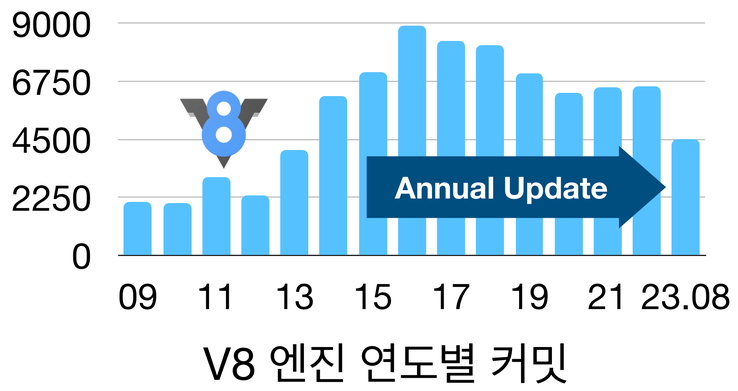
Black-box 구현체로부터 ECMAScript 명세 합성하기

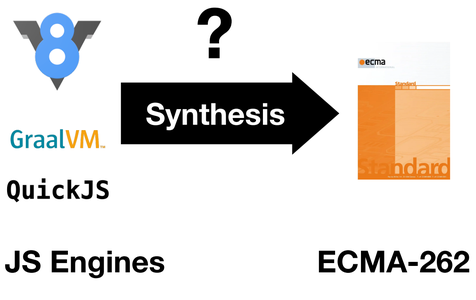
김현준 최민석 박지혁 I 고려대학교 프로그래밍 언어 연구실

1. Problem & Goal

Problem : 구현 우선의 빠르게 성장하는 JavaScript



V8 엔진 연도별 커밋

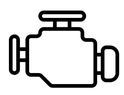


JS Engines

ECMA-262

Goal : 구현을 이용한 명세의 자동 합성

`Array.prototype.shift`



JS Engine

Synthesis

...
2. Let *len* be ? LengthOfArrayLike(*O*).
3. If *len* = 0, then
    a. Perform ? Set(*O*, **"length"**, +0$_{\mathbb{F}}$, **true**).
    b. Return **undefined**.
4. Let *first* be ? Get(*O*, **"0"**).
5. Let *k* be 1.
...

2. Insight

명세는 반복되는 패턴이 많음
기존 명세를 읽어 합성 부품들을 추출

...
2. Let *len* be ? LengthOfArrayLike(*O*).
3. If *len* = 0, then
    a. Perform ? Set(*O*, **"length"**, +0$_{\mathbb{F}}$, **true**).
    b. Return **undefined**.
4. Let *first* be ? Get(*O*, **"0"**).
5. Let *k* be 1.
...

`HasProperty(■,■)`
`ToObject(■)` `ToString(■)`
`LengthOfArrayLike(■)`
`Get(■,■)` `Set(■,■,■,■)`

Helper Function Brick

```
Let ■ be ■.
Repeat, while ■,
■
Set ■ to ■.
```

Structural Brick

부품들의 특징을 분석
합성에 활용

`ToObject(■)`

```
[MetaData]
- null, undefined에 대해
TypeError
```

`LengthOfArrayLike(■)`

```
[MetaData]
- “length”라는 property를
[[GET]]
- BigInt, Symbol에 대해
TypeError
```

```
Let ■ be ■.
Repeat, while ■,
■
Set ■ to ■.
```

```
[MetaData]
- 앞서
LengthOfArrayLike()를 호
출한 후 등장할 가능성 높음
```

