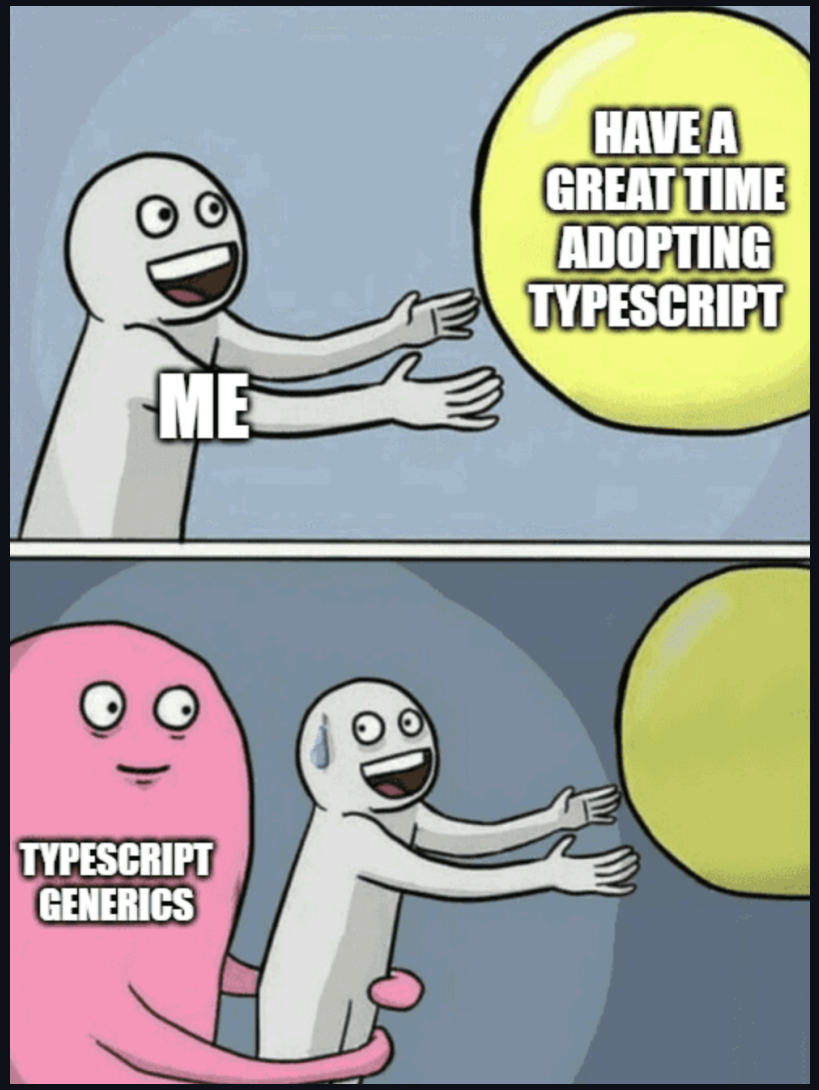
타입 레벨 프로그래밍


```
interface TS extends JS {
  typeSafety: true;
}
```

