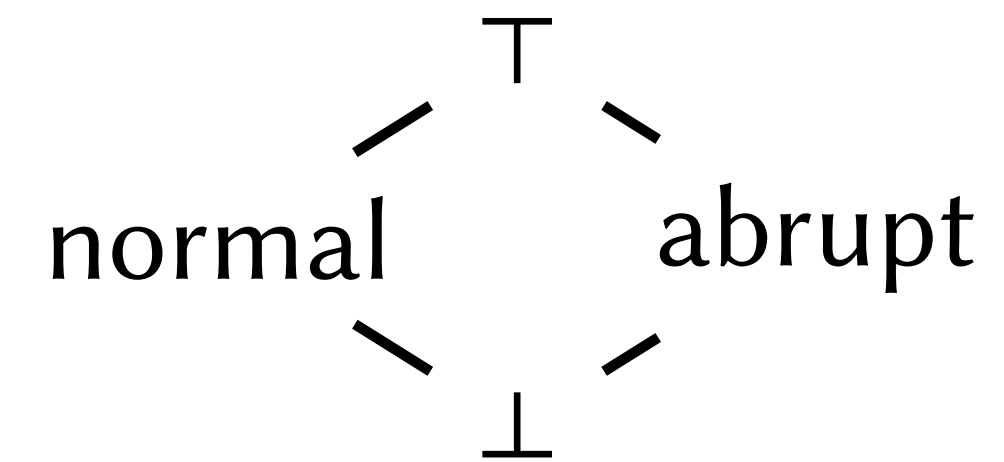
```
  else throw completion;
```

```
}
```

```
try { innerResult = Call(...); innerResult_flag = true; } catch ...
```

JS

Solution 1 - Completion Type Analysis



4. If *innerResult* is a **normal completion**, then

...

b. If *return* is undefined, return ? *completion*

c. Set *innerResult* to **Call(...)**

completion : T
innerResult : normal

TC
39

innerResult : T

T

```
if (innerResult_flag) {
```

```
...
```

```
if ( return === undefined) {
```

```
  if (completion_flag) return completion;
```

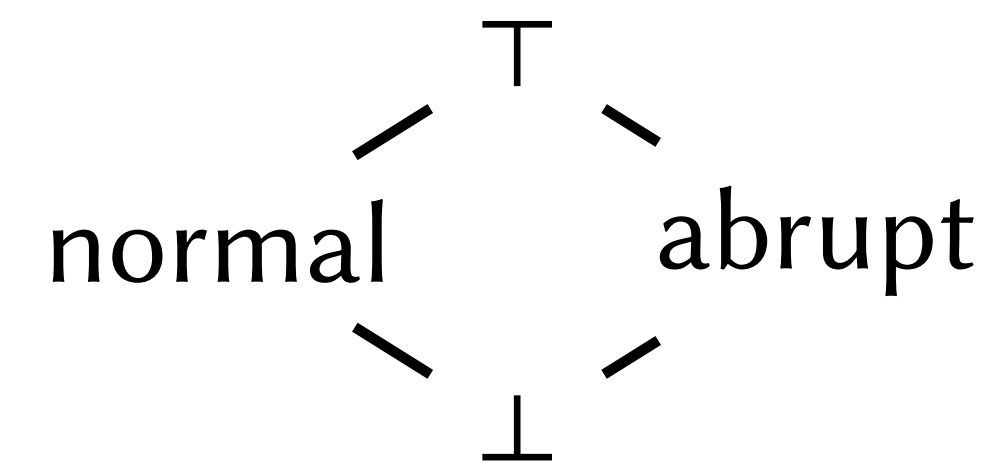
```
  else throw completion;
```

```
}
```

```
try { innerResult = Call(...); innerResult_flag = true; } catch ...
```

JS

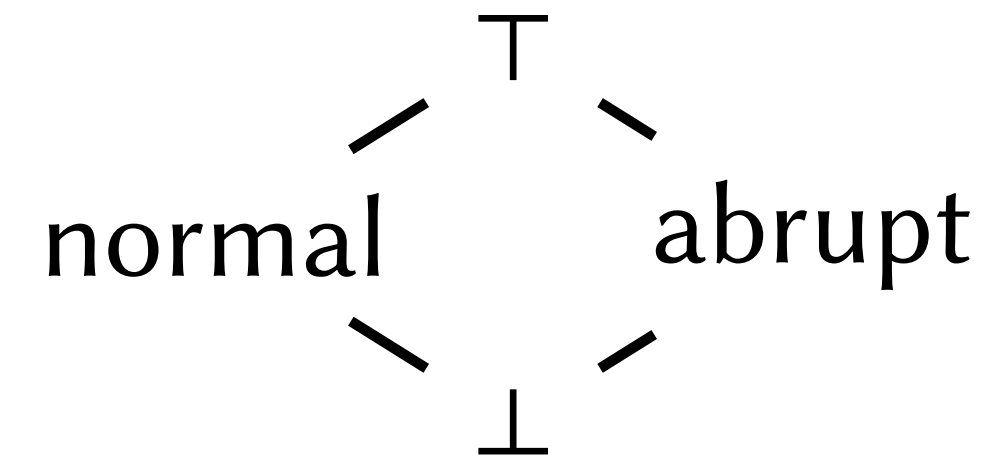
Solution 1 - Completion Type Analysis



5. If *completion* is a **throw completion**, return ? *completion*.
6. If *innerResult* is a **throw completion**, return ? *innerResult*.
- ...
8. Return ? *completion*.

TC
39

Solution 1 - Completion Type Analysis

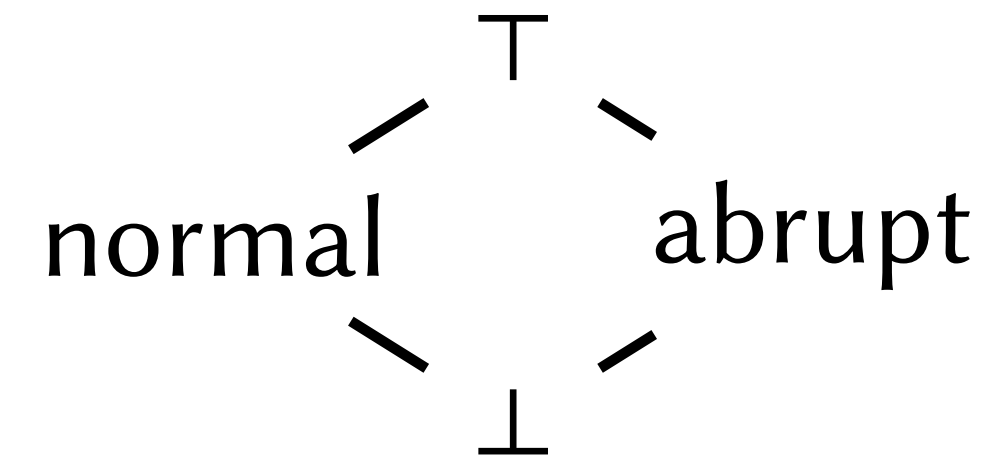


5. If *completion* is a **throw completion**, return ? *completion*.
6. If *innerResult* is a **throw completion**, return ? *innerResult*.
- ...
8. Return ? *completion*.

completion : abrupt

innerResult : abrupt

Solution 1 - Completion Type Analysis

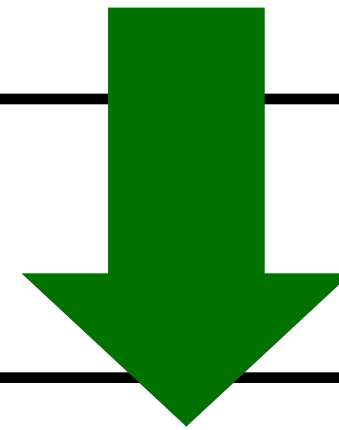


5. If *completion* is a **throw completion**, return ? *completion*.
6. If *innerResult* is a **throw completion**, return ? *innerResult*.
- ...
8. Return ? *completion*.

completion : abrupt

innerResult : abrupt

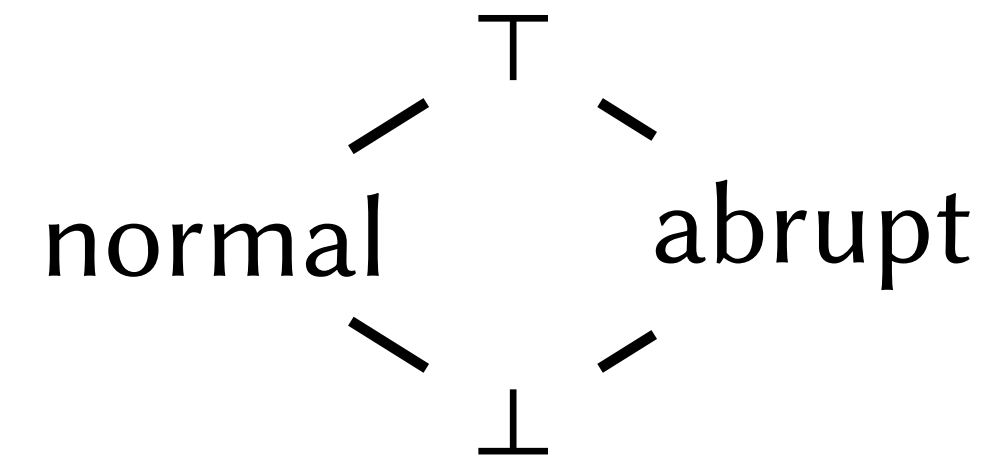
TC
39



```
if (!completion_flag) throw completion;  
if (!innerResult_flag) throw innerResult;
```

