BayeSmith: 정적 분석 알람을 위한 확률 모델 학습

김현수[1], Mukund Raghothaman[2], 허기홍[1]

[1]KAIST, [2]Univ. of Southern California

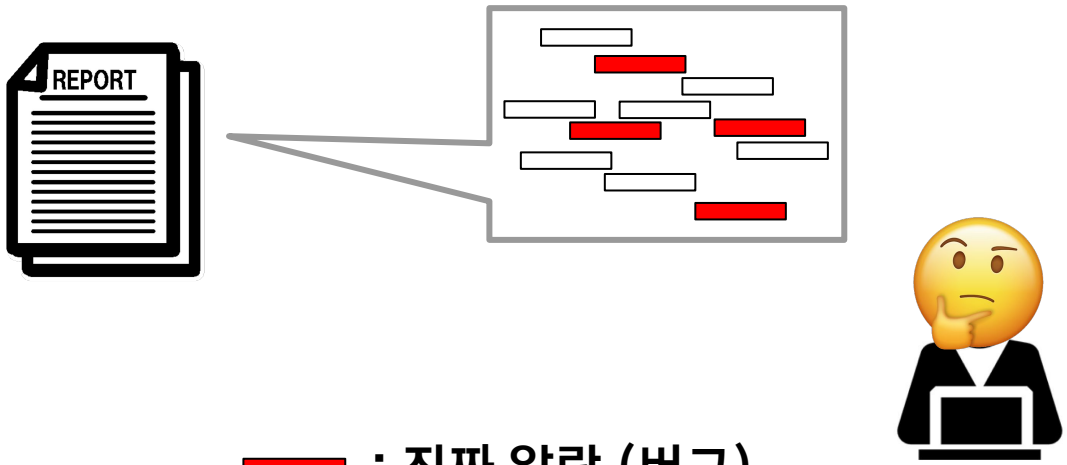
KAIST Programming Systems Laboratory

USC University of Southern California

정적 분석

프로그램 분석기 알람 보고서

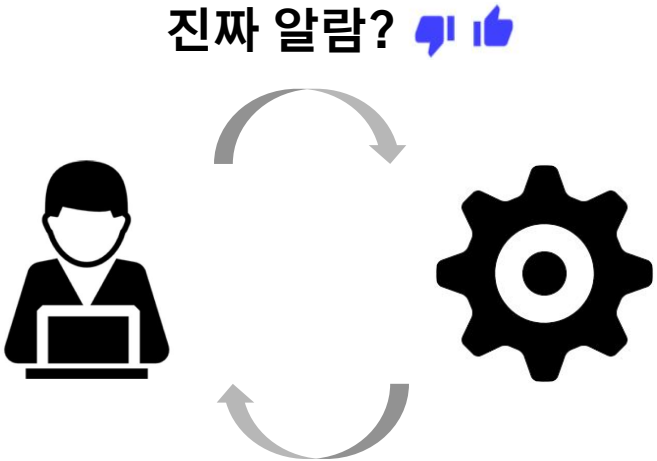
정적 분석의 한계

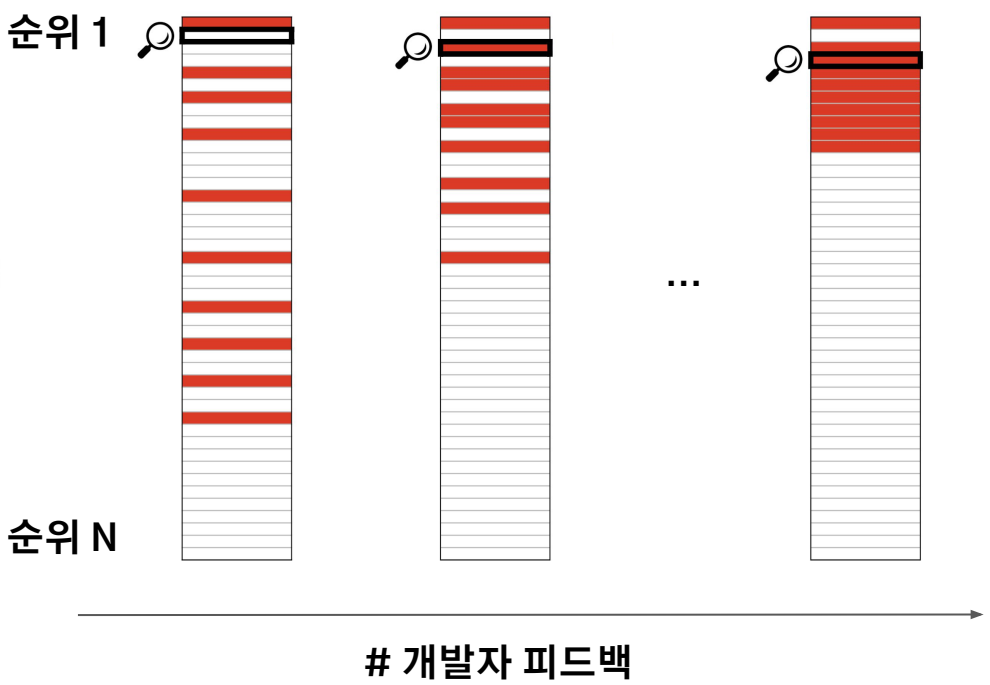


: 진짜 알람 (버그)

