

# DLFL 학습 데이터 구축 **가속화**를 위한 LLVM IR **변이체** 활용 및 분석 연구

---

양희찬, 이아청, 김문주

SWTV GROUP, KAIST

# 결함 위치 탐지 (FL: Fault Localization)

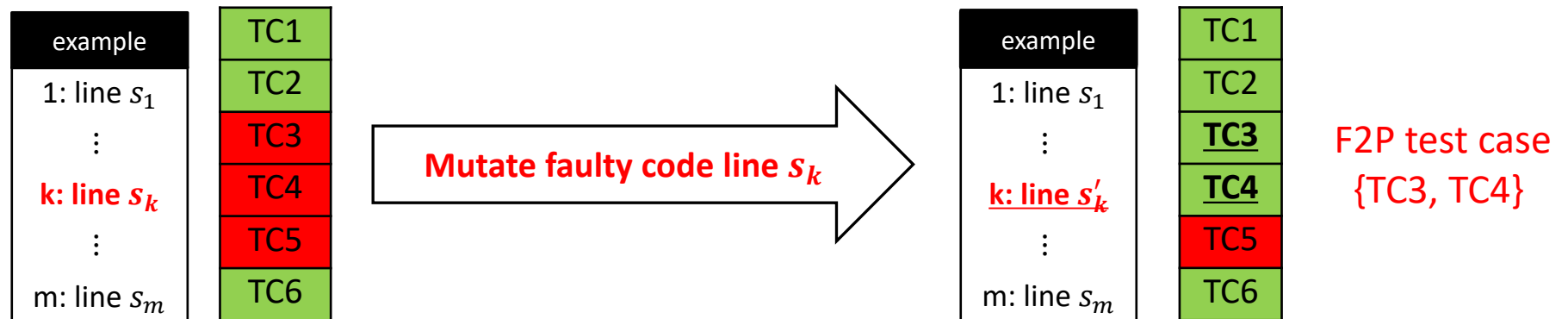
---

- 결함 위치 탐지: 프로그램 실패의 원인이 되는 코드를 식별하는 기술
  - 스펙트럼 기반 (SBFL: Spectrum-Based FL): 실패 및 통과 테스트들의 커버리지 기반 탐지 기법
  - 변이 기반 (MBFL: Mutation-Based FL): 변이 프로그램의 테스트 실행 결과 기반 탐지 기법
  - 딥러닝 기반 (DLFL: Deep Learning-Based FL): 프로그램의 특징(SBFL, MBFL)을 기반으로 학습시킨 신경망 모델 기반 탐지 기법

# MBFL 기법

- 핵심 아이디어 – 2가지 유형의 동작 변화 관찰:

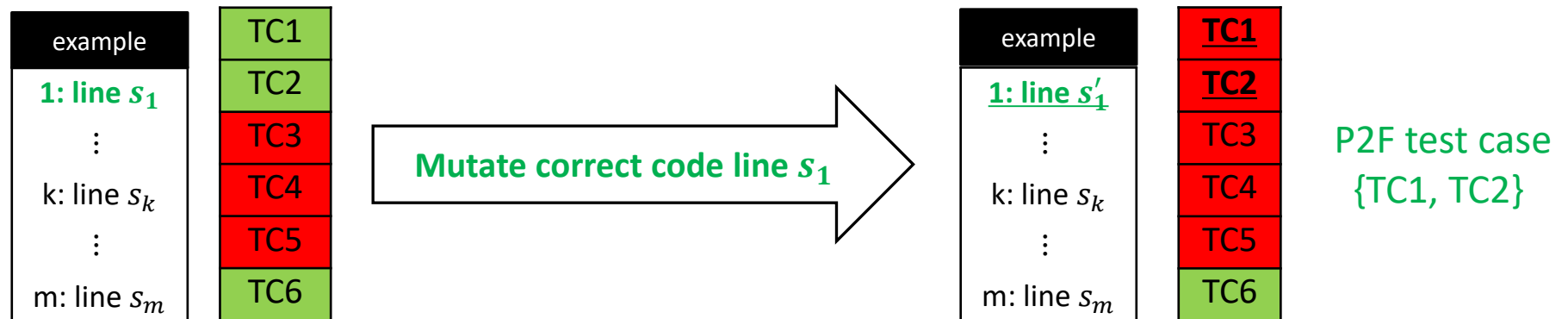
1. **F2P (Fail to Pass):** 결함 코드를 변조하면 부분적으로 문제가 해결(partial-fix)될 가능성이 높다



# MBFL 기법

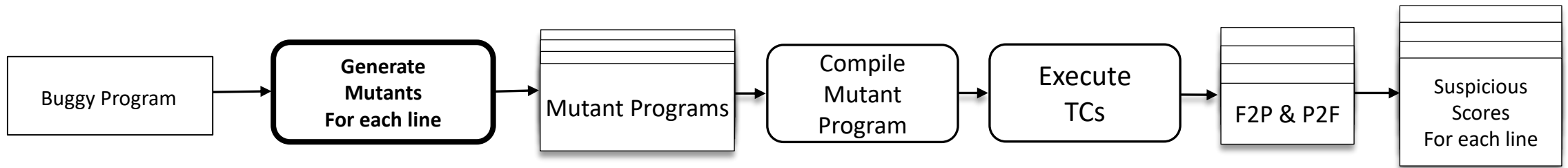
- 핵심 아이디어 – 2가지 유형의 동작 변화 관찰:

