

정적 분석 경보 사이의 종속 관계를 이용한 대표 경보 찾기

이우석, 김유일
서울대학교 프로그래밍 연구실

서론

- 정적 분석기는 허위 경보 포함, 많은 오류 경보를 생성한다.
- 경보 간 관계를 이용하여 대표 경보만 보여주면 보고되는 경보 수를 줄일 수 있다.

경보들 사이의 관계?

버퍼 인덱스들 사이의 관계

```
upad [21632];
void residual
(double *u, *ac, *ax, *ay, *q;)
{
    int nx = 50;
    u = &upad[nx+2];
    for (...)
    {
        r[0]=u[0]-u[-1]-u[1]-u[-nx-2]-u[nx+2];
        u++;
    }
}
```

복수 개의 경보가 영향끼침

```
cboard[64];
epash[64];
void MakeMove
(int side, int *move)
{
    f = ...;
    t = ...;
    fpiece = cboard[f];
    tpiece = cboard[t];
    ...
    if (abs(f-t) == 16)
    {
        sq = (f + t) / 2;
        HashKey ^= ephash[sq];
    }
}
```

함수 경계를 넘나드는 관계

```
invmergerules[8];
int lookup_mergearcs (char *rule)
{
    for (int i = 1; invmergerules[i]; i++)
        if (strcmp(rule, invmergerules[i]))
            return (i);
    return (0);
}
int apply_rule
(char *rule, struct sketch *sketch)
{
    code = [0,7]
    if ((code = lookup_mergearcs (rule)))
        res = rule_mergearcs (sketch, code, rcount);
}
```