: 거짓 알람

베이지안 알람 랭킹 시스템

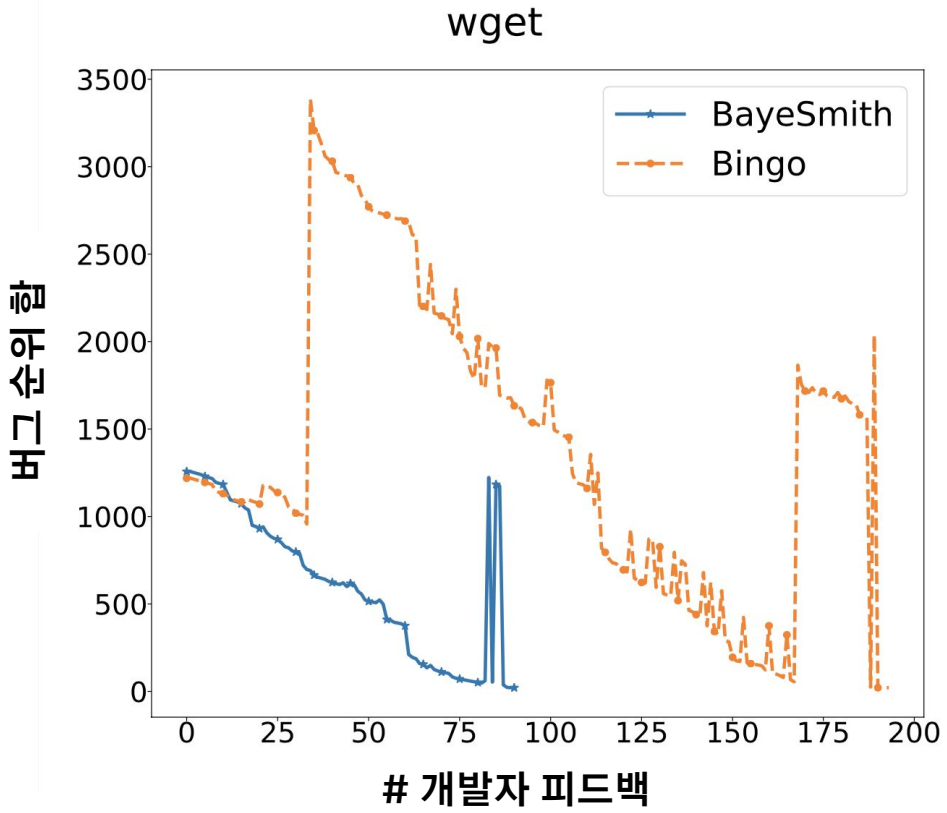




알람 랭킹 시스템의 성능

알람: 891개

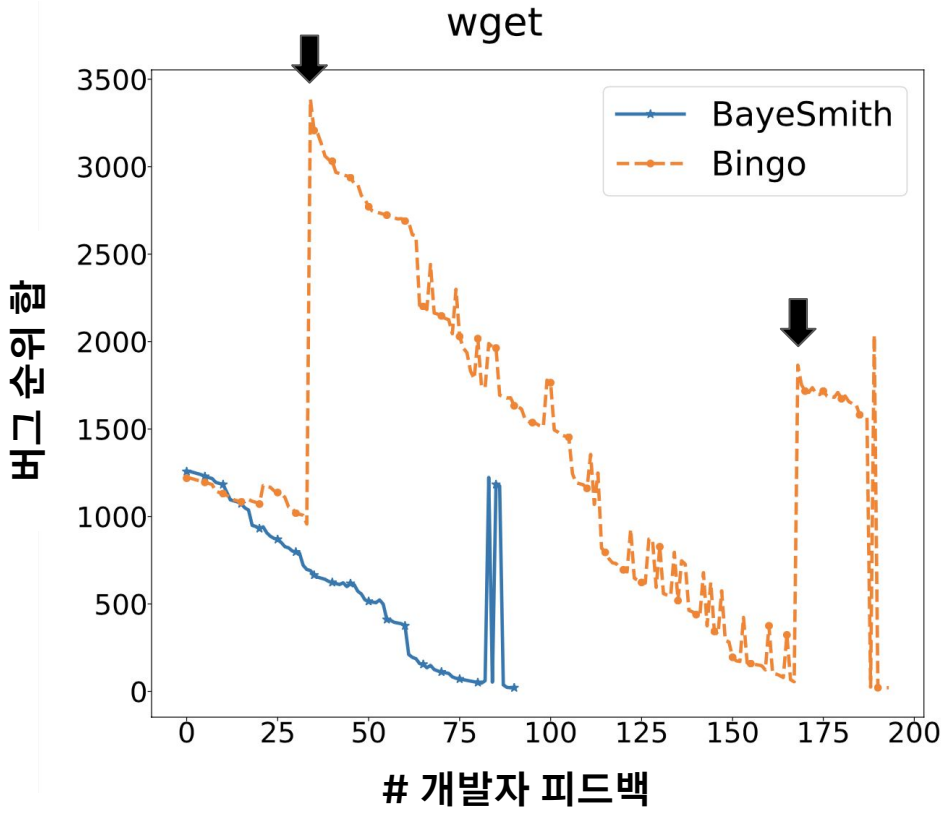
버그: 6개



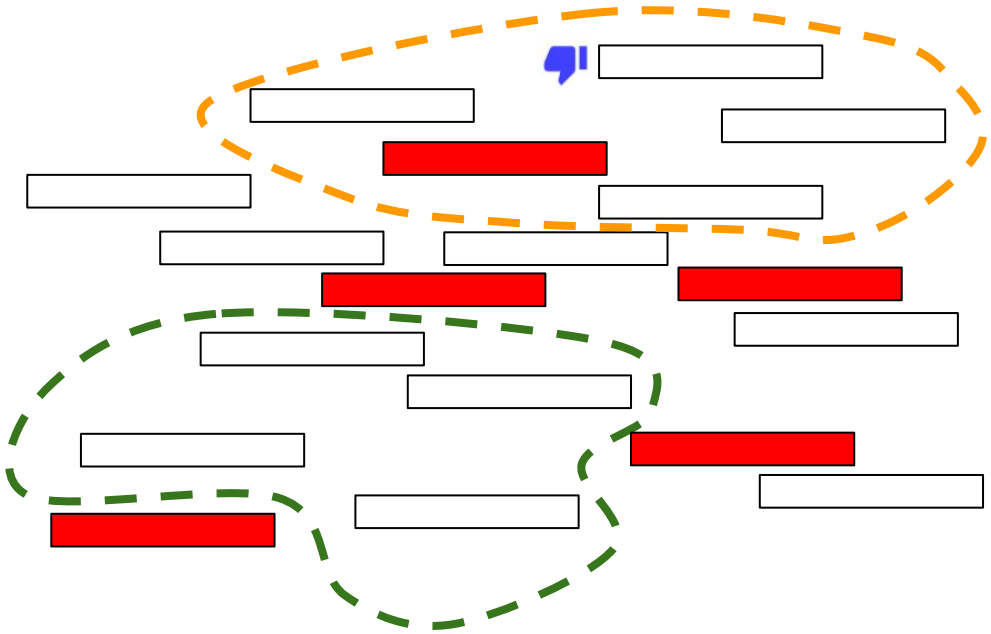
알람 랭킹 시스템의 성능

알람: 891개

버그: 6개



알람 랭킹 시스템의 한계



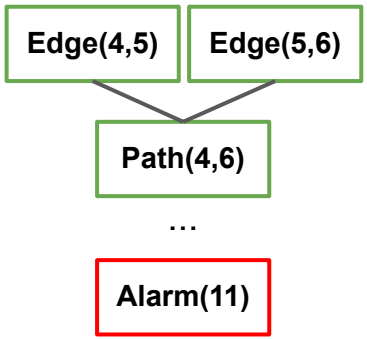
알람 랭킹 시스템 구성 과정

```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, ‘r’);
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, “_”);
6    char *p = tok + strlen(tok);
7    while (p > tok) {
8      if (!c_isdigit(*p)) break;   // false alarm #1
9      p--;
10   }
11   if (*(p - 1) != “^”)           // true alarm (buffer underflow)
12     *p = ‘\0’;                   // false alarm #2
13 }
```

wget-1.2 일부

알람 랭킹 시스템 구성 과정

정의-사용 관계 분석

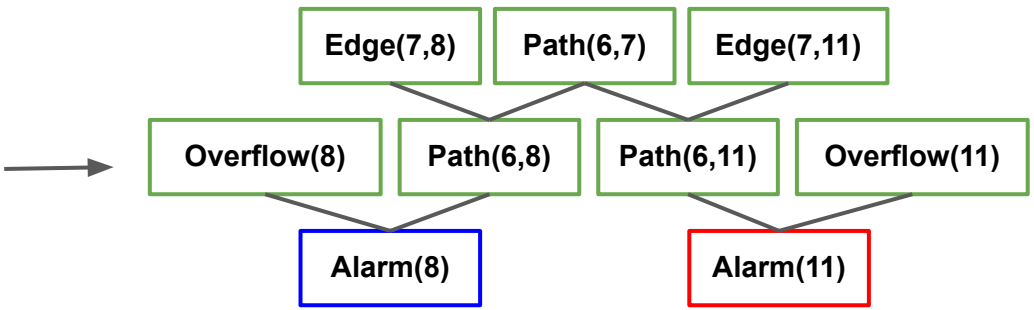



```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, 'r');
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, "_");
6    char *p = tok + strlen(tok);
7    while (p > tok) {
8      if (!c_isdigit(*p)) break;   // false alarm #1
9      p--;
10   }
11   if (*(p - 1) != "^")           // true alarm (buffer underflow)
12     *p = '\0';                   // false alarm #2
13 }
```

