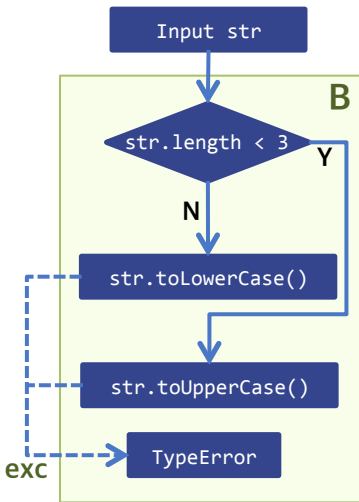


JavaScript 코드도 기존의 커버리지를 만족하도록 테스트하면 충분한가?

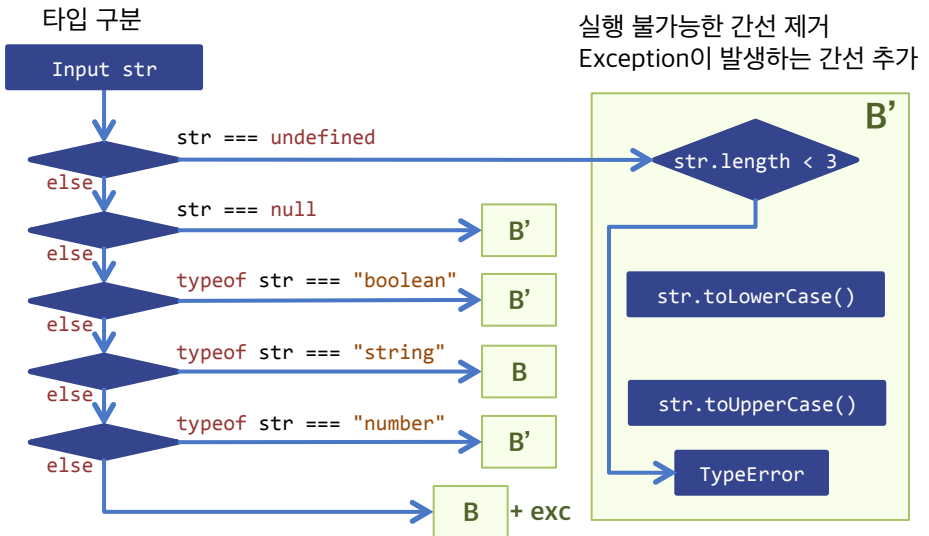
```
//str : symbolic variable
function f (str)
{
  if (str.length < 3) {
    return str.toUpperCase();
  } else {
    return str.toLowerCase();
  }
}
```

Branch Coverage 모든 조건문의 참과 거짓 branch를 실행					
#	str	Result	#	str	Result
1	""	""	3	{ length : -1, toUpperCase : function () { return this.length; } }	-1
	"ABC"	"abc"			
2	"ab"	"AB"		{ toLowerCase : function () { return 1; } }	1
	"AAA"	"aaa"			

기존 코드 커버리지



새로운 코드 커버리지



장점	단점
<ul style="list-style-type: none"> 프로그램의 더 많은 행동을 커버 - Type, Exception 	<ul style="list-style-type: none"> 실행 불가능한 간선을 제거하기 어려움 생성해야 할 테스트 케이스 양이 많아짐 - 보완책 : 버그를 찾기에 효과적인 테스트 케이스를 우선적으로 생성하는 Concolic 테스팅 기법을 제시

새로운 Concolic 테스팅 기법

$$Information = Loc \mapsto 2^{Type}$$

```
Type ::= undefined
       | null
       | boolean
       | string
       | number
       | ObjectId
```

ObjectId = UInt

ObjectIdMap = ObjectId \mapsto Object

Object = String \mapsto $2^{Type} \times Boolean$

#	호출 코드	타입 우선순위
1	f("ab"); f("AAA");	string ...
2	f({ length: 0, toUpperCase: function() {}, toLowerCase: function() {} });	{ length: number, toUpperCase: {...Call:...}, toLowerCase: {...Call:...} } > object > ...
3	f(""); f({});	string > {} > object > ...