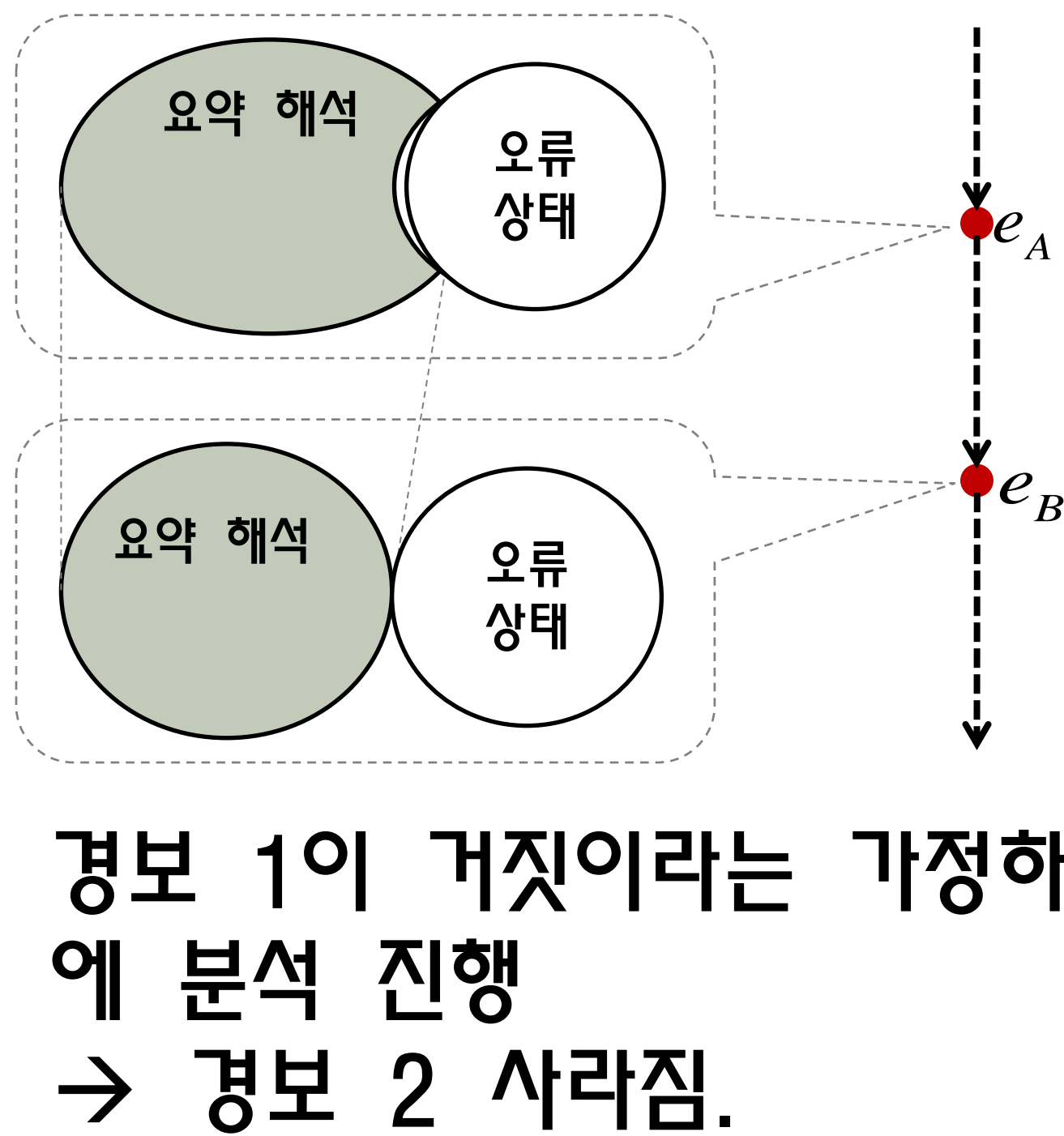
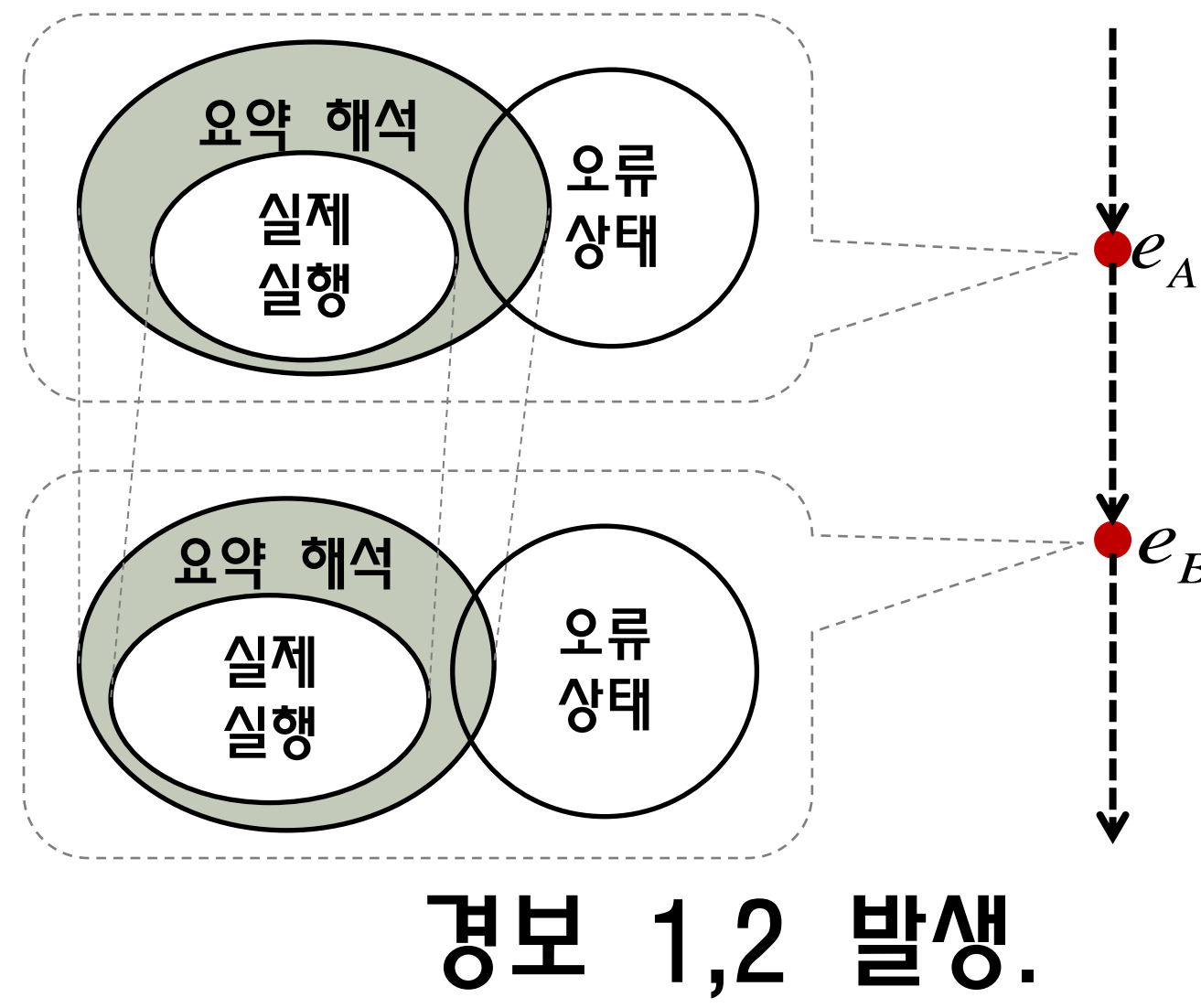
변수 간 관계

```
static int
is_unterminate_line
(const char *line)
{
    int i, len;
    len = strlen(line);
    if (line[len - 1] != '\n')
        return 0;
    for(i = len - 2; i >= 0; i--)
    {
        if (line[i] == '\\') {
            return 1;
        }
    }
    ...
}
```