wget-1.2 일부

알람 랭킹 시스템 구성 과정

```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, ‘r’);
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, “_”);
6    char *p = tok + strlen(tok);
7    while (p > tok) {
8      if (!c_isdigit(*p)) break;
9      p--;
10   }
11   if (*(p - 1) != “^”)
12     *p = ‘\0’;
13 }
```

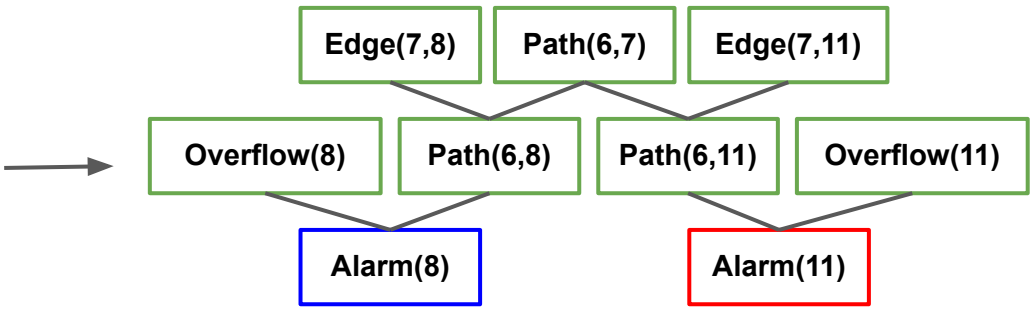


논리 표현
(정의-사용 관계)

알람 랭킹 시스템 구성 과정

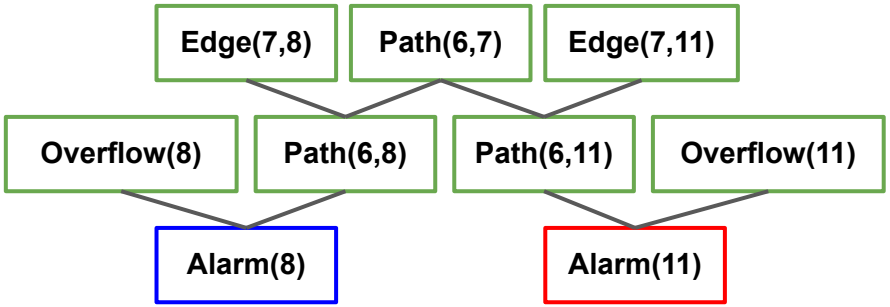
```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, 'r');
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, " ");
11   if (*(p - 1) != "^")
12     *p = '\0';
13 }
```

Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y)
Alarm(y) :- Path(x, y) /\ Overflow(y)



논리 표현
(정의-사용 관계)

알람 랭킹 시스템 구성 과정 – 베이지안 네트워크

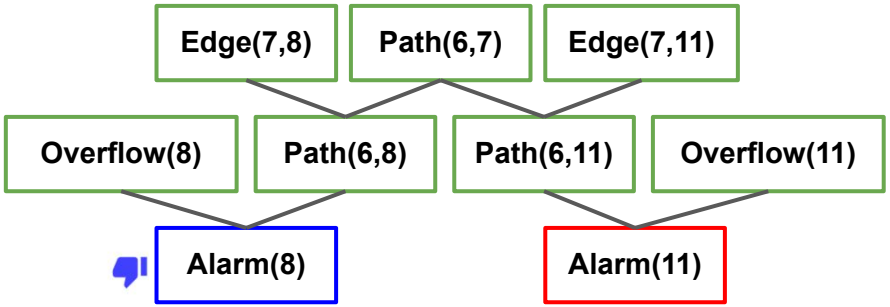


$$\Pr(\text{Alarm}(8)) = \Pr(\text{Alarm}(8) \mid \text{Path}(6,8), \text{Overflow}(8)) \times \Pr(\text{Path}(6,8) \mid \text{Path}(6,7), \text{Edge}(7,8)) \times \Pr(\text{Path}(6,7) \mid \ldots) \times \Pr(\text{Edge}(7,8)) \times \Pr(\text{Overflow}(8))$$

베이즈 법칙의 연쇄 적용

확률 표현
(베이지안 관계)

베이지안 알람 랭킹 시스템의 한계



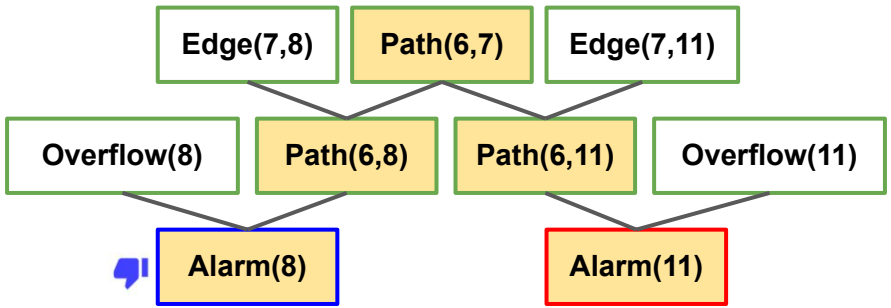
순위	알람	확률
1	Alarm(8)	**0.96**
2	Alarm(11)	**0.81**
...	...	...

➡

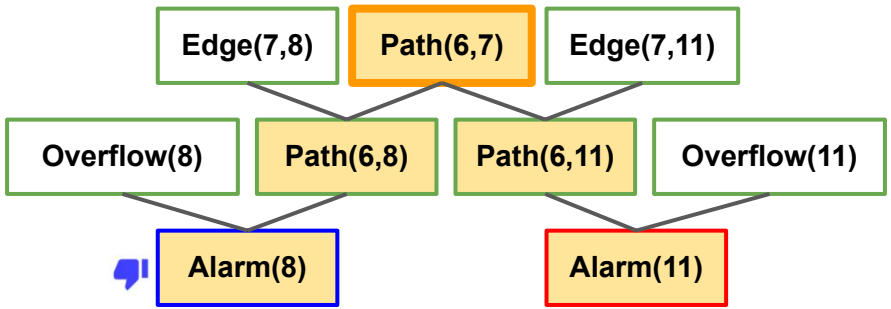
순위	알람	확률
...	...	...
136	Alarm(11)	**0.42**
...	...	...
-	Alarm(8)	-

거짓 일반화 문제
(False generalization problem)