JS

Solution 1 - Completion Type Analysis

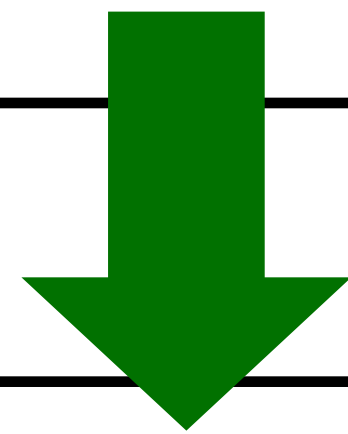


5. If *completion* is a **throw completion**, return ? *completion*.
6. If *innerResult* is a **throw completion**, return ? *innerResult*.
- ...
8. Return ? *completion*.

completion : abrupt

innerResult : abrupt

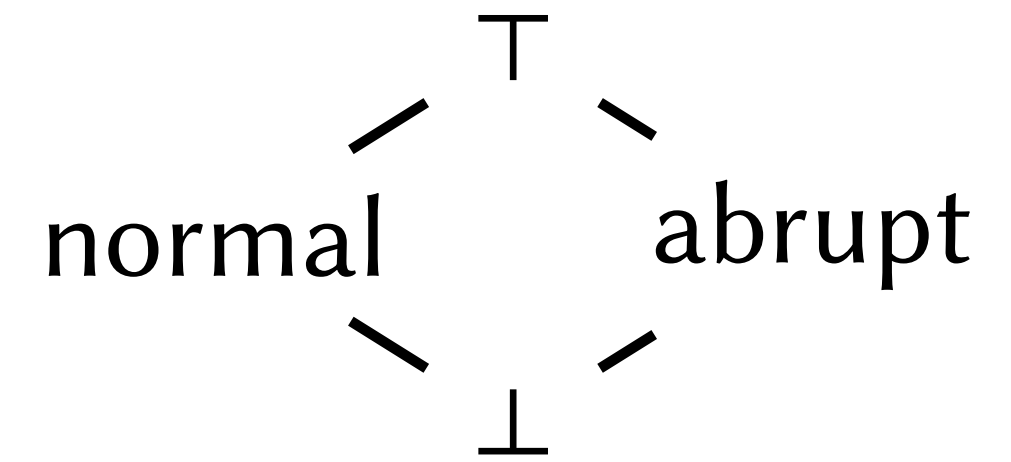
completion : normal



```
if (!completion_flag) throw completion;  
if (!innerResult_flag) throw innerResult;
```

JS

Solution 1 - Completion Type Analysis

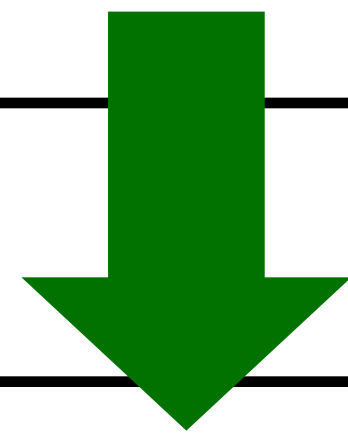


5. If *completion* is a **throw completion**, return ? *completion*.
6. If *innerResult* is a **throw completion**, return ? *innerResult*.
- ...
8. Return ? *completion*.

completion : abrupt

innerResult : abrupt

completion : normal



```
if (!completion_flag) throw completion;  
if (!innerResult_flag) throw innerResult;  
...  
return completion;
```

JS

Solution 1 - Completion Type Analysis

```
function IteratorClose(iteratorRecord, completion, completion_flag) {  
    ...  
    try { var innerResult = GetMethod(...); var innerResult_flag = true; }  
    catch (e) { var innerResult = e; var innerResult_flag = false; }  
    if (innerResult_flag) {  
        ...  
        if (_return === undefined) {  
            if (completion_flag) return completion;  
            else throw completion;  
        }  
        try { innerResult = Call(...); innerResult_flag = true; } catch ...  
    }  
    if (!completion_flag) throw completion;  
    if (!innerResult_flag) throw innerResult;  
    ...  
    return completion;  
}
```

JS

Challenge 2 - **Efficiency**

Set.prototype.add (*value*)

...

4. For each element *e* of *S*.[[SetData]], do
 - a. If *e* is not **EMPTY** and **SameValue**(*e*, *value*) is **true**, then

...

TC
39

Challenge 2 - Efficiency

Set.prototype.add (*value*)

Linear search

TC
39

...

4. For each element *e* of *S*.[[SetData]], do

a. If *e* is not **EMPTY** and SameValue(*e*, *value*) is **true**, then

...

Challenge 2 - Efficiency

$$\mathcal{O}(n)$$

⋮

Set.prototype.add (*value*)

Linear search

TC
39

...

4. For each element *e* of *S*.[[SetData]], do

a. If *e* is not **EMPTY** and SameValue(*e*, *value*) is **true**, then

...

Challenge 2 - Efficiency

$$\mathcal{O}(n)$$

vs.

$$\mathcal{O}(1)$$

Set.prototype.add (*value*)

Linear search

TC
39

...

4. For each element *e* of *S*.[[SetData]], do

a. If *e* is not **EMPTY** and **SameValue**(*e*, *value*) is **true**, then

...

core-js

Set.prototype.add

Challenge 2 - Efficiency

$$\mathcal{O}(n)$$

vs.

$$\mathcal{O}(1)$$

Set.prototype.add (*value*)

Linear search

TC
39

...
4. For each element *e* of *S*.[[SetData]], do
a. If *e* is not EMPTY and SameValue(*e*, *value*) is true, then

Set.prototype.delete (*value*)

...
4. For each element *e* of *S*.[[SetData]], do
a. If *e* is not EMPTY and SameValue(*e*, *value*) is true, then

Set.prototype.has (*value*)

...
4. For each element *e* of *S*.[[SetData]], do
a. If *e* is not EMPTY and SameValue(*e*, *value*) is true, then
...

core-js

Set.prototype.add

Set.prototype.delete

Set.prototype.has

Solution 2 - Rewrite Rules

```
pattern: |
  1. For each element {{ elem : var }} of {{ base : ref }}, do
    1. If {{ elem }} is not ~empty~ and SameValue({{ elem }},
      {{ value : var }}) is *true*, then
      1. {{ body : step }}
```

(* simplified)

`Set.prototype.add ( value )`

...

4. For each element *e* of `S.[[SetData]]`, do
 - a. If *e* is not **EMPTY** and `SameValue(e, value)` is **true**, then
 - ...

TC
39

Solution 2 - Rewrite Rules

```
pattern: |
  1. For each element {{ elem : var }} of {{ base : ref }}, do
    1. If {{ elem }} is not ~empty~ and SameValue({{ elem }},
      {{ value : var }}) is *true*, then
        1. {{ body : step }}
where:
  - isSetData(base)
  - { isSetData: "_.[[SetData]].copy*" }
```

(* simplified)

`Set.prototype.add ( value )`

...

4. For each element *e* of `S.[[SetData]]`, do

a. If *e* is not **EMPTY** and `SameValue(e, value)` is **true**, then

...

TC
39

Solution 2 - Rewrite Rules

pattern: |

1. For each element `{{ elem : var }}` of `{{ base : ref }}`, do
 1. If `{{ elem }}` is not `~empty~` and `SameValue({{ elem }}, {{ value : var }})` is `*true*`, then
 1. `{{ body : step }}`

where:

- `isSetData(base)`
- `{ isSetData: "_.[[SetData]].copy*" }`

(* simplified)

`Set.prototype.add ( value )`

...

4. For each element `e` of `S.{{SetData}}` do
 - a. If `e` is not **EMPTY** and `SameValue(e, value)` is **true**, then
 - ...

TC
39

Solution 2 - Rewrite Rules

pattern: |

1. For each element `{{ elem : var }}` of `{{ base : ref }}`, do
 1. If `{{ elem }}` is not `~empty~` and `SameValue({{ elem }}, {{ value : var }})` is `*true*`, then
 1. `{{ body : step }}`

where:

- `isSetData(base)`
- `{ isSetData: " _.[[SetData]].copy* " }`

replace: |

1. If `SET_DATA_HAS({{ base }}, {{ value }})` is `*true*`, then
 1. `{{ body }}`

(* simplified)

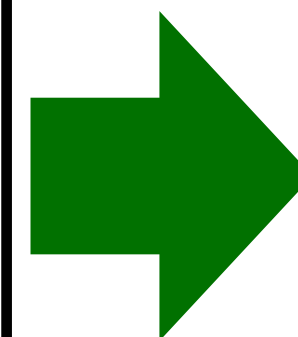
`Set.prototype.add ( value )`

...

4. For each element `e` of `S.[[SetData]]`, do
 - a. If `e` is not **EMPTY** and `SameValue(e, value)` is **true**, then

...