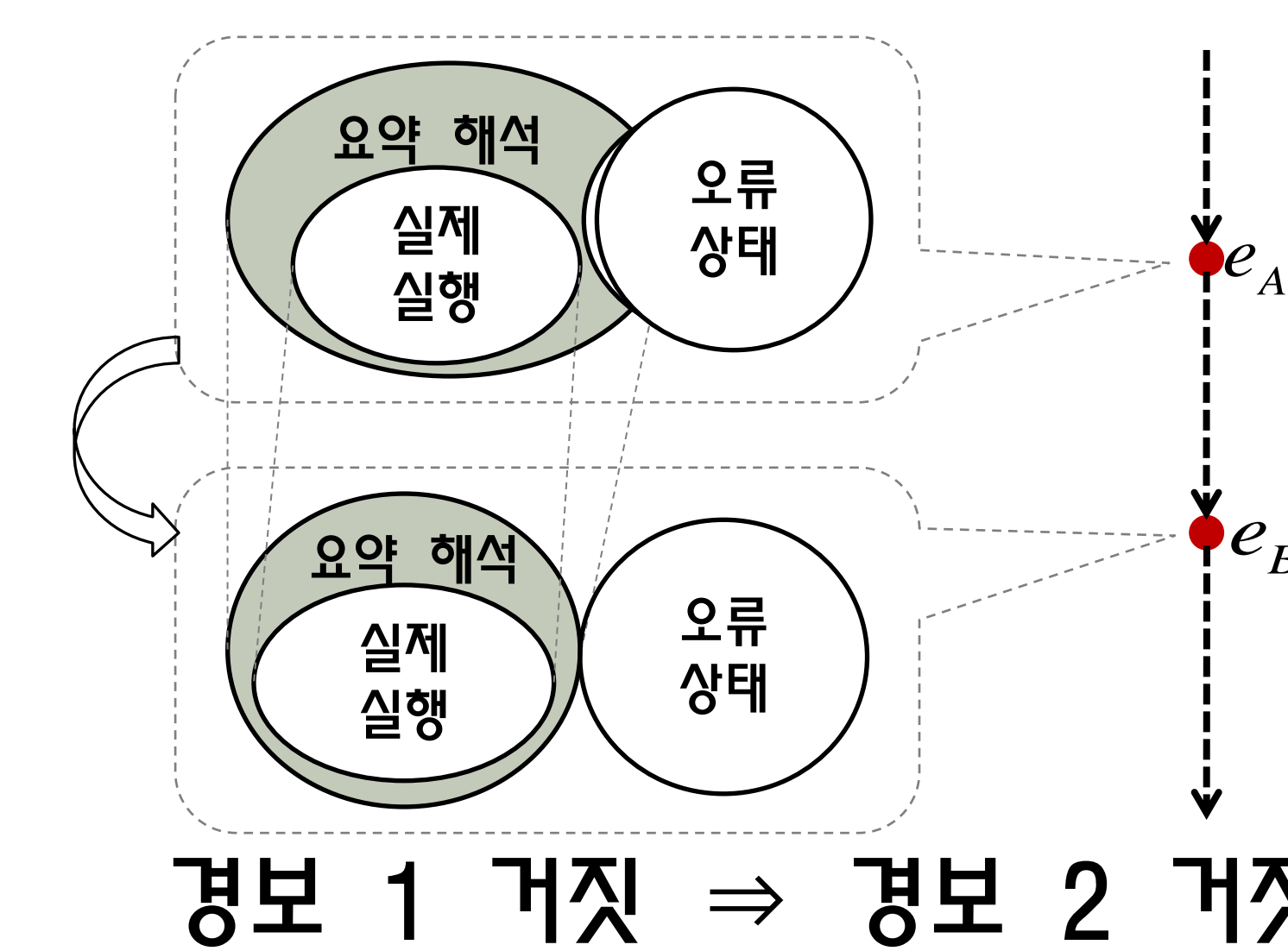
```
int rule_mergearcs
(struct sketch *s, int rule, int rcount)
{
    ...
    if (debug)
        printf("%s\n", invmergerules[rule]);
}
```