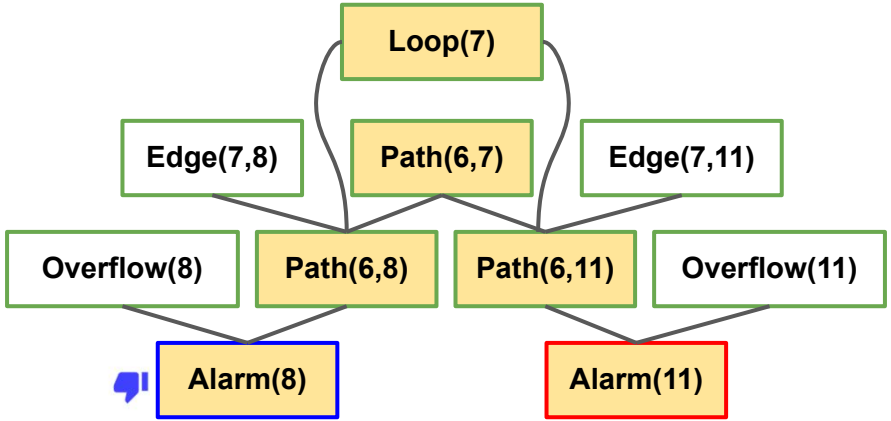
베이지안 알람 랭킹 시스템의 한계



베이지안 알람 랭킹 시스템 개선 방안

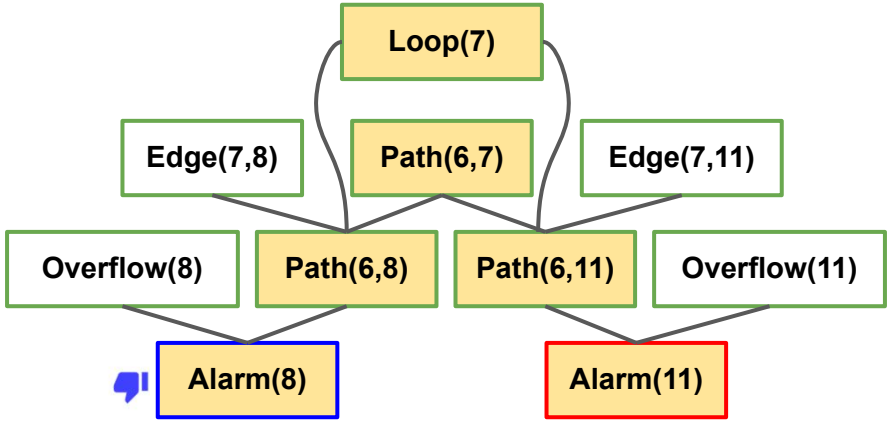


베이지안 알람 랭킹 시스템 개선 방안



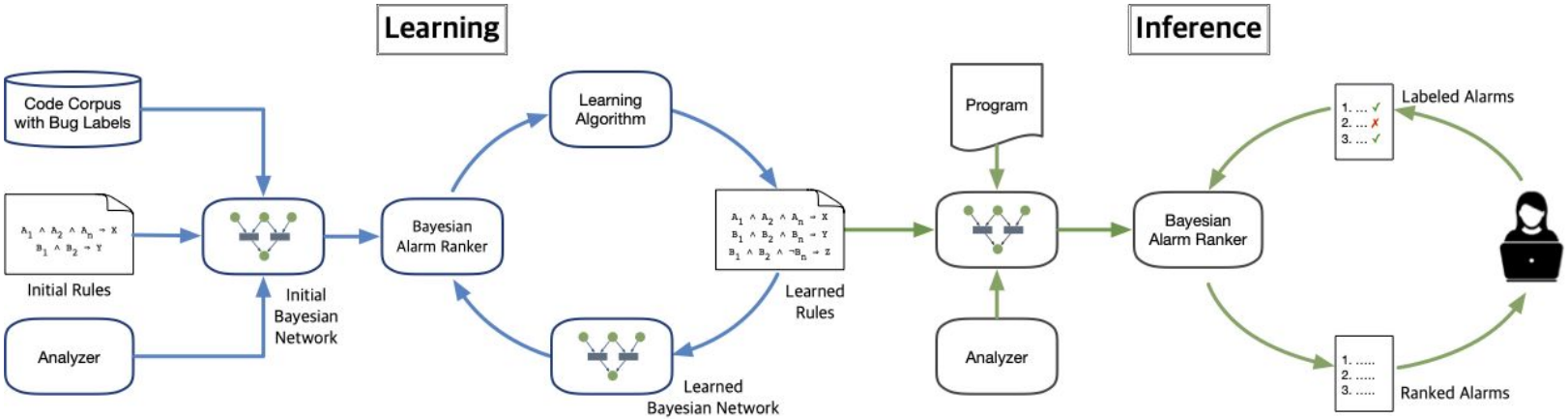
새로운 정보를 추가하여 오탐의 비난을 효과적으로 분산!

베이지안 알람 랭킹 시스템 개선 방안



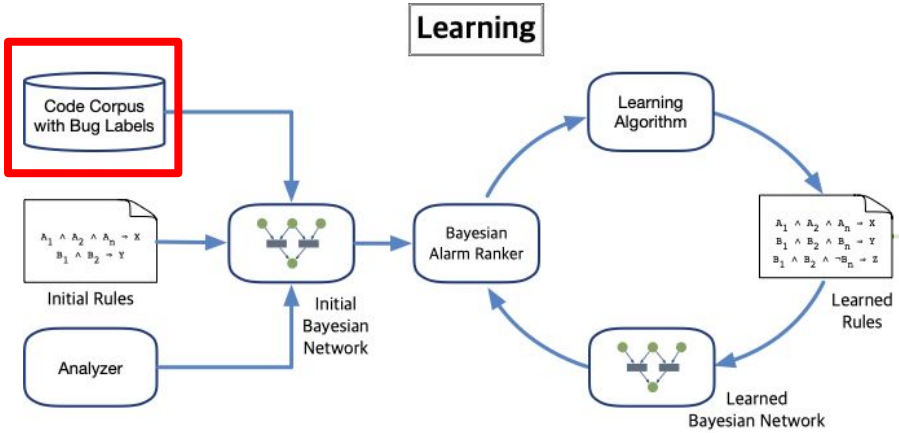
문법 정보를 추가하여 오탐의 비난을 효과적으로 분산!