TC
39



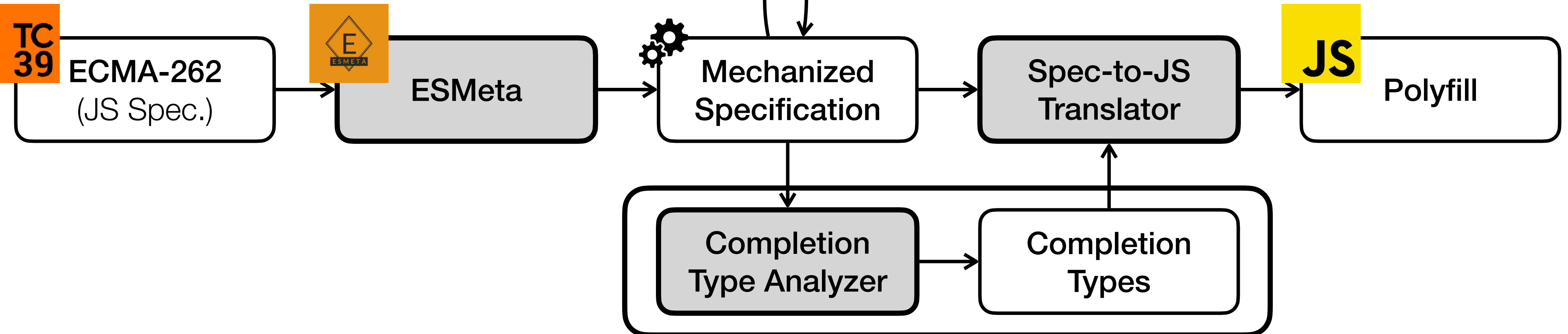
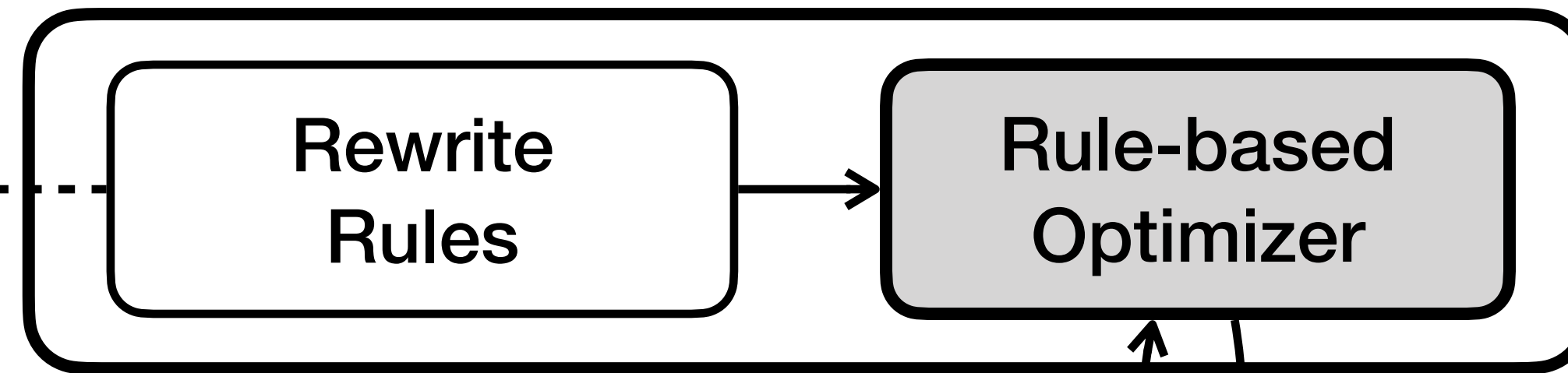
```
function add(value) {  
  ...  
  if (SET_DATA_HAS(S.SetData, value)  
      === true) {  
    ...  
  }  
}
```

JS

MiniPoly - Overall Structure

Solution 2: Rewrite Rules

 Polyfill Developer



Solution 1: Correctness

Evaluation

- Target **89 built-in APIs** in **ES2025**
- **Baselines** - `core-js` and `es-shims` (final release versions of 2025)
- Research Questions

RQ1) Correctness

RQ2) Reduced Manual Effort

RQ3) Speed

RQ4) Forward Compatibility

RQ5) Real-World Usability

RQ1) Correctness

- Tested by **Test262** - official conformance tests
- **(# of Fail) core-js - 344 / es-shims - 618** (of 6,785 supported) / **MiniPoly - 16**

Category	# Built-ins	Tests	core-js		es-shims [†]		MiniPoly		MiniPoly-opt	
			Pass (%)	Time	Pass (%)	Time	Pass (%)	Time	Pass (%)	Time
Array	24	4,713	4,617 (98.0)	26.4s	4,355 (94.9)	48.5s	4,713 (100.0)	25.4s	4,713 (100.0)	25.6s
String	12	638	616 (96.6)	6.2s	602 (94.4)	6.6s	638 (100.0)	6.0s	638 (100.0)	6.0s
Set	18	748	710 (94.9)	8.5s	682 (91.7)	9.4s	746 (99.7)	8.2s	746 (99.7)	8.4s
Map	12	322	298 (92.5)	4.8s	256 (79.5)	5.0s	320 (99.4)	4.7s	320 (99.4)	4.6s
WeakMap	5	198	168 (84.8)	2.3s	–	–	196 (99.0)	2.2s	196 (99.0)	2.3s
WeakSet	5	166	142 (85.5)	2.2s	–	–	164 (98.8)	2.1s	164 (98.8)	2.1s
Promise	13	1,264	1,154 (91.3)	9.8s	272 (55.5)	19.8s	1,256 (99.4)	9.6s	1,256 (99.4)	9.6s
Total	89	8,049	7,705 (95.7)	60.1s	6,167 (90.9)	89.3s	8,033 (99.8)	58.1s	8,033 (99.8)	58.4s

[†] es-shims evaluated on supported subset, and (–) indicates unsupported built-ins.

RQ1) Correctness

- Tested by **Test262** - official conformance tests
- (# of Fail) **core-js** - 344 / **es-shims** - 618 (of 6,785 supported) / **MiniPoly** - 16

Category	# Built-ins	Tests	core-js		es-shims [†]		MiniPoly		MiniPoly-opt	
			Pass (%)	Time	Pass (%)	Time	Pass (%)	Time	Pass (%)	Time
Array	24	4,713	4,617 (98.0)	26.4s	4,355 (94.9)	48.5s	4,713 (100.0)	25.4s	4,713 (100.0)	25.6s
String	12	638	616 (96.6)	6.2s	602 (94.4)	6.6s	638 (100.0)	6.0s	638 (100.0)	6.0s
Set	18	748	710 (94.9)	8.5s	682 (91.7)	9.4s	746 (99.7)	8.2s	746 (99.7)	8.4s
Map	12	322	298 (92.5)	4.8s	256 (79.5)	5.0s	320 (99.4)	4.7s	320 (99.4)	4.6s
WeakMap	5	198	168 (84.8)	2.3s	–	–	196 (99.0)	2.2s	196 (99.0)	2.3s
WeakSet	5	166	142 (85.5)	2.2s	–	–	164 (98.8)	2.1s	164 (98.8)	2.1s
Promise	13	1,264	1,154 (91.3)	9.8s	272 (55.5)	19.8s	1,256 (99.4)	9.6s	1,256 (99.4)	9.6s
Total	89	8,049	7,705 (95.7)	60.1s	6,167 (90.9)	89.3s	8,033 (99.8)	58.1s	8,033 (99.8)	58.4s

[†] es-shims evaluated on supported subset, and (–) indicates unsupported built-ins.

RQ1) Correctness

- Tested by **Test262** - official conformance tests
- (# of Fail) **core-js - 344** / **es-shims - 618** (of 6,785 supported) / **MiniPoly - 16**

Category	# Built-ins	Tests	core-js		es-shims [†]		MiniPoly		MiniPoly-opt	
			Pass (%)	Time	Pass (%)	Time	Pass (%)	Time	Pass (%)	Time
Array	24	4,713	4,617 (98.0)	26.4s	4,355 (94.9)	48.5s	4,713 (100.0)	25.4s	4,713 (100.0)	25.6s
String	12	638	616 (96.6)	6.2s	602 (94.4)	6.6s	638 (100.0)	6.0s	638 (100.0)	6.0s
Set	18	748	710 (94.9)	8.5s	682 (91.7)	9.4s	746 (99.7)	8.2s	746 (99.7)	8.4s
Map	12	322	298 (92.5)	4.8s	256 (79.5)	5.0s	320 (99.4)	4.7s	320 (99.4)	4.6s
WeakMap	5	198	168 (84.8)	2.3s	–	–	196 (99.0)	2.2s	196 (99.0)	2.3s
WeakSet	5	166	142 (85.5)	2.2s	–	–	164 (98.8)	2.1s	164 (98.8)	2.1s
Promise	13	1,264	1,154 (91.3)	9.8s	272 (55.5)	19.8s	1,256 (99.4)	9.6s	1,256 (99.4)	9.6s
Total	89	8,049	7,705 (95.7)	60.1s	6,167 (90.9)	89.3s	8,033 (99.8)	58.1s	8,033 (99.8)	58.4s

[†] es-shims evaluated on supported subset, and (–) indicates unsupported built-ins.

RQ1) Correctness

- Tested by **Test262** - official conformance tests
- (# of Fail) **core-js - 344** / **es-shims - 618** (of 6,785 supported) / **MiniPoly - 16**

Category	# Built-ins	Tests	core-js		es-shims [†]		MiniPoly		MiniPoly-opt	
			Pass (%)	Time	Pass (%)	Time	Pass (%)	Time	Pass (%)	Time
Array	24	4,713	4,617 (98.0)	26.4s	4,355 (94.9)	48.5s	4,713 (100.0)	25.4s	4,713 (100.0)	25.6s
String	12	638	616 (96.6)	6.2s	602 (94.4)	6.6s	638 (100.0)	6.0s	638 (100.0)	6.0s
Set	18	748	710 (94.9)	8.5s	682 (91.7)	9.4s	746 (99.7)	8.2s	746 (99.7)	8.4s
Map	12	322	298 (92.5)	4.8s	256 (79.5)	5.0s	320 (99.4)	4.7s	320 (99.4)	4.6s
WeakMap	5	198	168 (84.8)	2.3s	–	–	196 (99.0)	2.2s	196 (99.0)	2.3s
WeakSet	5	166	142 (85.5)	2.2s	–	–	164 (98.8)	2.1s	164 (98.8)	2.1s
Promise	13	1,264	1,154 (91.3)	9.8s	272 (55.5)	19.8s	1,256 (99.4)	9.6s	1,256 (99.4)	9.6s
Total	89	8,049	7,705 (95.7)	60.1s	6,167 (90.9)	89.3s	8,033 (99.8)	58.1s	8,033 (99.8)	58.4s

[†] es-shims evaluated on supported subset, and (–) indicates unsupported built-ins.

