

# 이벤트 순서를 고려한 웹애플리케이션 정적 분석

선광원, 류석영

KAIST PLRG

# 이벤트 순서를 고려한 분석

- 웹 애플리케이션은 사용자와의 상호동작에 따라 수행되는 코드의 비중이 높음
- 이벤트 순서를 고려하지 않는 분석 보다는, 순서를 고려하여 분석 → 정확도 개선

# 예를 들자면

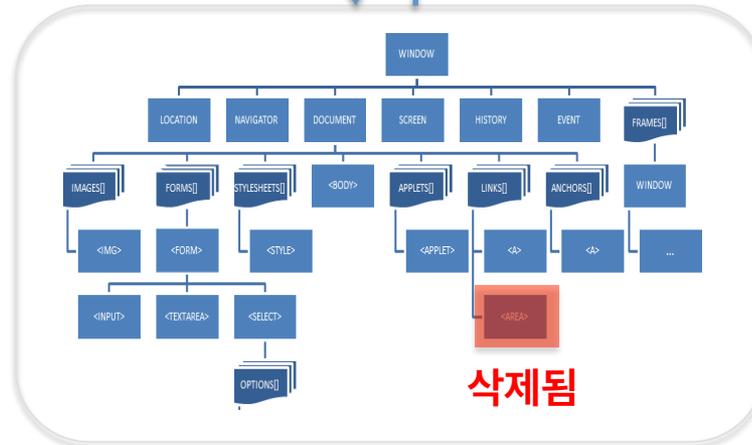
선행 이벤트

`document.remove (HTMLObject)`

후행 이벤트

`document.getElementById (HTMLObject)`

**TypeError**  
: null is not an object



**DOM** (Document Object Model)

# 단순한 문제 같지만...

- 이벤트 개수와 이벤트간의 조합이 많이 존재하는 경우, 개발자가 충분히 검토하기 어려움
- 실제 웹 애플리케이션에서 발생하는 버그중 DOM과 관련된 문제가 가장 많고(65%)[1]  
그 중 상당수는 기본적인 DOM 조작 코드임

# 접근방향

로드 시점까지의 분석 힙  
(web page load, ready)



## 가능한 이벤트 순서 집합

e1 → e3 → e5 → ...

e3 → e2 → e6 → ...

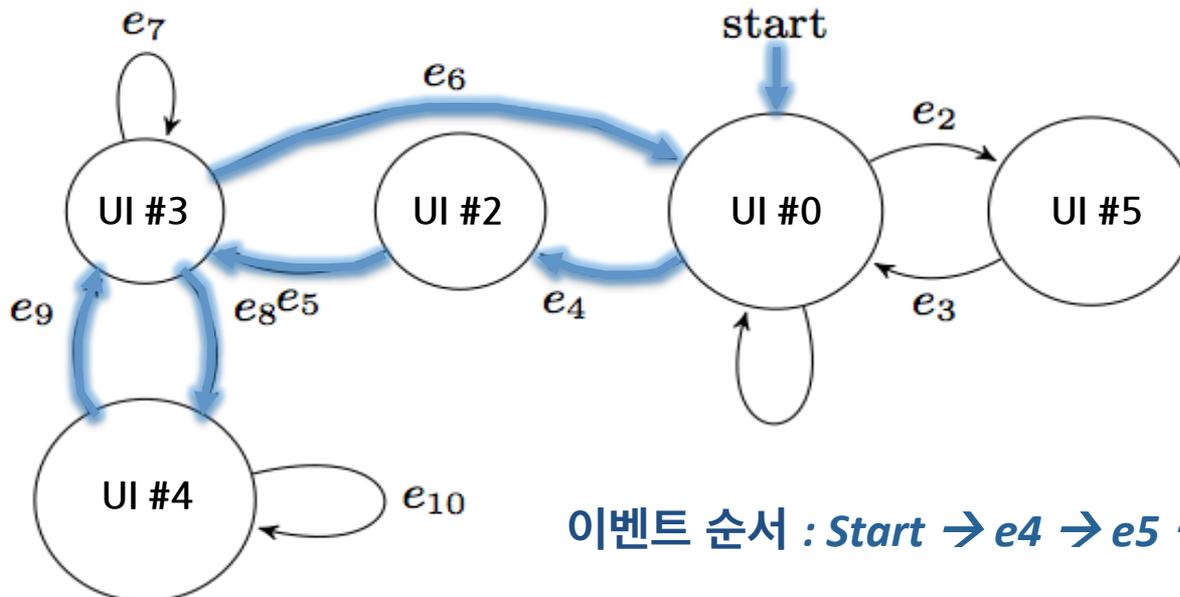
e2 → e4 → e7 → ...

e1 → e3 → e2 → ...

⋮

# 이벤트 순서 도출

- UI 전이 다이어그램 기반
  - 상태: GUI, 전이: 이벤트
- 다이어그램에서 도달 가능한 경로 → 이벤트 순서 집합



이벤트 순서 :  $Start \rightarrow e_4 \rightarrow e_5 \rightarrow e_8 \rightarrow e_9 \rightarrow e_6$

# 앞으로의 할일

- 버그 저장소 분석을 통해, 목표 결함을 구체적으로 정의
  - 선행 이벤트: DOM 요소 무효화 → 후행 이벤트: 그 요소 사용
    - 무효화: 없는(혹은 삭제/변경된), disabled된, invisible한
- 웹애플리케이션으로 부터 UI 전이 다이어그램 도출
  - 정적 혹은 동적으로
- 덜 정확한 UI 전이 다이어그램 기반, 이벤트 순서와 이를 만족 시키는 입력값/Constraints 도출을 위한 방법 고안
- 체커들 + 이벤트 순서 → 검증

감사합니다