베이지안 알람 랭킹 시스템 학습 파이프라인



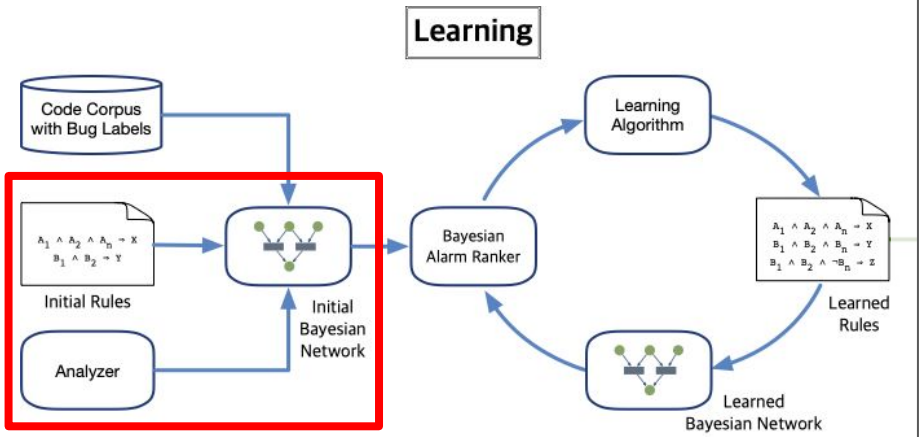
BayeSmith

베이지안 알람 랭킹 시스템 학

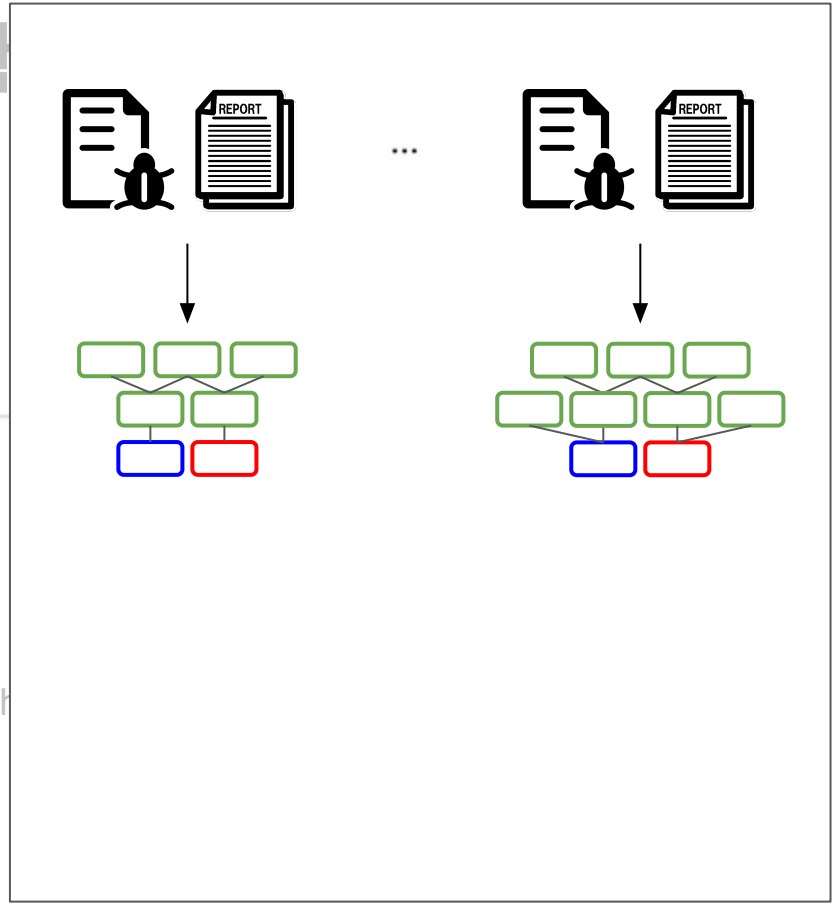


BayeSmith

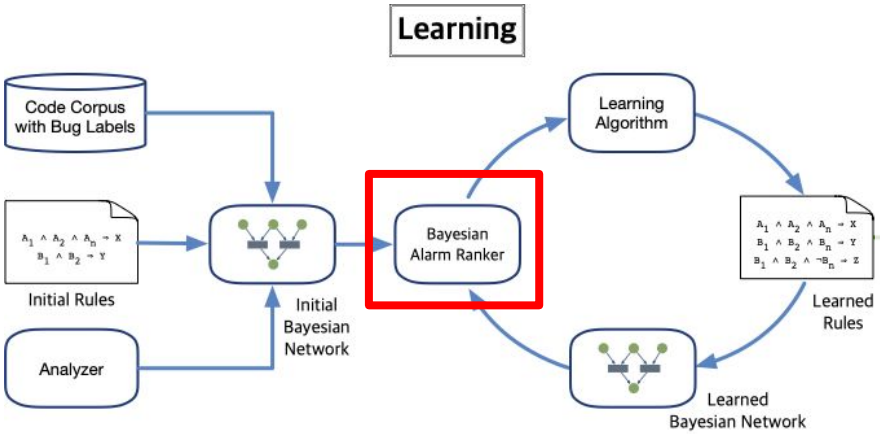
베이지안 알람 랭킹 시스템 학



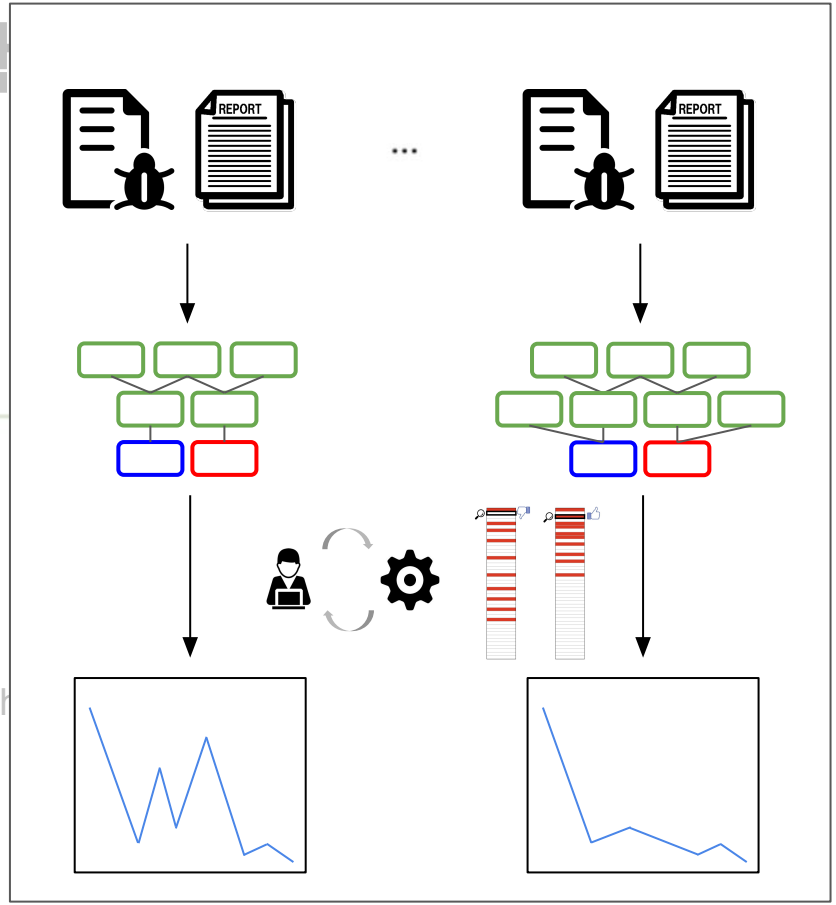
BayeSmith



베이지안 알람 랭킹 시스템 학

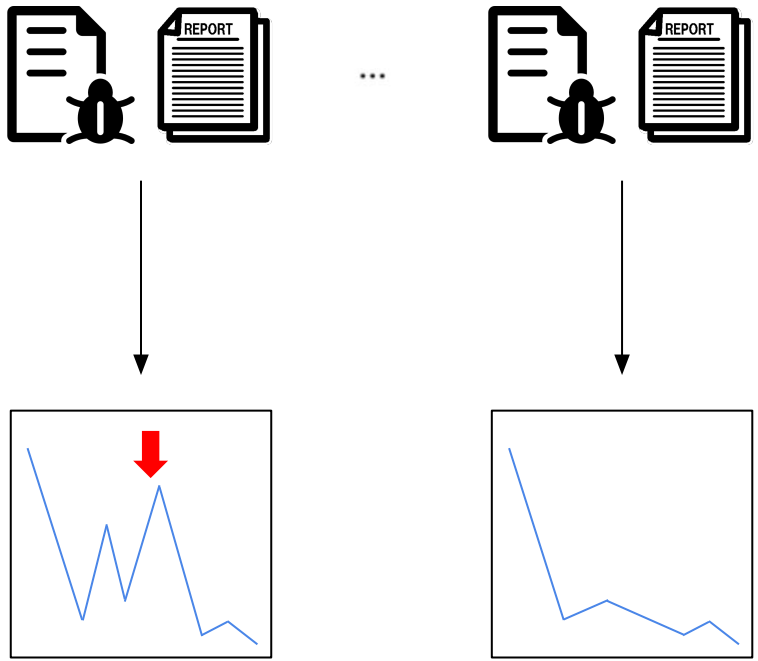


BayeSmith



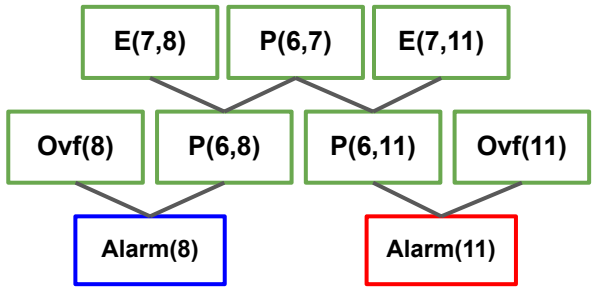
베이지안 알람 랭킹 시스템 학습 파이프라인

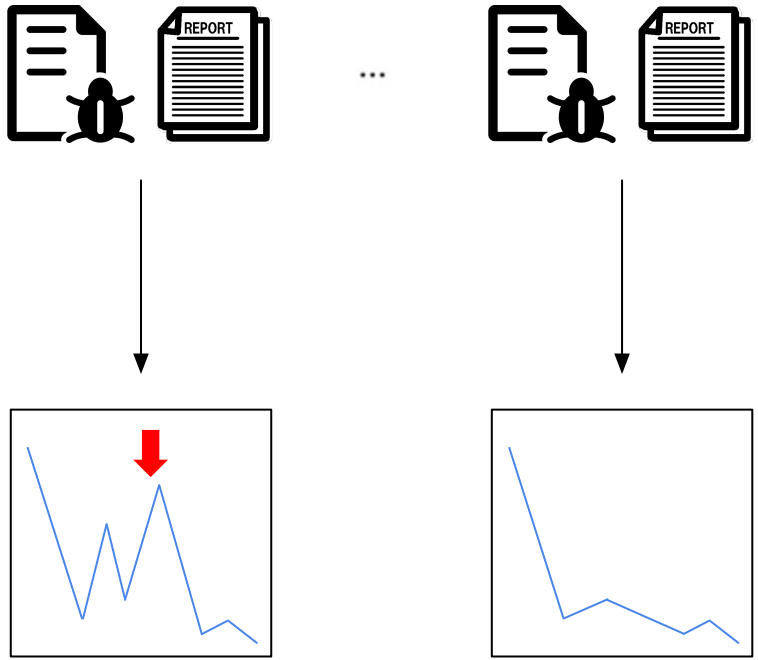