1. **F2P (Fail to Pass)**: 결함 코드를 변조하면 **부분적으로 문제가 해결(partial-fix)**될 가능성이 높다
2. **P2F (Pass to Fail)**: 올바른 코드를 변조하면 **새로운 실패가 발생할** 가능성이 높다



# MBFL 특징 추출 과정의 문제

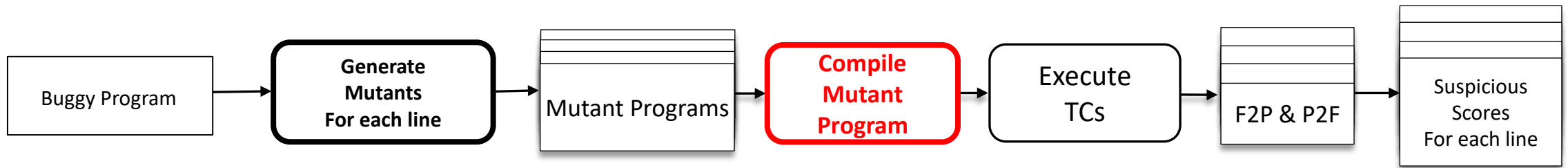
- 문제: 막대한 시간 비용



- 각 코드 라인에 생성된 **N개의 변이 프로그램을 컴파일하고 테스트 스위트를 실행하여** 의심도 점수를 측정한다.
- 이 과정 (**라인 수 x 변이 개수 x (컴파일 + 테스트)**)은 막대한 비용이 사용된다

# MBFL 기법의 문제

- 문제: 막대한 시간 비용



- 컴파일 시간의 비중: 변이 특징을 추출하는 전체 과정 중, 컴파일 시간이 약 **75.45%**를 차지한다.
- 해결 방법 (LLVM IR 활용):
  - **소스 코드 수준**이 아닌 **LLVM IR(Intermediate Representation) 수준**에서 변이를 생성함으로써 반복적인 **컴파일 과정을 생략**할 수 있다.

# 실험 결과

---

1. **[효율성]** LLVM IR 수준의 변이를 통해 컴파일 과정을 생략함으로써 MBFL 특징 추출 시간(Wall-clock time)을 기존 4.22시간에서 1.65시간으로 **60.9%** 단축했다.  
(28대 머신, 16 동시 프로세스 기준)
2. **[유효성]** LLVM IR 수준의 특징을 사용했을 때, 소스 코드 수준 대비 Top-1 정확도가 **5.2%p** 향상(60.3% → 65.5%)되는 유의미한 결과를 보였다.
3. **[기여도]** MBFL 특징을 DLFL에 학습 특징으로 사용할 경우, 제외하고 학습하는 것에 대비 Top-1 정확도가 **8.0%p** (57.5% → 65.5%) 상승했다.

# Thank You

Poster Session →

## DLFL 학습 데이터 구축 가속화를 위한 LLVM IR 변이체 활용 및 분석 연구

양희찬, 이아청, 김문주 (KAIST)

### 1. 서론

#### • 문제:

- 딥러닝 기반 결함 위치 추정(DLFL)은 차세대 기술로 주목받고 있으나, 모델 학습을 위한 변이체(Mutant) 특징 추출 과정에서 **막대한 시간 비용**이 발생한다.

#### • 동기:

- LLVM IR 수준에서 변이체 특징을 추출하면 별도의 컴파일 과정이 필요 없어 소스 코드 수준 대비 추출 시간을 크게 단축할 수 있다.
- 그러나 두 수준의 변이체가 DLFL 정확도에 미치는 영향에 대한 비교 연구는 아직 미흡한 실정이다.

#### • 목표:

- 인공 결함을 대상으로 LLVM IR 및 소스 코드 수준의 변이체가 DLFL 정확도에 미치는 영향을 비교 분석한다.
- 또한, 상용 소프트웨어인 'CROWN'의 실제 결함 사례에 본 기술을 적용하여 실효성을 검증한다.

### 2. 배경 지식

#### • DLFL (Deep Learning-Based FL):

- 프로그램의 동적 특징을 학습하여 결함 위치를 예측하는 기법.
- DLFL에 사용되는 특징 구성:
  - Spectrum-Based FL: 테스트 케이스의 코드 커버리지 정보.
  - Mutation-Based FL: 변이체에 따른 테스트 실행 결과의 변화 양상.
  - Stack Trace Relevance: 실패한 테스트의 Call Stack 정보.

#### • 변이체 생성 도구:

- MUSICUP: C/C++ 소스 코드 수준의 변이체 생성 도구.
- MUSIC\_LLVM: LLVM IR 수준의 변이체 생성 도구.