RQ1) Correctness

Inherent limitations
shared by all polyfills

- Tested by **Test262** - official conformance tests
- (# of Fail) **core-js** - 344 / **es-shims** - 618 (of 6,785 supported) / **MiniPoly** - 16

Category	# Built-ins	Tests	core-js		es-shims [†]		MiniPoly		MiniPoly-opt	
			Pass (%)	Time	Pass (%)	Time	Pass (%)	Time	Pass (%)	Time
Array	24	4,713	4,617 (98.0)	26.4s	4,355 (94.9)	48.5s	4,713 (100.0)	25.4s	4,713 (100.0)	25.6s
String	12	638	616 (96.6)	6.2s	602 (94.4)	6.6s	638 (100.0)	6.0s	638 (100.0)	6.0s
Set	18	748	710 (94.9)	8.5s	682 (91.7)	9.4s	746 (99.7)	8.2s	746 (99.7)	8.4s
Map	12	322	298 (92.5)	4.8s	256 (79.5)	5.0s	320 (99.4)	4.7s	320 (99.4)	4.6s
WeakMap	5	198	168 (84.8)	2.3s	–	–	196 (99.0)	2.2s	196 (99.0)	2.3s
WeakSet	5	166	142 (85.5)	2.2s	–	–	164 (98.8)	2.1s	164 (98.8)	2.1s
Promise	13	1,264	1,154 (91.3)	9.8s	272 (55.5)	19.8s	1,256 (99.4)	9.6s	1,256 (99.4)	9.6s
Total	89	8,049	7,705 (95.7)	60.1s	6,167 (90.9)	89.3s	8,033 (99.8)	58.1s	8,033 (99.8)	58.4s

[†] es-shims evaluated on supported subset, and (–) indicates unsupported built-ins.

RQ2) Reduced Manual Effort

Object	#	Baseline		MiniPoly		MiniPoly-opt		Rules
		core-js	es-shims	Auto	Man.	Auto	Man.	
Array	24	2,298	8,978	1,332	317			
String	12	1,281	6,885	418	130			
Set	18	2,313	5,880	864	190	861	283	114 (11)
WeakSet	5	1,882	–	387	123	379	214	
Map	12	1,986	3,062	756	217	747	327	81 (6)
WeakMap	5	2,034	–	446	115	436	231	
Promise	13	2,518	8,031	1,511	203			
Total[†]	89	14,312	32,836	5,714	1,295	5,684	1,705	195 (17)

RQ2) Reduced Manual Effort

Object	#	Baseline		MiniPoly		MiniPoly-opt		Rules
		core-js	es-shims	Auto	Man.	Auto	Man.	
Array	24	2,298	8,978	1,332	317			
String	12	1,281	6,885	418	130			
Set	18	2,313	5,880	864	190	861	283	114 (11)
WeakSet	5	1,882	–	387	123	379	214	
Map	12	1,986	3,062	756	217	747	327	81 (6)
WeakMap	5	2,034	–	446	115	436	231	
Promise	13	2,518	8,031	1,511	203			
Total [†]	89	14,312	32,836	5,714	1,295	5,684	1,705	195 (17)

9.05%

RQ2) Reduced Manual Effort

Object	#	Baseline		MiniPoly		MiniPoly-opt		Rules
		core-js	es-shims	Auto	Man.	Auto	Man.	
Array	24	2,298	8,978	1,332	317			
String	12	1,281	6,885	418	130			
Set	18	2,313	5,880	864	190	861	283	114 (11)
WeakSet	5	1,882	–	387	123	379	214	
Map	12	1,986	3,062	756	217	747	327	81 (6)
WeakMap	5	2,034		446	115	436	231	
Promise	13	2,518	8,031	1,511	203			
Total [†]	89	14,312	32,836	5,714	1,295	5,684	1,705	195 (17)

3.94%

9.05%

RQ2) Reduced Manual Effort

Object	#	Baseline		MiniPoly		MiniPoly-opt		Rules
		core-js	es-shims	Auto	Man.	Auto	Man.	
Array	24	2,298	8,978	1,332	317			
String	12	1,281	6,885	418	130			
Set	18	2,313	5,880	864	190	861	283	114 (11)
WeakSet	5	1,882	–	387	123	379	214	
Map	12	1,986	3,062	756	217	747	327	81 (6)
WeakMap	5	2,034	–	446	115	436	231	
Promise	13	2,518	8,031	1,511	203			
Total[†]	89	14,312	32,836	5,714	1,295	5,684	1,705	195 (17)

17 Manual Rules

RQ2) Reduced Manual Effort

Object	#	Baseline		MiniPoly		MiniPoly-opt		Rules
		core-js	es-shims	Auto	Man.	Auto	Man.	
Array	24	2,298	8,978	1,332	317			
String	12	1,281	6,885	418	130			
Set	18	2,313	5,880	864	190	861	283	114 (11)
WeakSet	5	1,882	–	387	123	379	214	
Map	12	1,986	3,062	756	217	747	327	81 (6)
WeakMap	5	2,034	–	446	115	436	231	
Promise	13	2,518	8,031	1,511	203			
Total [†]	89	14,312	32,836	5,714	1,295	5,684	1,705	195 (17)

17 Manual Rules

13.28%

RQ2) Reduced Manual Effort

Object	#	Baseline		MiniPoly		MiniPoly-opt		Rules
		core-js	es-shims	Auto	Man.	Auto	Man.	
Array	24	2,298	8,978	1,332	317			
String	12	1,281	6,885	418	130			
Set	18	2,313	5,880	864	190	861	283	114 (11)
WeakSet	5	1,882	–	387	123	379	214	
Map	12	1,986	3,062	756	217	747	327	81 (6)
WeakMap	5	2,034	–	446	115	436	231	
Promise	13	2,518	8,031	1,511	205			
Total [†]	89	14,312	32,836	5,714	1,295	5,684	1,705	195 (17)

5.79%

13.28%

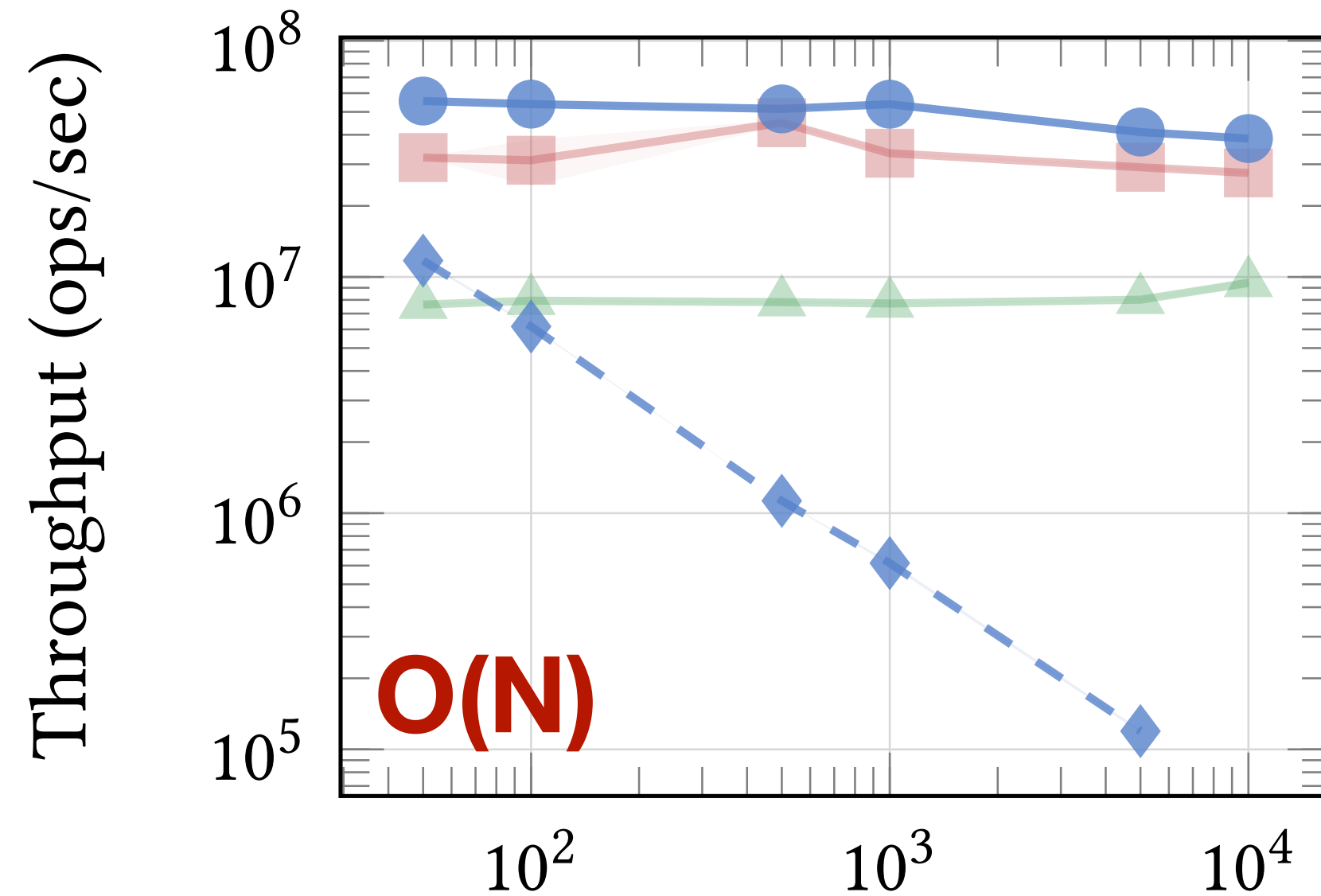
17 Manual Rules

RQ3) Speed

- We measured **throughput** (*y-axis* = ops/sec) across **sizes** (*x-axis* = N)