P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y)



베이지안 알람 랭킹 시스템 학습 파이프라인

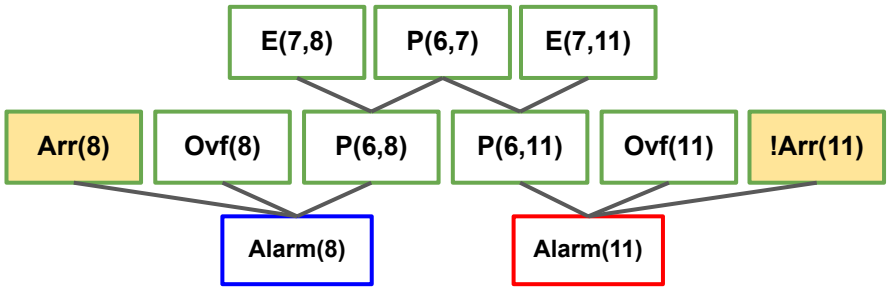
P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y)

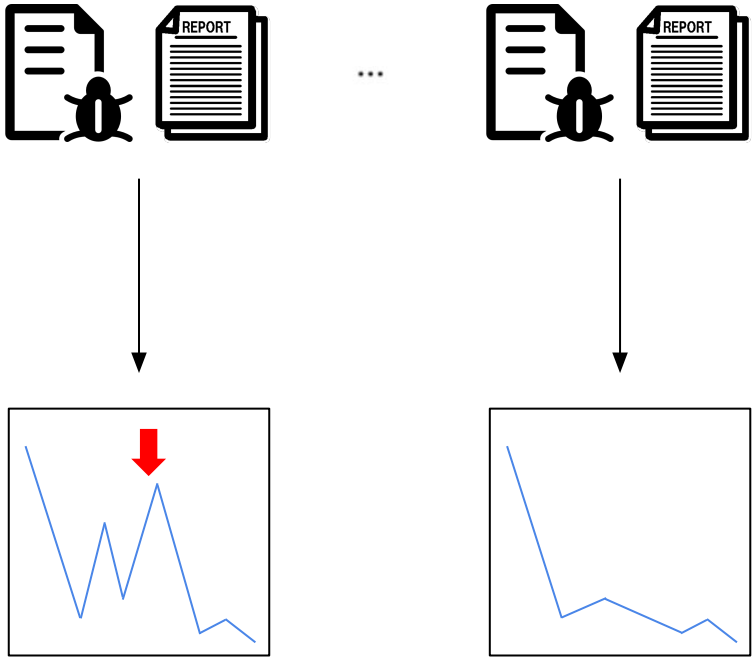




베이지안 알람 랭킹 시스템 학습 파이프라인

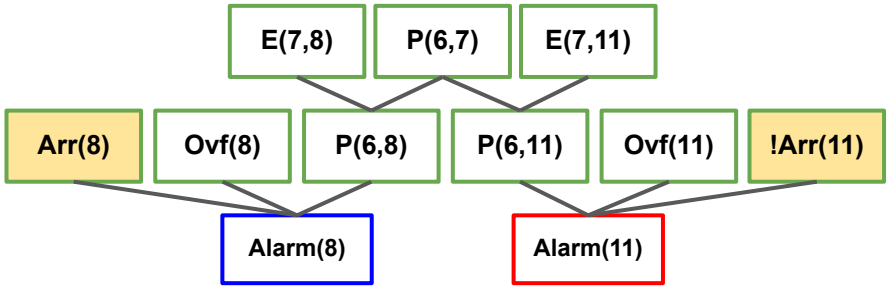
P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y) **/\ Arr(y)**
Alarm(y) :- P(x, y) /\ Overflow(y) **/\ !Arr(y)**