`Array.prototype.shift`

Test using `null`

```
TypeError 발생:
MetaData를 바탕으로 ToObject, ... 추정
```

`ToObject(0)`

```
추정된 Brick들의
MetaData를 바탕으로 테스트를 생성하고 결정
```

Test using `{}`

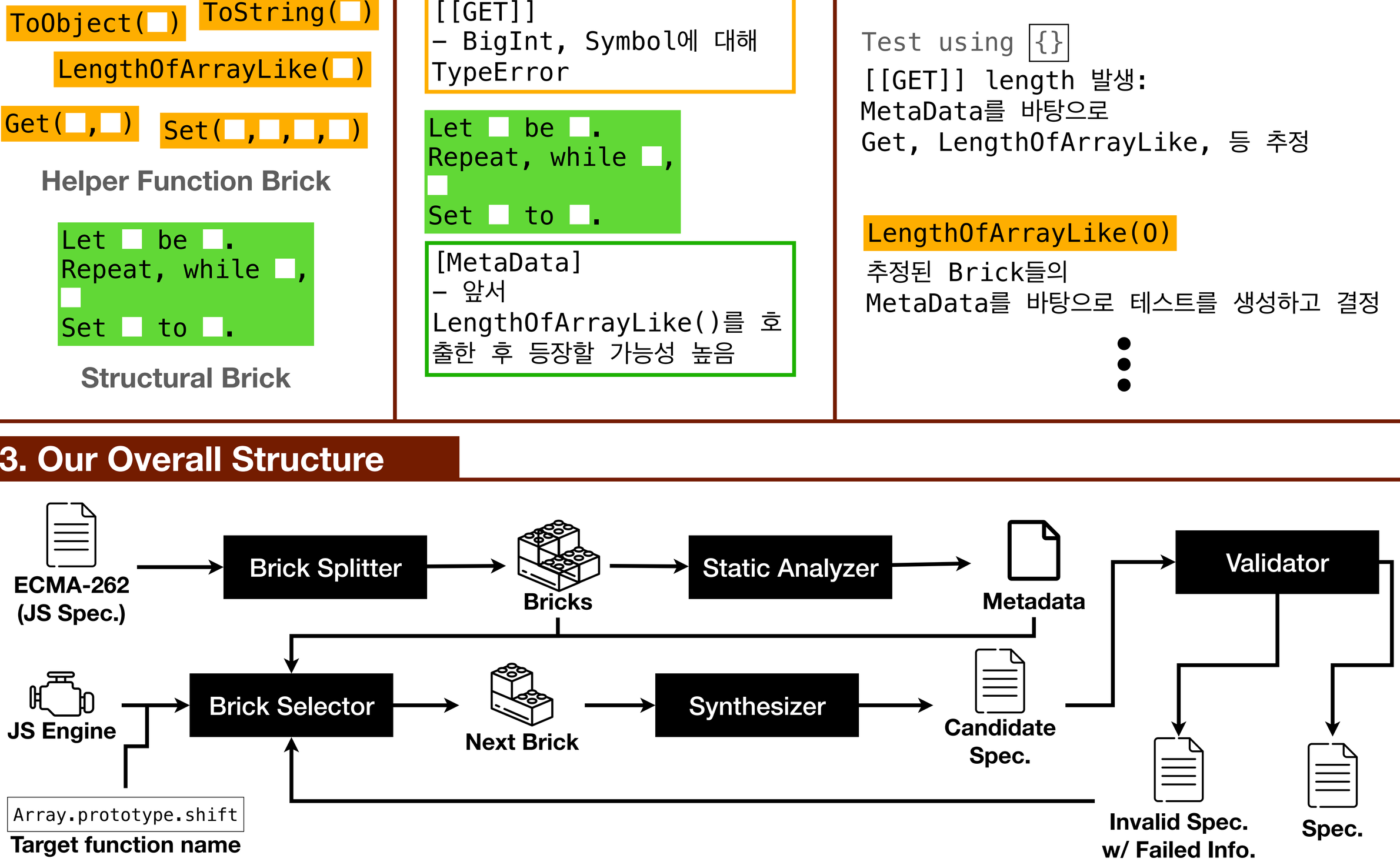
```
[[GET]] length 발생:
MetaData를 바탕으로
Get, LengthOfArrayLike, 등 추정
```

`LengthOfArrayLike(0)`

```
추정된 Brick들의
MetaData를 바탕으로 테스트를 생성하고 결정
```

⋮

3. Our Overall Structure



4. Related Works

[FSE’15] S. Heule, et al. "Mimic: Computing Models for Opaque Code"

[NDSS’19] Han, et al. “CodeAlchemist: Semantics-Aware Code Generation to Find Vulnerabilities in JavaScript Engines”