

자동으로 타입 가드 생성하기: JavaScript 명세의 더 나은 타입 분석을 위해

김준겸¹ 최용욱¹ 백도현² 박지혁¹ 류석영²

고려대학교 프로그래밍언어연구실¹
KAIST PLRG²

Background

- 타입 가드란?
 - 조건문을 이용한 타입 좁히기의 일종
- 사용자가 직접 타입 가드를 만들 수도 있음
 - Python: User-defined type guard
 - TypeScript: Type predicates

타입 가드를 이용하는 예제

```
1 let obj = ???;
2 // isArray라는 타입 가드가 있다고 가정
3 let flag = isArray(obj);
4 // 타입 가드의 반환값을 조건문에 사용
6 if (flag) {
7   obj // 배열(any[])로 추론하기를 기대
9 } else {
10  obj // any
11 }
```

- JavaScript 명세란?
 - ECMA-262 라는 문서로 관리됨
 - 언어의 동작이 절차적 언어처럼 기술됨
 - 2014~: GitHub으로 관리, 2600+ commits
 - 명세를 기반으로 구현체를 만들기 때문에 작성된 명세의 올바름이 보장되는 것이 중요
- 이전 연구 (JSTAR)
 - 명세의 타입 분석을 통해 잘못 작성된 부분 찾기
 - 값 비교와 같은 가벼운 조건문 기반 타입 좁히기

Motivation

- IsCallable의 반환값으로 얻을 수 있는 정보:
 - True: *adder*는 FunctionObject이다.
 - False: *adder*는 FunctionObject가 아니다.
- 정보를 사용하지 않으면 (현재):
 - AddEntriesFromIterable 호출 시 인자 타입에 대한 불일치 발생
- 정보를 사용한다면 (우리 연구):
 - *adder*는 FunctionObject이기 때문에 인자에 대한 **False alarm**이 없어짐

Key Idea

- 함수 분석 시 인자 타입이 좁혀졌다는 정보를 저장하여, 이를 반환 정보에 추가한다.

이전 IsCallable: Boolean을 반환

```
IsCallable: {
  True →
    { argument: FunctionObject }
  False →
    { argument: ESValue \ FunctionObject }
```

JavaScript Source Code

```
1 let map = new Map([["KU", "PLRG"]])
```

Semantics를 기술

JavaScript Specification

Map ([*iterable*])

...

1. Let *map* be ? OrdinaryCreateFromConstructor(...).

...

2. Let *adder* be ? Get(*map*, "set"). *adder*: ESValue

3. If IsCallable(*adder*) is false, throw a TypeError exception.

4. Return ? AddEntriesFromIterable(*map*, *iterable*, *adder*).

adder: FunctionObject를 기대

- Get (*O*, *P*): ECMAScript language value을 반환
- AddEntriesFromIterable (*target*, *iterable*, *adder*)
 - 인자의 *adder*: FunctionObject를 받음

IsCallable (*argument*): Returns Boolean

- True가 반환되는 경우 인자는 FunctionObject
- *argument*는 ECMAScript language value를 받음

1. If *argument* is not an Object, return false.

argument: ESValue \ Object

2. If *argument* has a [[Call]] internal method, return true.

argument: FunctionObject

3. Return false.

argument: Object \ FunctionObject