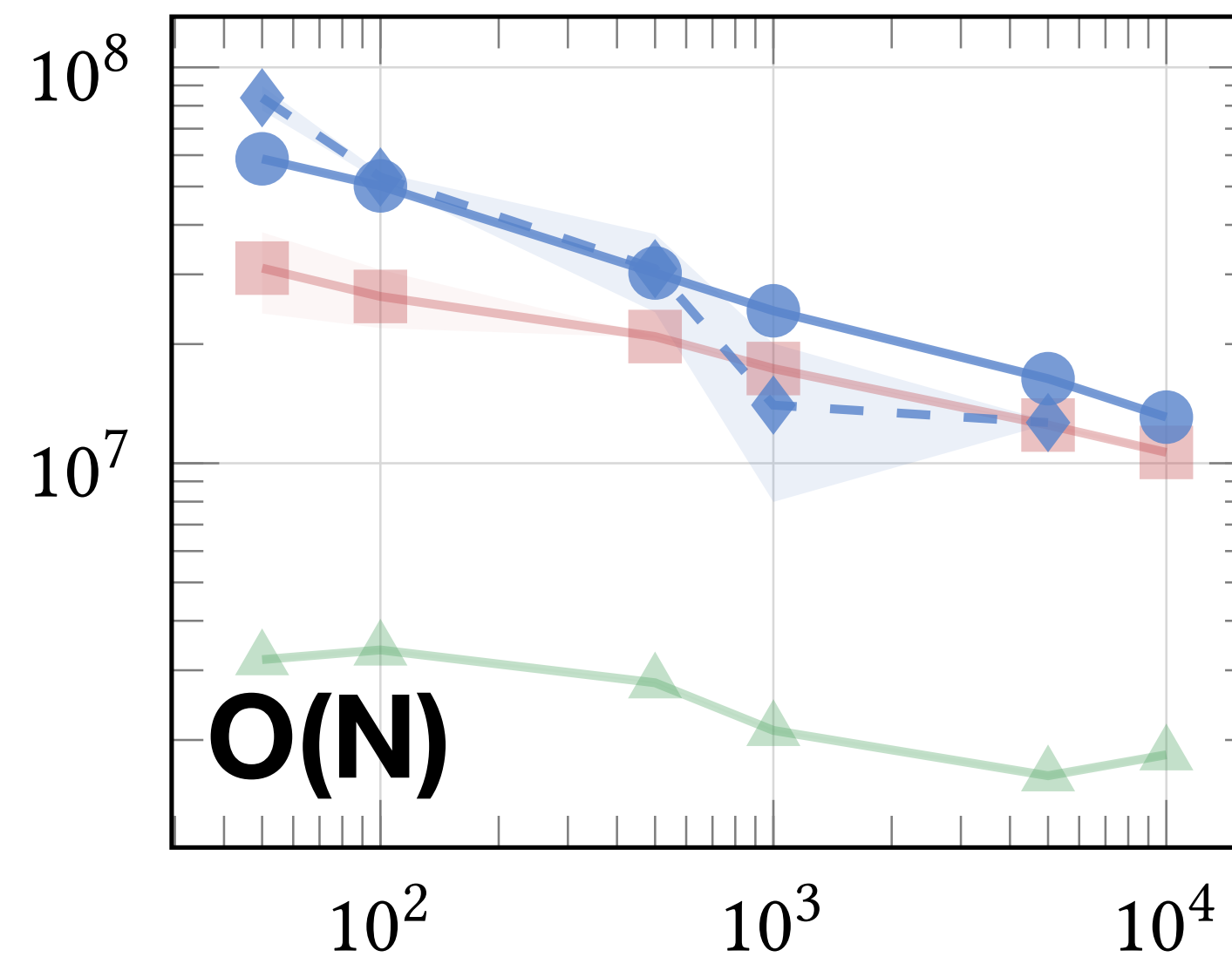
add, delete, has, size

Set — Simple



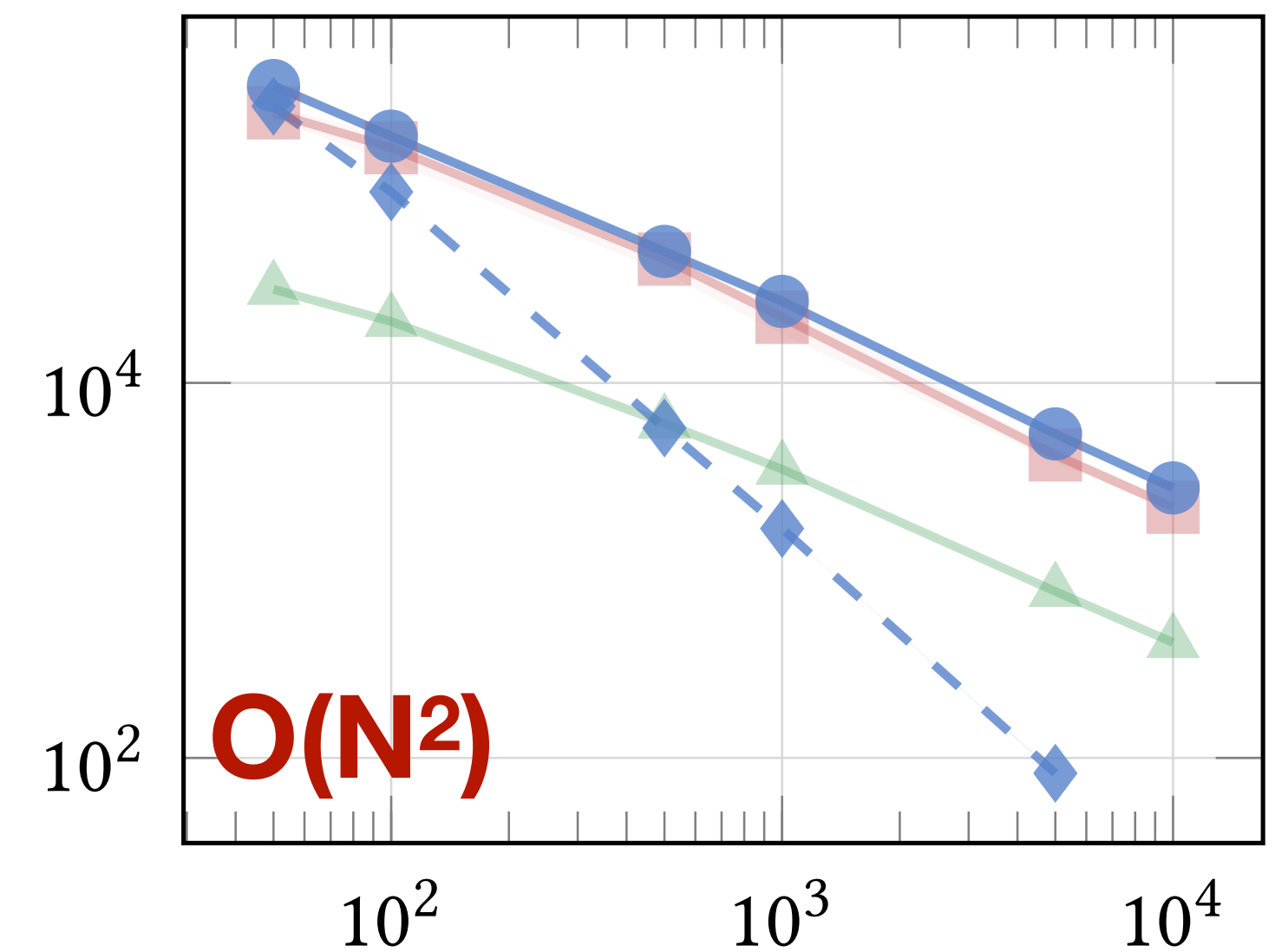
entries, forEach, ...

Set — Iterate



isSupersetOf, union, ...

Set — Composite



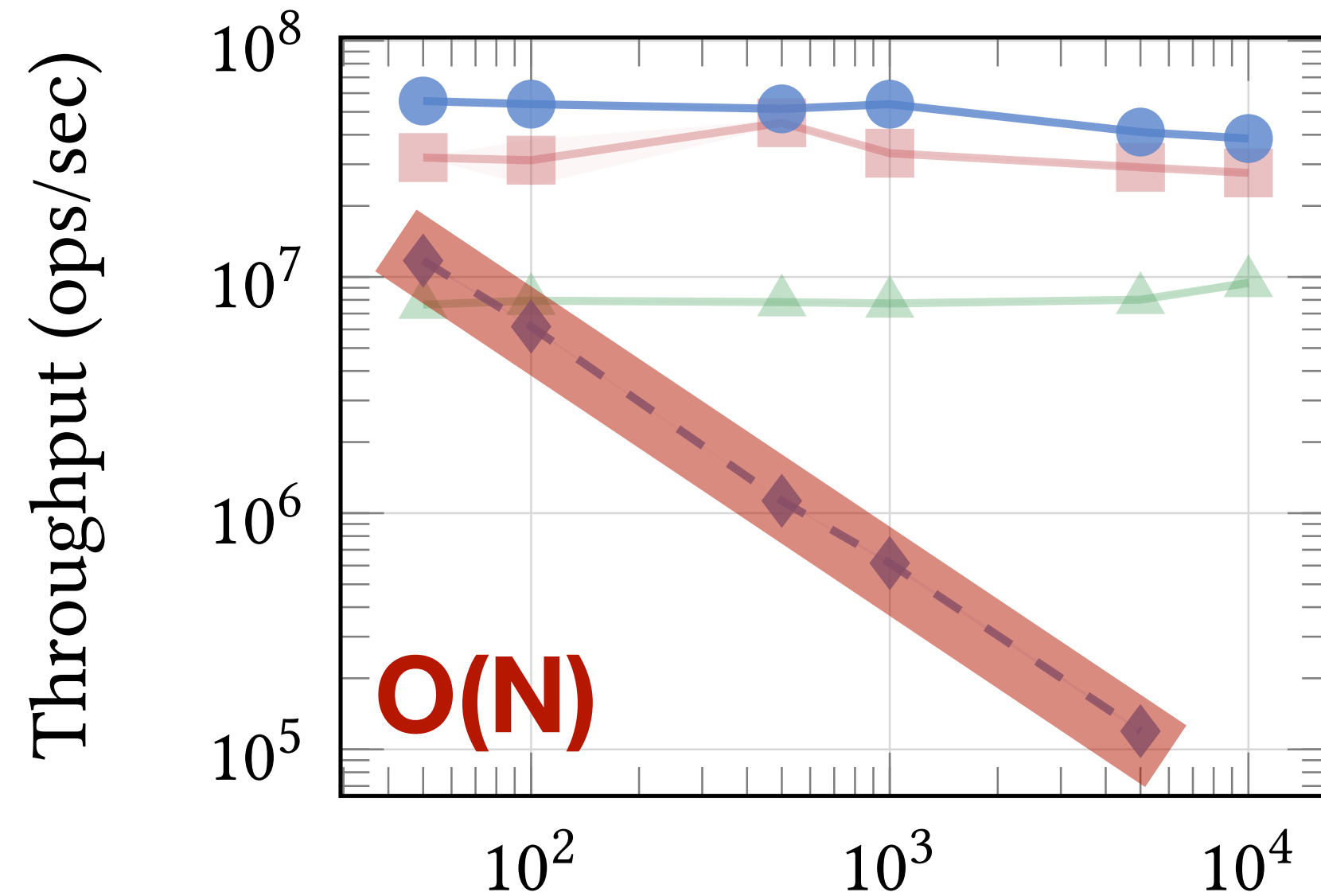
core-js es-shims MiniPoly MiniPoly-opt

RQ3) Speed

- We measured **throughput** (*y-axis* = ops/sec) across **sizes** (*x-axis* = N)