### 3. 연구 질문

- RQ1: [효율성] MUSIC\_LLVM은 기존 MUSICUP 대비 변이체 특징 추출의 시간적 효율성을 얼마나 향상시키는가?
- RQ2: [유효성] LLVM IR 수준의 변이체 특징을 활용하는 것이 소스 코드 수준 대비 DLFL 모델의 결함 탐지 정확도에 어떠한 영향을 미치는가?
- RQ3: [기여도] MBFL 특징의 통합이 DLFL 모델의 결함 탐지(Fault Localization) 성능 향상에 실질적으로 기여하는가?

### 4. 실험 세팅

#### • 대상 소프트웨어 및 결함 데이터셋:

- 대상 시스템: 테스트 도구인 CROWN
- 결함 데이터셋: 총 100개의 인공 결함 생성 및 활용
- 프로그램 규모:
  - 코드 라인: 약 2KLoC
  - 함수 개수: 총 159개
- 테스트 케이스: 총 24개 실행 가능한 테스트 스위트 구성

#### • 특징 추출 및 학습 환경:

- 하드웨어 인프라: 대규모 변이체 실행을 서버 28대를 활용
- 검증 방법: 10-Fold Cross Validation 총 10회 반복 수행

#### • 성능 평가 지표:

- Top-N: 각 결함에 대해, Top-N 안에 하나 이상의 결함 함수가 포함된 경우의 수
- MFR (Mean First Rank): 각 결함에 대해 가장 먼저 발견된 구문의 평균 순위

### 5. 주요 결과

#### • RQ1: 특징 추출의 시간적 효율성 제고:

- Finding: MUSIC\_LLVM은 컴파일 과정을 생략함으로써 MBFL 특징 추출 시간(Wall-clock time)을 기존 4.22시간에서 1.65시간으로 약 60.9% 단축하였다.
- Insight: 이는 대규모 소프트웨어 프로젝트에서 DLFL 적용에 시간적 병목 현상을 획기적으로 해결할 수 있음을 시사한다.

criteria	MUSICUP	MUSIC_LLVM
Total # of mutants tested	309,800	224,148
Wall clock time (hours)	4.22	1.65
CPU hours for		
MBFL extraction	Only Compile 824.95	Only TC exec. 507.80
Total	1,093.34	507.80

#### • RQ2: 변이체 수준에 따른 탐지 성능의 유의미한 차이:

- Finding: LLVM IR 수준의 특징을 사용했을 때, 소스 코드 수준 대비 Top-1 정확도가 5.2%p 향상(60.3% → 65.5%)되는 유의미한 결과를 보였다.
- Insight: LLVM IR 기반 변이체가 소스 코드 수준보다 결함의 특징을 더 세밀하게 반영하거나, 학습 모델에 최적화된 형태의 정보를 제공함을 입증하였다.

criteria	MUSICUP	MUSIC_LLVM
DLFL accuracy (function level)		
Top-1	60.3%	65.5%
Top-3	82.1%	82.8%
Top-5	90.2%	90.8%
MFR	2.53	2.44

#### • RQ3: MBFL 특징의 기여도:

- Finding: MBFL 특징을 추가할 경우, SBFL+ST 조합 대비 Top-1 정확도가 8.0%p(57.5% → 65.5%) 상승하고 MFR은 8.6% 개선되었다.
- Insight: 변이체 정보(MBFL)가 결함 위치 추정 모델의 변별력을 높이는 필수적인 특징임을 데이터로 증명하였다.

Feature Group	Top-1(%)	Top-3(%)	Top-5(%)	MFR
SBFL+ST	57.5	84.4	91.0	2.67
SBFL+ST+MBFL	65.5	82.8	90.8	2.44

### 6. 사례 연구: 상용 SW CROWN 적용 및 실효성 검증

#### • 결함 유형: 타입 불일치 (Type Mismatch) 결함

#### • 주요 분석 결과:

- 제안하는 DLFL 모델은 버그 함수를 높은 순위로 (2.2, 3.6) 예측하여 개발자의 디버깅 비용을 효과적으로 절감하였습니다.

FL technique	MUSICUP	MUSIC_LLVM
DLFL	2.2	3.6

### 7. 결론 및 향후 연구

#### • 결론:

- LLVM IR 기반 변이체 활용을 통해 DLFL의 비용 효율성(시간 60.9% 단축)과 탐지 정확도(Top-1 5.2%p 향상)를 동시에 제고하였다.
- 상용 소프트웨어(CROWN)의 실제 결함 사례 연구에서 월등한 성능(버그 함수 2.2위)을 기록하며 실무 적용 가능성을 입증하였다.

#### • 향후 연구:

- 대규모 프로젝트 확장: 현재 2KLoC 규모의 검증 결과를 수십만 라인 이상의 대형 산업용 소프트웨어 및 임베디드 프로젝트로 확장하여 기법의 일반화 가능성을 검증할 계획이다.