방법 및 이론

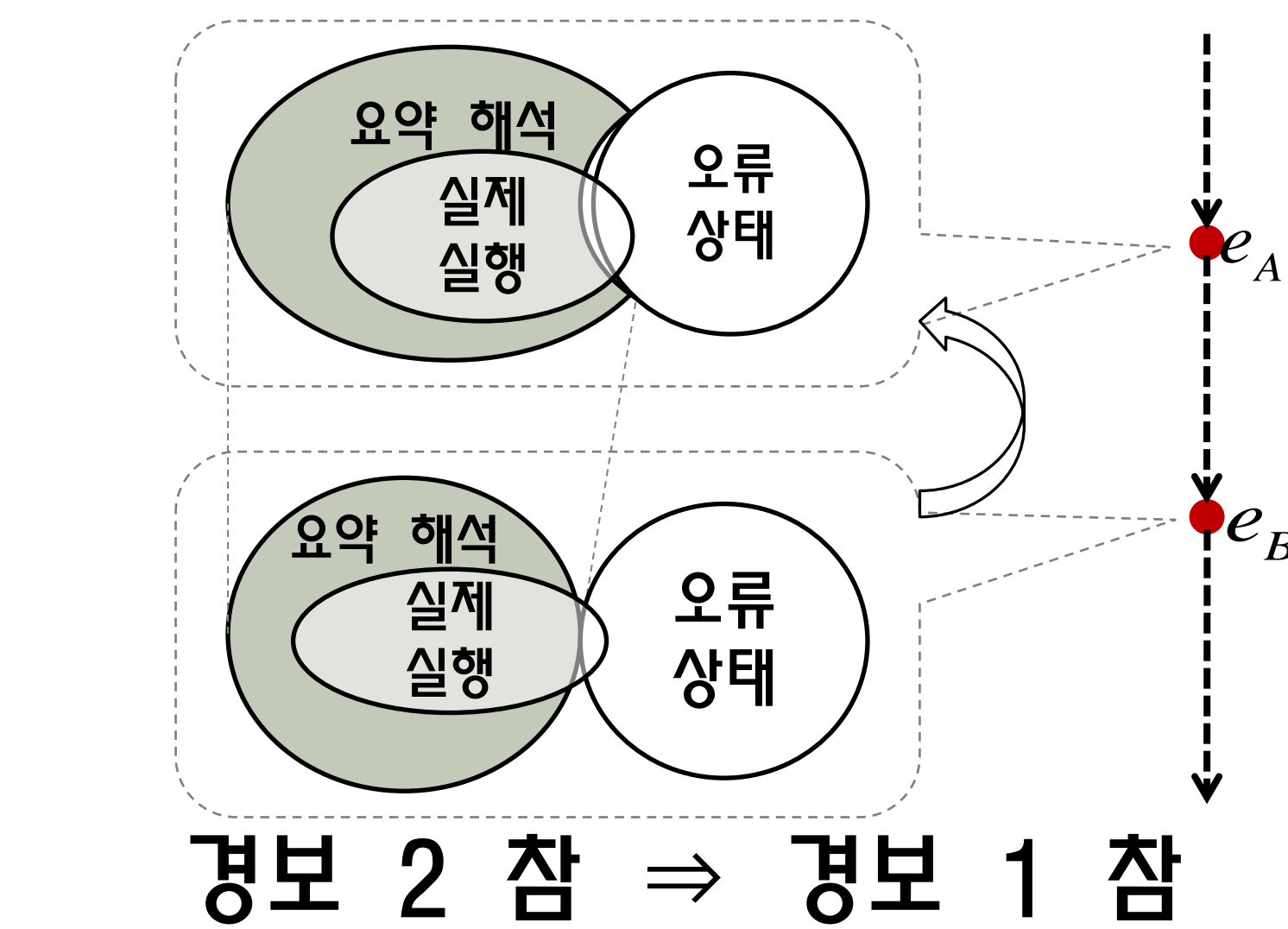
상황.