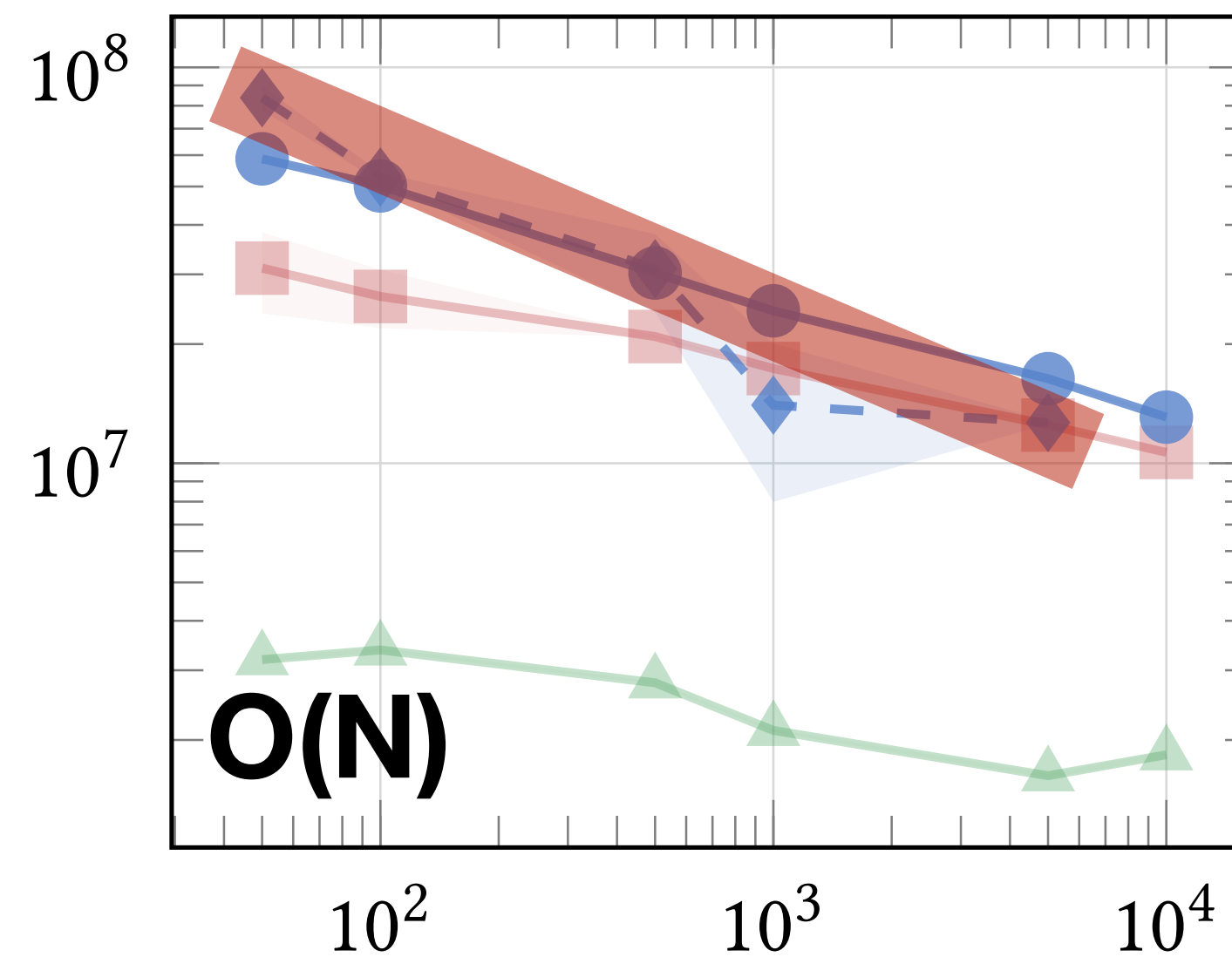
add, delete, has, size

Set — Simple



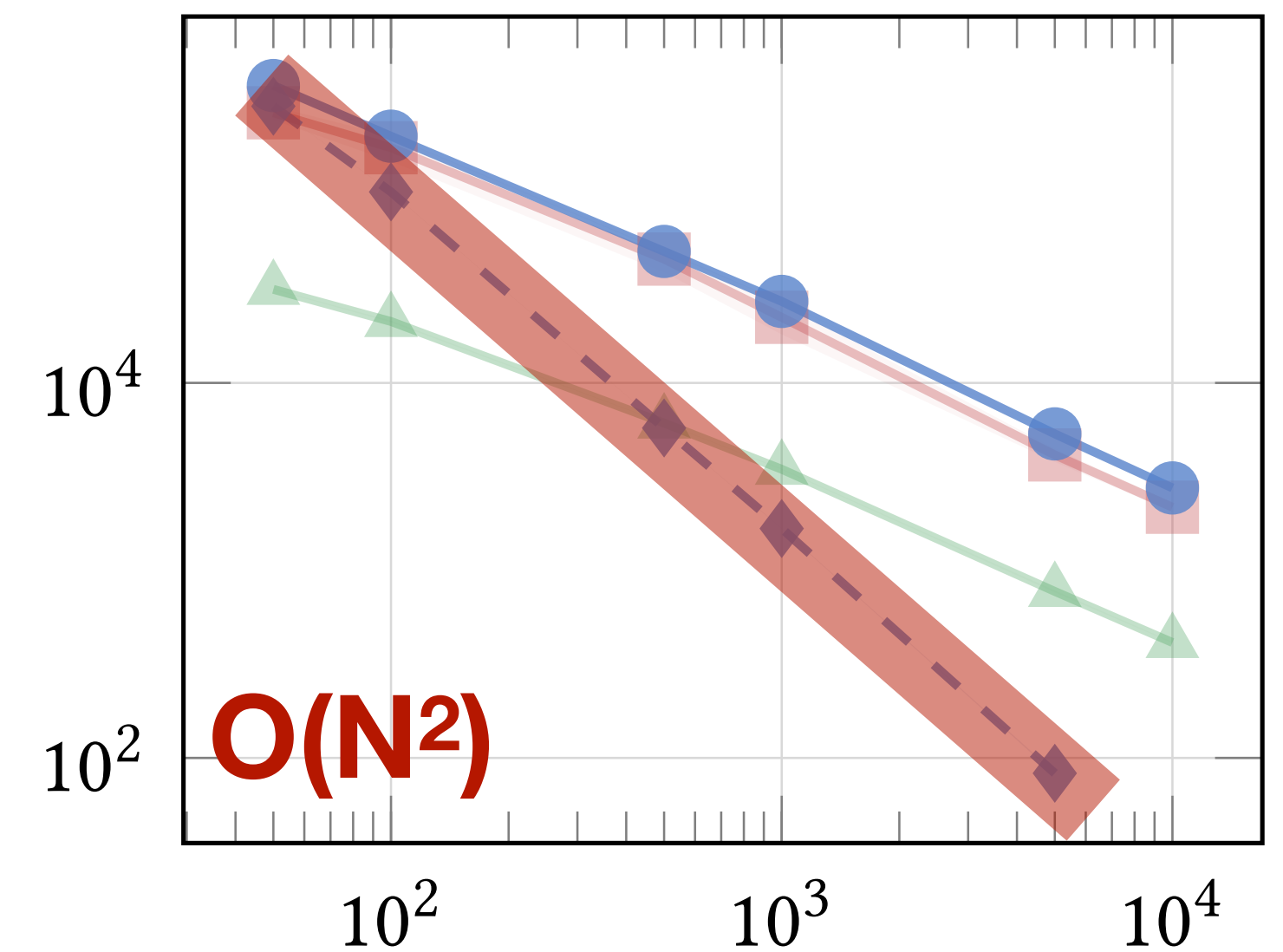
entries, forEach, ...

Set — Iterate



isSupersetOf, union, ...