타입스크립트

- 2012년에 등장
- 자바스크립트의 슈퍼셋
- 정적 타입 언어

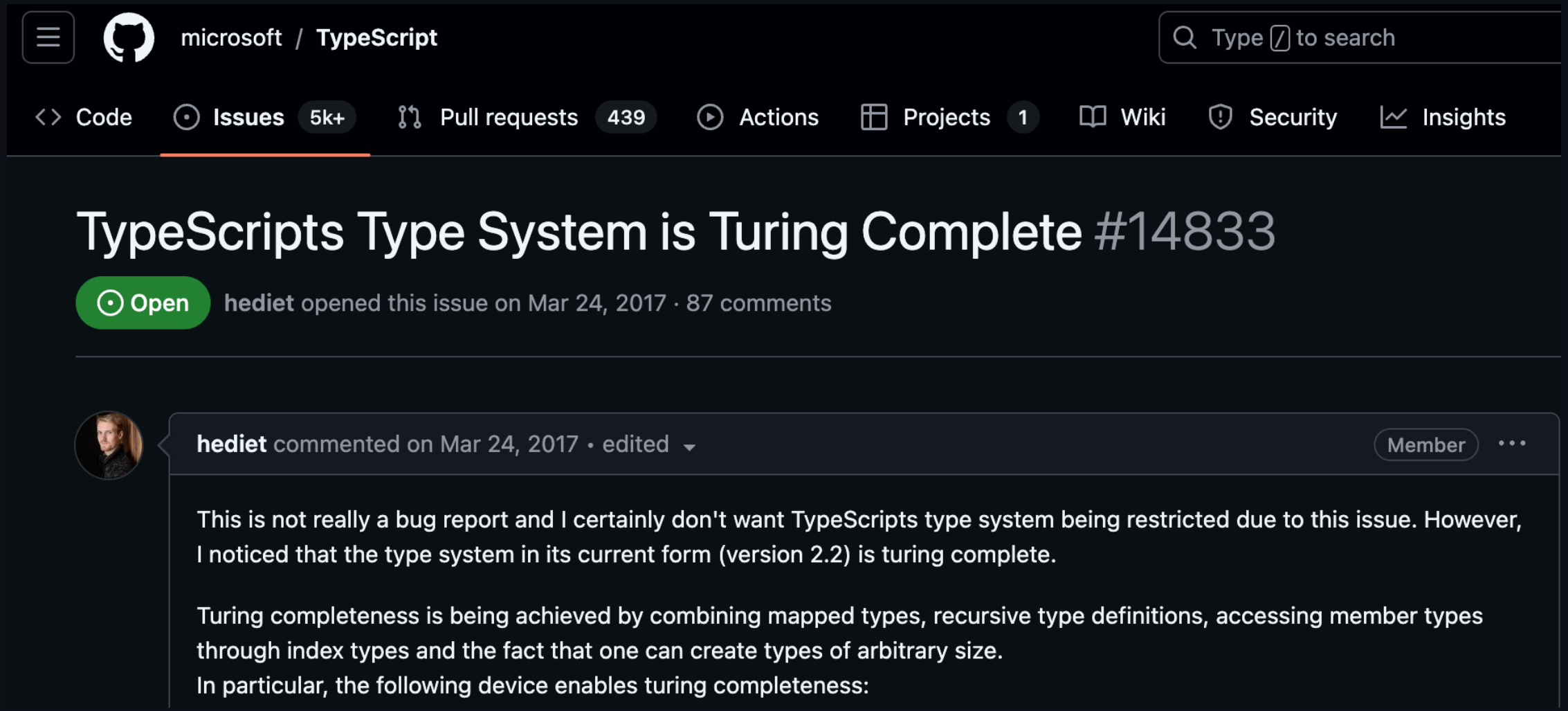

HAVE A GREAT TIME ADOPTING TYPESCRIPT
ME
TYPESCRIPT GENERICS


TYPE SAFETY
WITH TYPESCRIPT
AS ANY

타입시스템은 완전 튜링하다.

- **튜링 머신** 이란?
  수학자 앨런 튜링이 1936년에 제시한 개념으로 계산하는 기계의 일반적인 개념을 설명하기 위한 가상의 기계이며 오토마타의 일종이다

증명


microsoft / TypeScript
Type / to search
Code
Issues 5k+
Pull requests 439
Actions
Projects 1
Wiki
Security
Insights
TypeScripts Type System is Turing Complete #14833
Open
hediet opened this issue on Mar 24, 2017 · 87 comments
hediet commented on Mar 24, 2017 • edited
Member
This is not really a bug report and I certainly don't want TypeScripts type system being restricted due to this issue. However, I noticed that the type system in its current form (version 2.2) is turing complete.
Turing completeness is being achieved by combining mapped types, recursive type definitions, accessing member types through index types and the fact that one can create types of arbitrary size.
In particular, the following device enables turing completeness:

튜링 완전 언어 특징

- 계산 능력
- `조건문`과 `반복문`
- 기억과 상태 (변수 할당)
- ...

타입 레벨 프로그래밍

제네릭 (Generics)

- 제네릭은 선언 시점이 아니라 생성 시점에 타입을 명시하여 하나의 타입만이 아닌 다양한 타입을 사용할 수 있도록 하는 기법

```
type A<T> = T

type B = A<number> // number

type C = A<string> // string
```

개인적으론 타입레벨의 함수라고 생각함.

```
type Foo<
  A /* arg 1 */,
  B /* arg 2 */,
  C /* arg 3 */,
> = A & B & C /* return Type */
```

조건문

`extends` 의 역할은 두가지가 있는데 그중하나가 if의 역할이다.

```
type User {
  name: string;
  email: string;
}

type IsHasField<T, FieldName extends string> = FieldName extends keyof T
  ? true
  : false;

type HasEmailField<T> = IsHasField<T, 'email'>

type UseHasEmailField = HasEmailField<User> // true
```

반복문

- 반복문은 존재하지않지만 타입 재귀는 가능하다!
- 여기서 중요한 키워드는 `infer`

```
type IsAllNumber<T extends unknown[]> = T extends [infer First, ...infer Rest]
  ? First extends number
    ? IsAllNumber<Rest>
    : false
  : true;

type A = IsAllNumber<[1, 2, 3]>; // true
type B = IsAllNumber<[1, 2, 'knowre']>; // false
```

Currying의 타입 구현하기

에디터에서 진행

타입스크립트 챌린지

https://github.com/type-challenges/type-challenges

```
Type<Challenge[]>
```


hard 53
6·Simple Vue
17·Currying 1
55·Union to Intersection
57·Get Required
59·Get Optional
89·Required Keys
90·Optional Keys
112·Capitalize Words
114·CamelCase
147·C-printf Parser
213·Vue Basic Props
223·IsAny
270·Typed Get
300·String to Number
399·Tuple Filter
472·Tuple to Enum Object
545·printf
553·Deep object to unique
651·Length of String 2
730·Union to Tuple
847·String Join
956·DeepPick
1290·Pinia
1383·Camelize
2059·Drop String
2822·Split
2828·ClassPublicKeys
2857·IsRequiredKey
2949·ObjectFromEntries
4037·IsPalindrome
5181·Mutable Keys
5423·Intersection
6141·Binary to Decimal
7258·Object Key Paths
8804·Two Sum
9155·ValidDate
9160·Assign
9384·Maximum
9775·Capitalize Nest Object Keys
13580·Replace Union
14080·FizzBuzz
14188·Run-length encoding
15260·Tree path array
19458·SnakeCase
25747·IsNegativeNumber
28143·OptionalUndefined
30178·Unique Items
30575·BitwiseXOR
31797·Sudoku
31824·Length of String 3
32427·Unbox
32532·Binary Addition
34286·Take Elements
extreme 17
5·Get Readonly Keys
151·Query String Parser
216·Slice
274·Integers Comparator
462·Currying 2
476·Sum
517·Multiply
697·Tag
734·Inclusive Range
741·Sort
869·DistributeUnions
925·Assert Array Index
6228·JSON Parser
7561·Subtract
31447·CountReversePairs
31997·Parameter Intersection
33345·Dynamic Route

감사합니다