정리1. 거짓 경보간의 종속 관계



정리2. 오류간의 종속 관계



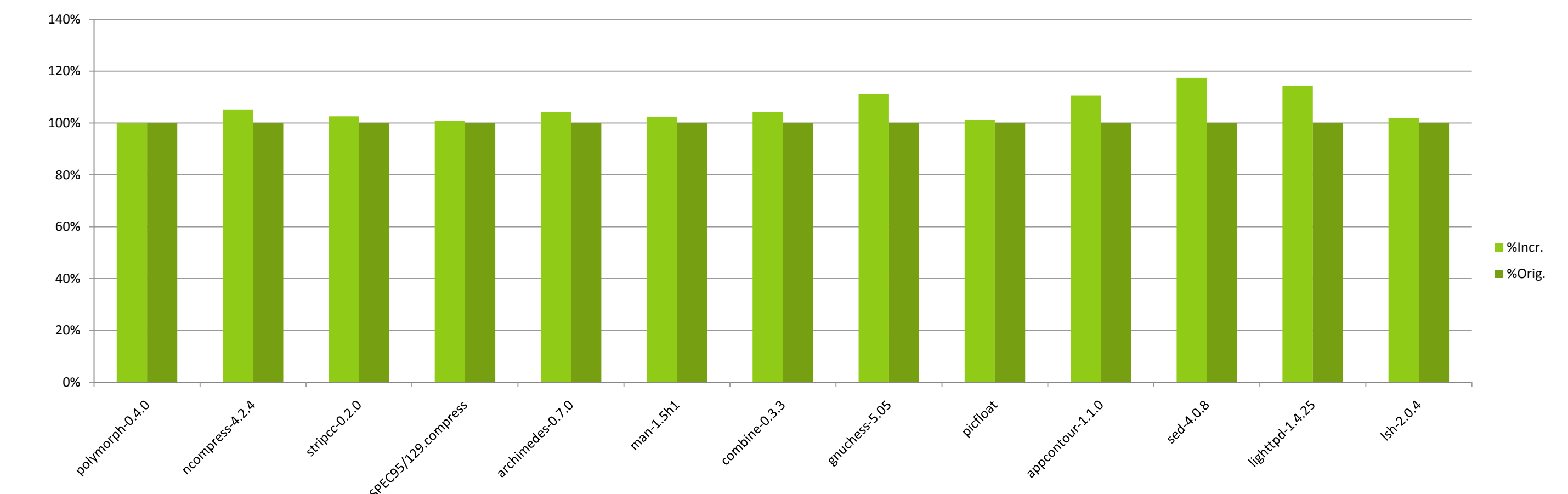
실험 결과

버퍼오버런 분석기 Airac에 적용

프로그램	라인 수	본래 경보 수	최종 경보 수	감소 비율(%)
polymorph-0.4.0	1,357	25	13	48%
ncompress-4.2.4	2,195	66	37	44%
stripcc-0.2.0	2,555	247	155	37%
SPEC95/129.compress	5,585	57	25	56%
man-1.5h1	7,232	276	176	36%
archimedes-0.7.0	7,569	711	215	70%
combine-0.3.3	11,472	733	297	59%
gnuchess-5.05	11,629	972	311	68%
picfloat	11,959	41	21	49%
appcontour-1.1.0	14,930	533	365	32%
sed-4.0.8	25,903	843	562	33%
lighttpd-1.4.25	56,518	1481	652	56%
lsh-2.0.4	110,898	616	317	49%
TOTAL	269,802	6,601	3,146	52%

경보 수 평균 52% 감소

추가 수행 시간



분석 시간 평균 3% 증가

기존 기술과 결합

옥타곤 후 분석을 통한 허위 경보 제거

- 무인비행체 소프트웨어에 특화된 분석기에 사용된 방법
- 값이 싼 인터벌 분석 후 경보가 난 지점에 선택적으로 값이 비싼 옥타곤 분석을 적용
- 성능을 위해 관계를 추적할 변수를 줄이기 위해 경보와 관련된 변수들만 분석

두 가지 기술을 함께 적용한 결과

프로그램	라인 수	본래 경보 수	대표 경보 구하기 후	옥타곤 분석 후	허위 경보 제거 비율
BIND-1	931	46	38	34	9% → 26%
BIND-2	1,095	42	40	38	5% → 10%
BIND-3	380	2	2	2	0%
BIND-4	645	21	8	5	5% → 76%
FTP-1	503	2	2	2	0%
FTP-2	744	43	33	33	0% → 23%
FTP-3	689	27	11	11	59% → 59%
TOTAL	4987	183	134	125	17% → 32%

- 허위 경보 기존보다 15% 더 제거
- 옥타곤 후 분석 시간 20% 감소(42.61초 → 33.93초)

향후 과제 및 결론

경보 간 종속 관계를 안전하게 계산하는 틀

- 믿을 수 있음
- 추가 비용이 낮은 편
- 기존의 기술과 잘 어울림