Set — Composite



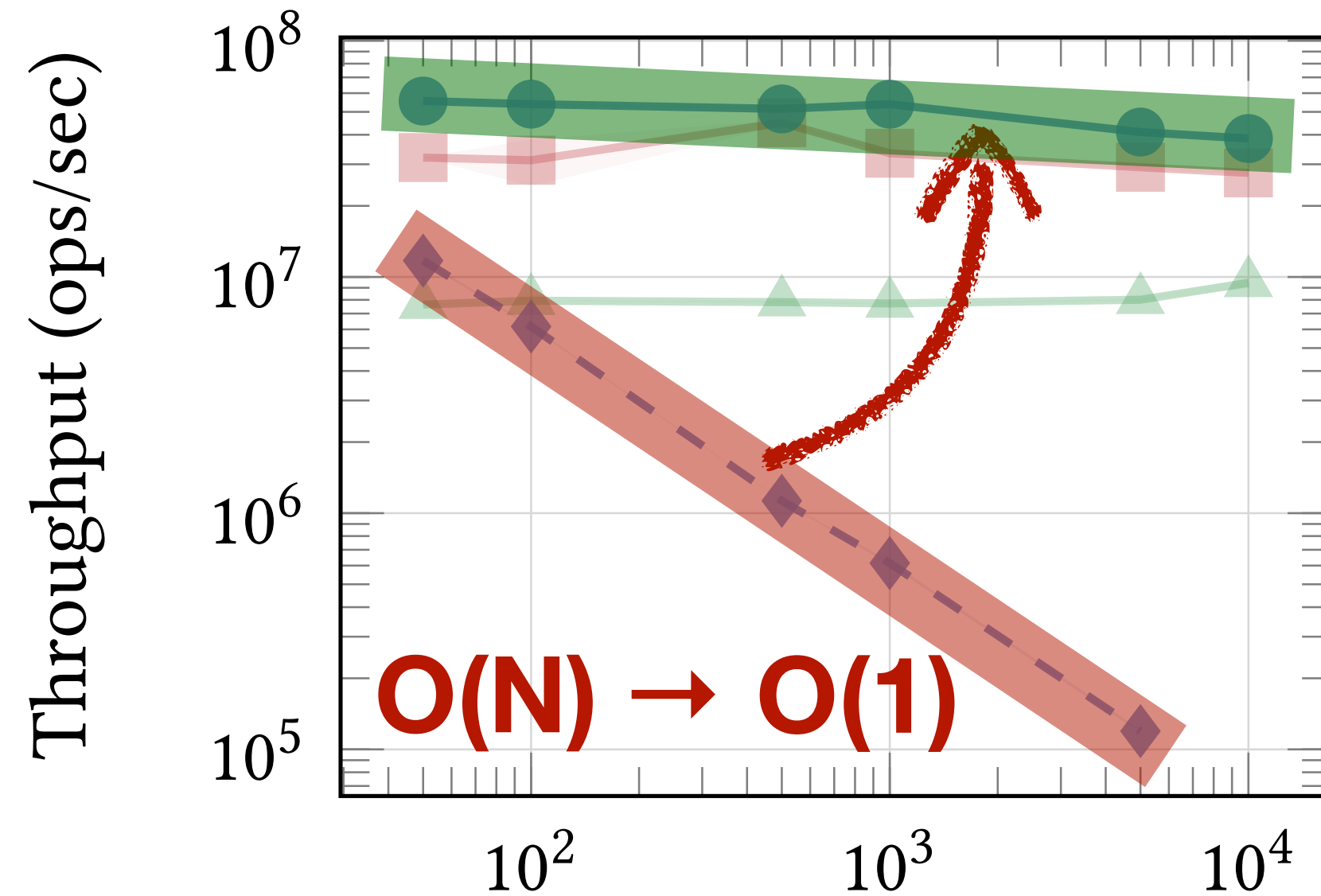
core-js es-shims MiniPoly MiniPoly-opt

RQ3) Speed

- We measured **throughput** (*y-axis* = ops/sec) across **sizes** (*x-axis* = N)

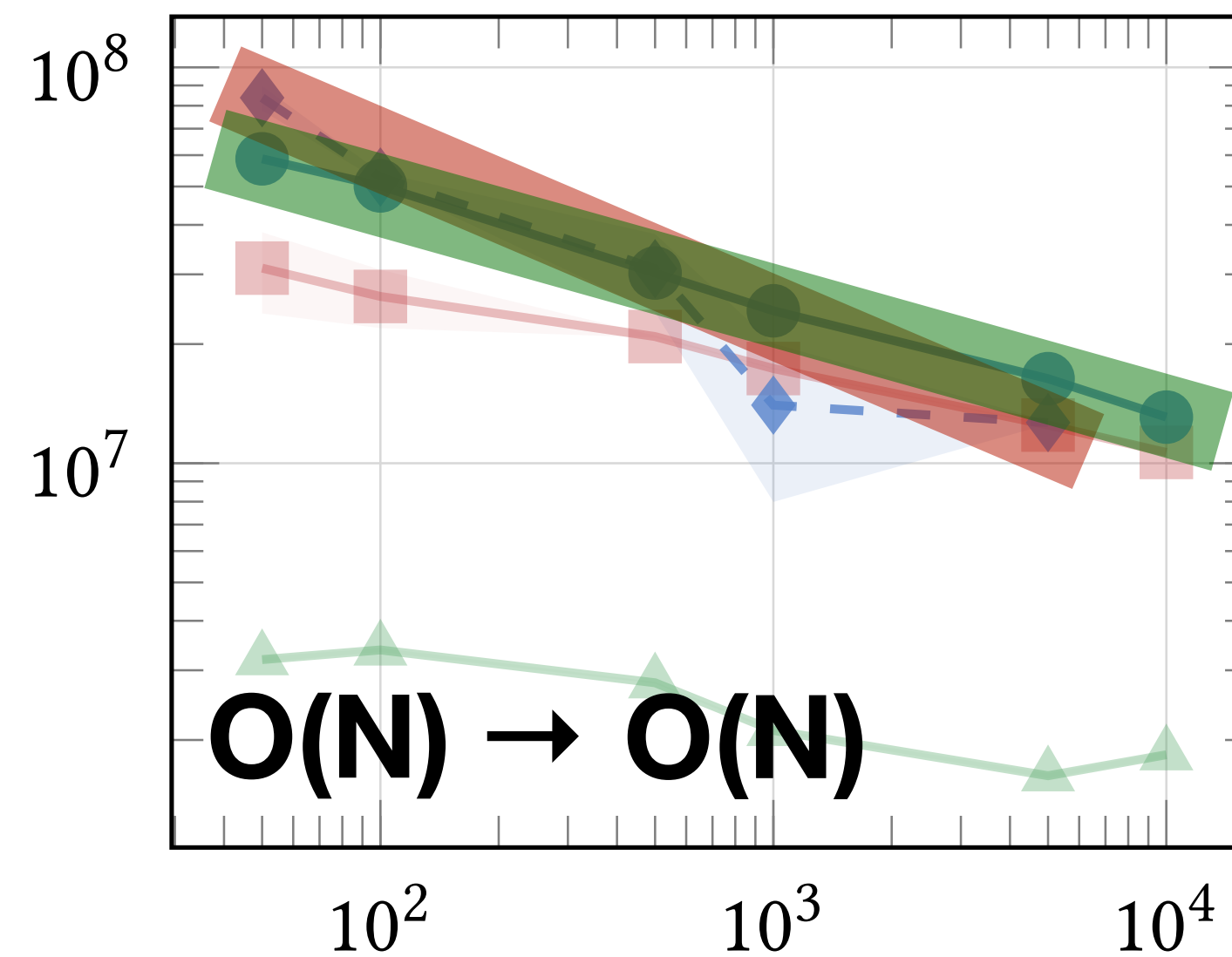
add, delete, has, size

Set — Simple



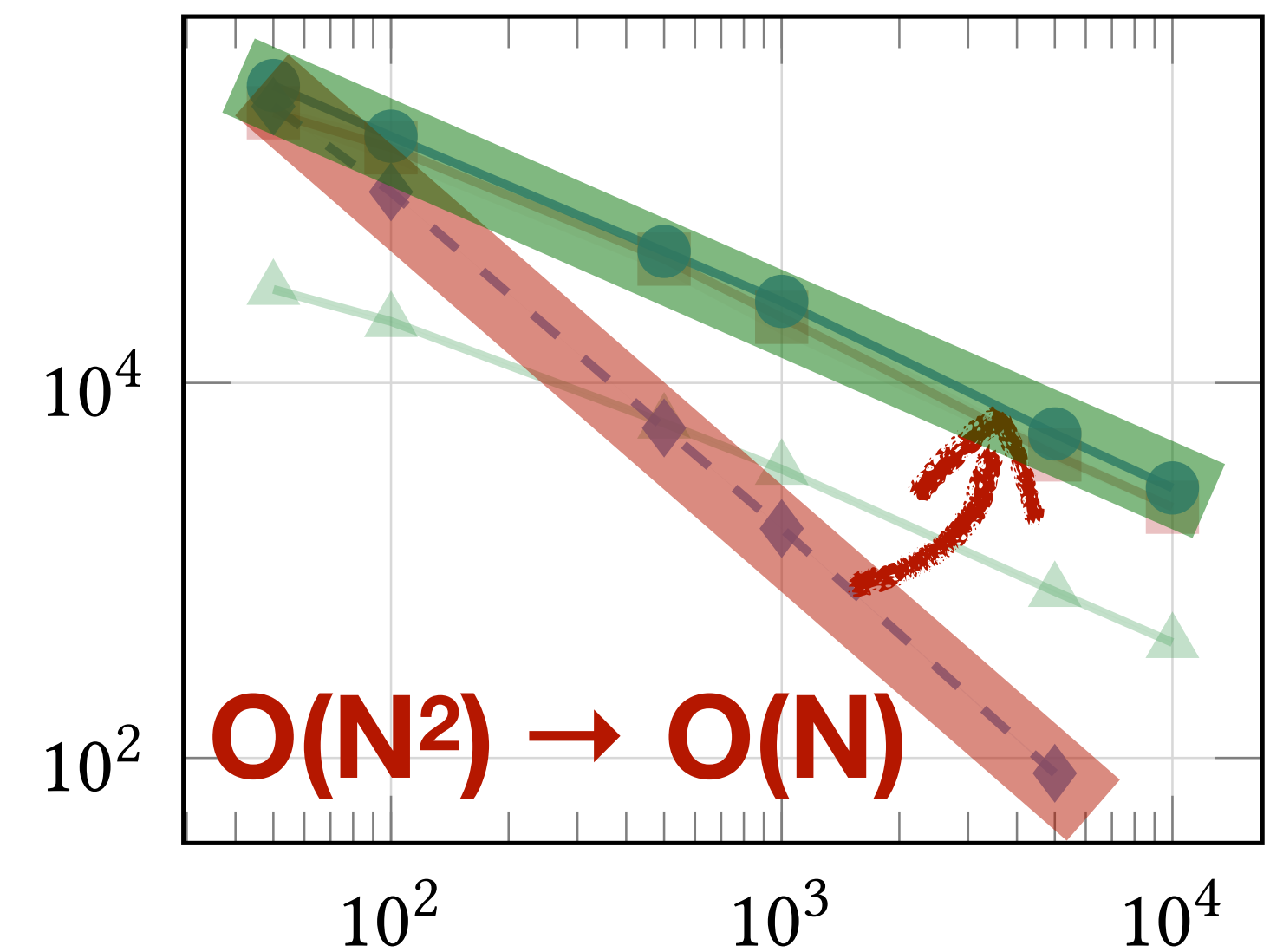
entries, forEach, ...