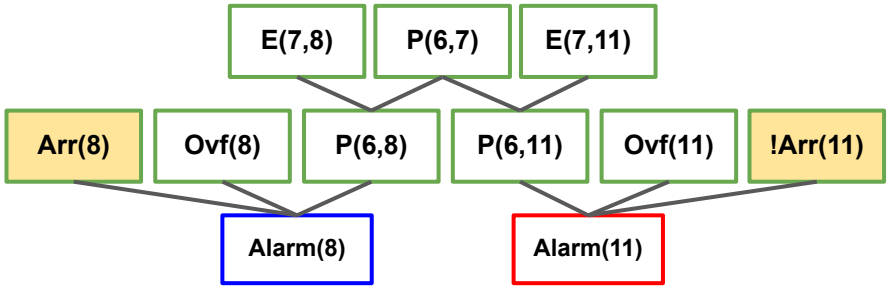
베이지안 알람 랭킹 시스템 학습 파이프라인

P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y) **/\ Arr(y)**
Alarm(y) :- P(x, y) /\ Overflow(y) **/\ !Arr(y)**



베이지안 알람 랭킹 시스템 학습 파이프라인

```
P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y) /\ Arr(y)
Alarm(y) :- P(x, y) /\ Overflow(y) /\ !Arr(y)
```

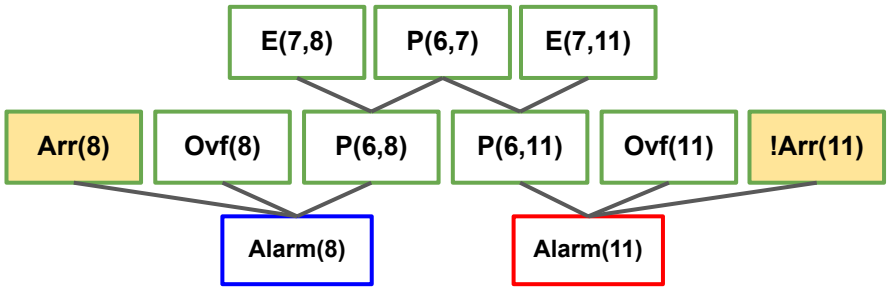


최소 X% 개선
=> 학습 결과 채택

Prog	Before	After
1	145	107
2	6	3
3	54	60
4	122	121

베이지안 알람 랭킹 시스템 학습 파이프라인

P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y) **/\ Arr(y)**
Alarm(y) :- P(x, y) /\ Overflow(y) **/\ !Arr(y)**

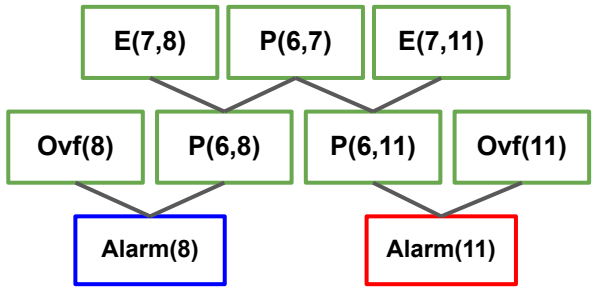


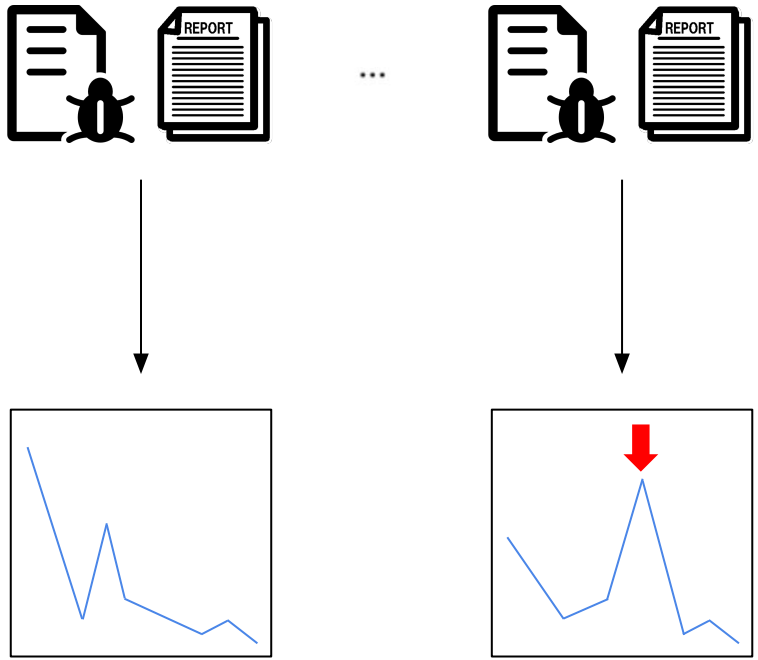
X% 보다 적게 개선
=> 학습 결과 번복

Prog	Before	After
1	145	176
2	6	5
3	54	89
4	122	127

베이지안 알람 랭킹 시스템 학습 파이프라인

P(x, y) :- E(x, y)
P(x, y) :- P(x, z) /\ E(z, y)
Alarm(y) :- P(x, y) /\ Overflow(y)



