Stat. Extractor → Statistics → Mutator

Sampler → {;} JS → Injector → {;} JS → JS Engines → Localizer → Spec. Bugs

**Bug Detection & Localization**

**Conformance Test Generation**

**Conformance Test Generation**

**Bug Detection & Localization**

**Conformance Test Generation**

**Bug Detection & Localization**

**Conformance Test Generation**

Language Spec.

Seed Synthesizer

Seed

Target Program

Program Mutator

Mutated Program

Target Selector

Programs

Assertion Injector

Tests

**Bug Detection & Localization**

Engine-1

...

Engine-N

Bug Localizer

Spec. Bugs

Engine Bugs

### 7.2.15 Abstract Equality Comparison

1. If $Type(x)$ is the same as $Type(y)$, then
   a. Return the result of performing Strict Equality Comparison $x === y$.
2. If $x$ is **null** and $y$ is **undefined**, return **true**.
3. If $x$ is **undefined** and $y$ is **null**, return **true**.

$\cdots$

10. If $Type(x)$ is either String, Number, BigInt, or Symbol and $Type(y)$ is Object, return the result of the comparison $x ==$ ToPrimitive($y$).
11. If $Type(x)$ is Object and $Type(y)$ is either String, Number, BigInt, or Symbol, return the result of the comparison ToPrimitive($x$) $== y$.

$\cdots$

*ArrayLiteral*[Yield, Await] :

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

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

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

*MemberExpression* **:**

    *PrimaryExpression*

    *MemberExpression* **[** *Expression* **]**

    *MemberExpression* **.** *IdentifierName*

    **new** *MemberExpression* *Arguments*

| **new** | *MemberExpression* | *Arguments* |
|---|---|---|
| **new** | x | **()** |
| **new** | x | **(x)** |
| | (the shortest string of *MemberExpression*) | **(...x)** |
| | | **(x,)** |

➡
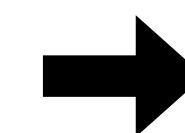
```
new x()
new x(x)
new x(...x)
new x(x,)
```

**Array.prototype.indexOf (** *searchElement* **[ ,** *fromIndex* **] )**