Set — Iterate



isSupersetOf, union, ...

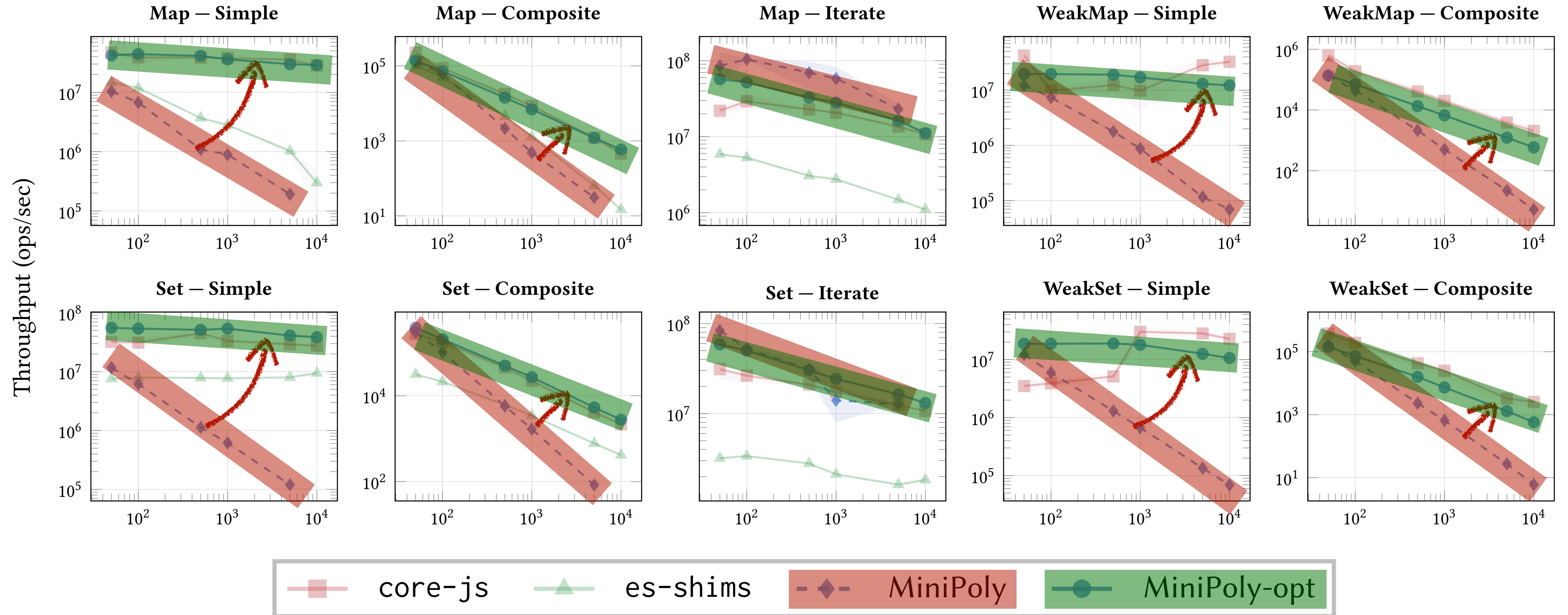
Set — Composite



core-js es-shims MiniPoly MiniPoly-opt

RQ3) Speed

Competitive with baselines in most cases

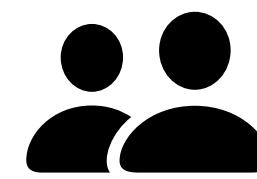


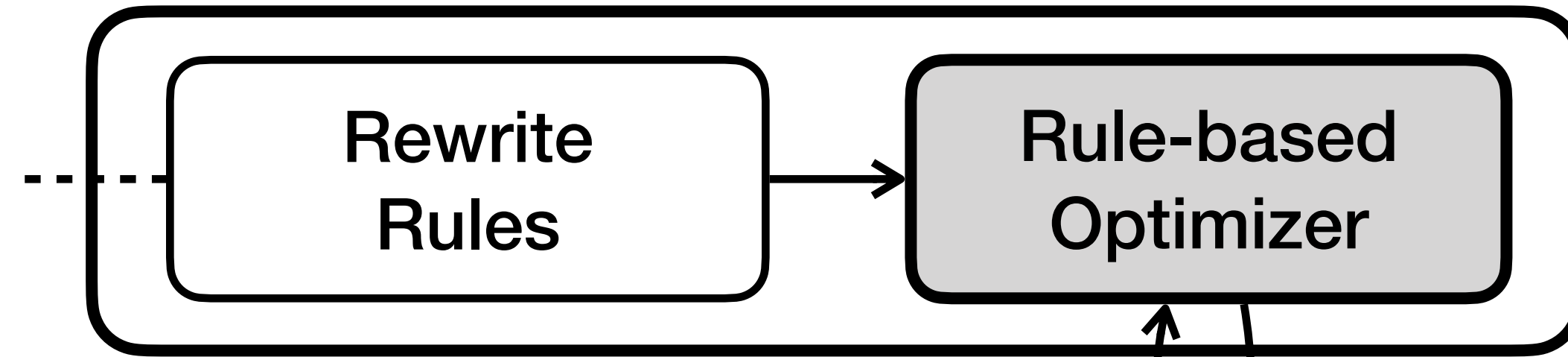
RQ4) Forward Compatibility

- Supports Built-in APIs in **ES2026** and **Stage 2.7**
with **100% correctness** and **comparable speed** with manual baselines

Proposal	Method	Tests	Correctness (Pass %)				Throughput (ops/sec) — N = 5,000			
			core-js	es-shims	MiniPoly	MiniPoly-opt	core-js	es-shims	MiniPoly	MiniPoly-opt
Upsert (ES2026)	Map.prototype									
	.getOrInsert	28	26 (92.9)	24 (85.7)	28 (100.0)	28 (100.0)	13.7M / 2.0M	61.2K	107K	8.2M / 2.5M
	.getOrInsertComputed	37	35 (94.6)	33 (89.2)	37 (100.0)	37 (100.0)	11.7M / 1.8M	40.8K	77.7K	6.5M / 1.9M
	WeakMap.prototype									
	.getOrInsert	34	26 (76.5)	—	34 (100.0)	34 (100.0)	18.4M / 3.1M	—	104K	8.2M / 2.3M
	.getOrInsertComputed	43	32 (74.4)	—	43 (100.0)	43 (100.0)	5.7M / 1.2M	—	73.1K	5.0M / 1.8M
Await Dictionary (Stage 2.7)	Promise									
	.allKeyed	12	—	—	12 (100.0)	12 (100.0)	—	—	—	—
	.allSettledKeyed	12	-23	-8	12 (100.0)	12 (100.0)	—	—	—	—
Total		166	119 (83.8)	57 (87.7)	166 (100.0)	166 (100.0)	11.3M / 1.9M	50.0K	89.3K	6.8M / 2.1M

Challenge 2: Efficiency

 **Polyfill Developer**



Solution 2 - Rewrite Rules

```

pattern: |
  1. For each element {{ elem : var }} of {{ base : ref }}, do
    1. If {{ elem }} is not ~empty~ and SameValue({{ elem }}, {{ value : var }}) is *true*, then
      1. {{ body : step }}
where:
  - isSetData(base)
  - { isSetData: " _.[[SetData]] " }
replace: |
  1. If SET_DATA HAS {{ base }}, {{ value }} is *true*, then
    1. {{ body }}
    (* simplified)
  
```

TC 39

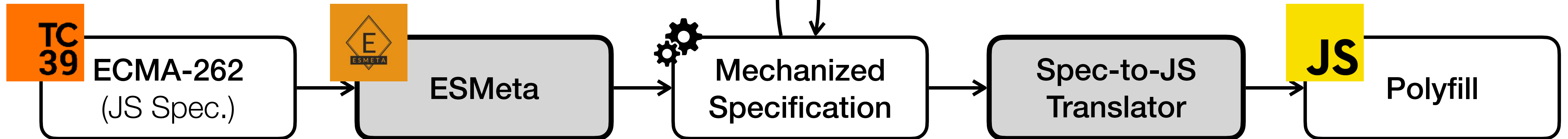
```

Set.prototype.add ( value )
...
4. For each element e of S.[[SetData]], do
  a. If e is not EMPTY and SameValue(e, value) is true, then
    ...
  
```

JS

function add(value) {
 ...
 if (SET_DATA HAS S.SetData, value)
 ...
}

PLRG MiniPoly: Automatic Extraction of Efficient JavaScript Polyfill from Language Specification 23/32



Solution 1 - Completion Type Analysis

```

function IteratorClose(iteratorRecord, completion, completion_flag) {
  ...
  try { var innerResult = GetMethod(...); var innerResult_flag = true; }
  catch (e) { var innerResult = e; var innerResult_flag = false; }
  if (innerResult_flag) {
    ...
    if (_return === undefined) {
      if (completion_flag) return completion;
      else throw completion;
    }
    try { innerResult = Call(...); innerResult_flag = true; } catch ...
  }
  if (!completion_flag) throw completion;
  if (!innerResult_flag) throw innerResult;
  ...
  return completion;
}
  
```

JS

PLRG MiniPoly: Automatic Extraction of Efficient JavaScript Polyfill from Language Specification 19/32

Challenge 1: Correctness