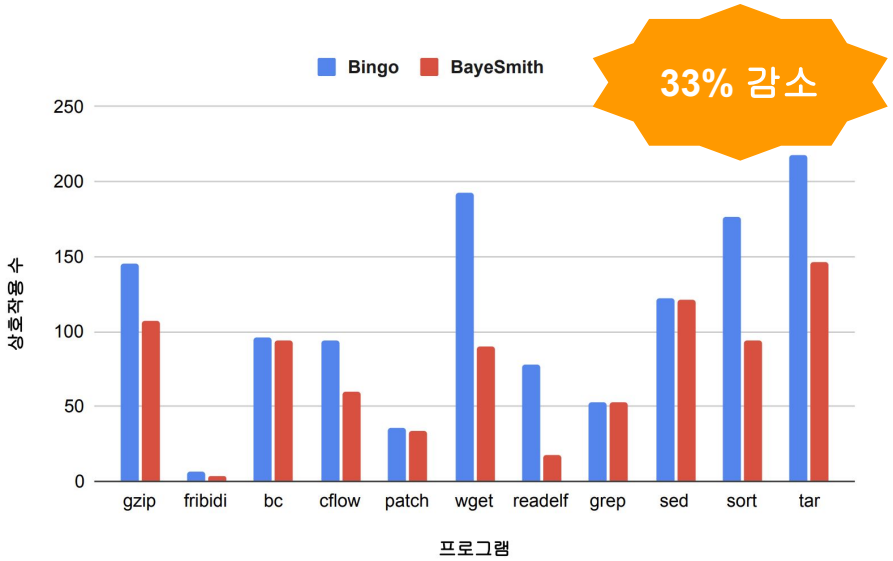


실험 방법

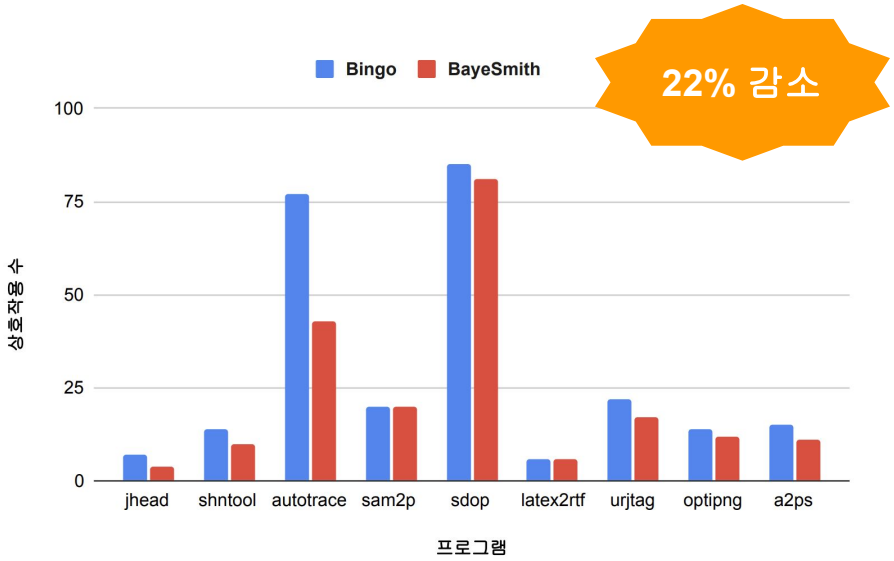
1. 벤치마크 구성: 다양한 크기(9~112 KLoc)의 GNU 프로그램들로 구성
   - 인터벌 분석 (11개), 테인트 분석 (9개)

2. 학습 방법:
   - 한 프로그램을 테스트 데이터, 나머지를 훈련 데이터

실험 결과 – 인터벌 & 테인트 분석



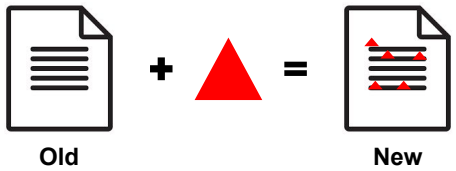
인터벌 분석

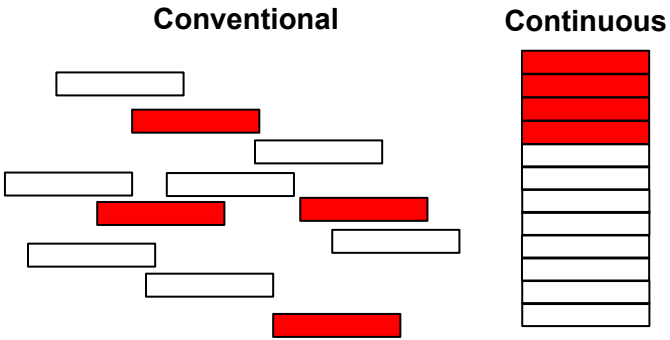


테인트 분석

개발자 피드백 + ?

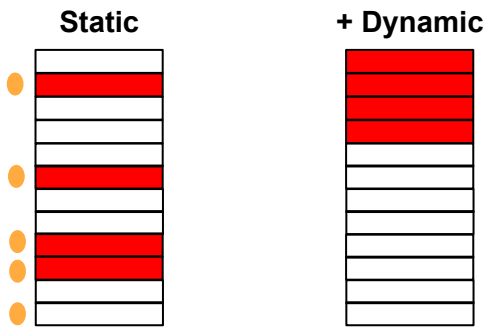
소스 코드 변화





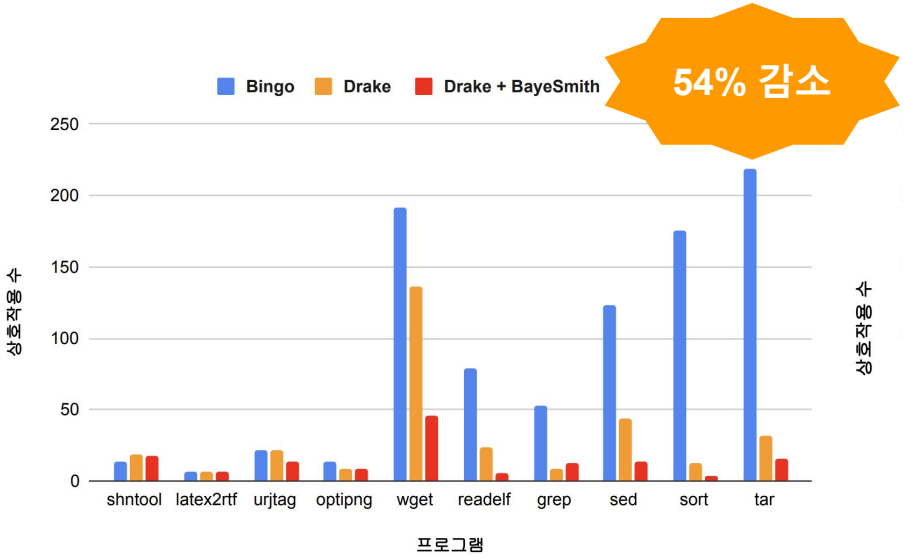
Drake
[PLDI'19]

실행 결과

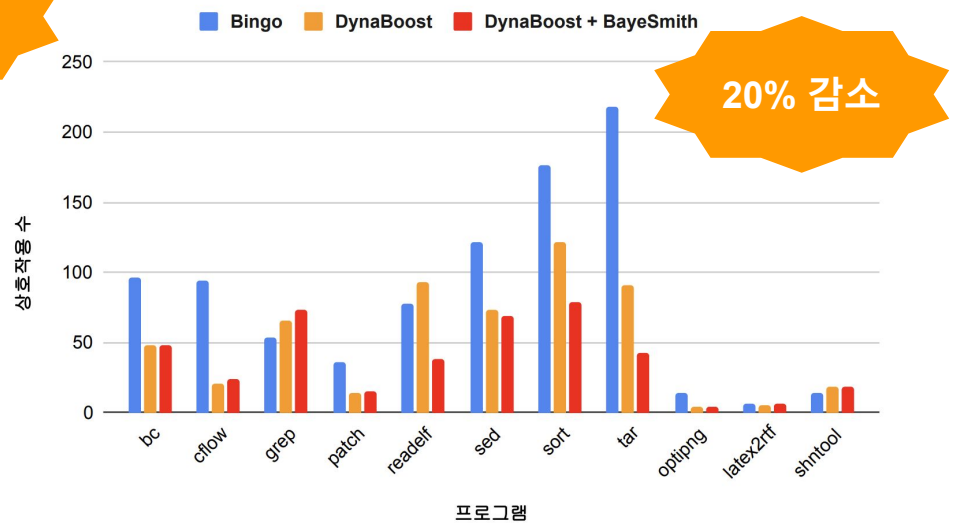


DynaBoost
[FSE'21]

실험 결과 – Drake & DynaBoost



Drake



DynaBoost

감사합니다