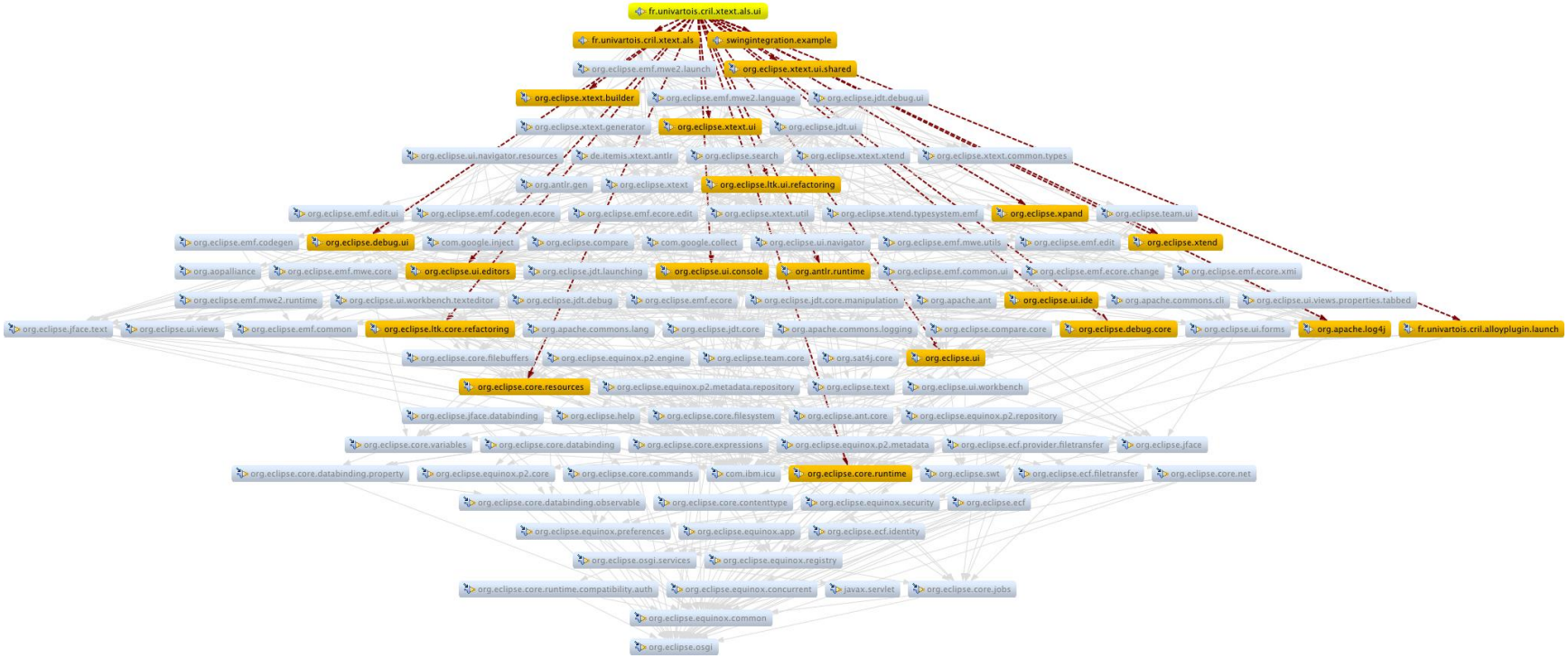
SAT/SMT 다양한 응용들

이승중
ROPAS
2011.06.26

사례1. 이클립스 의존성 분석

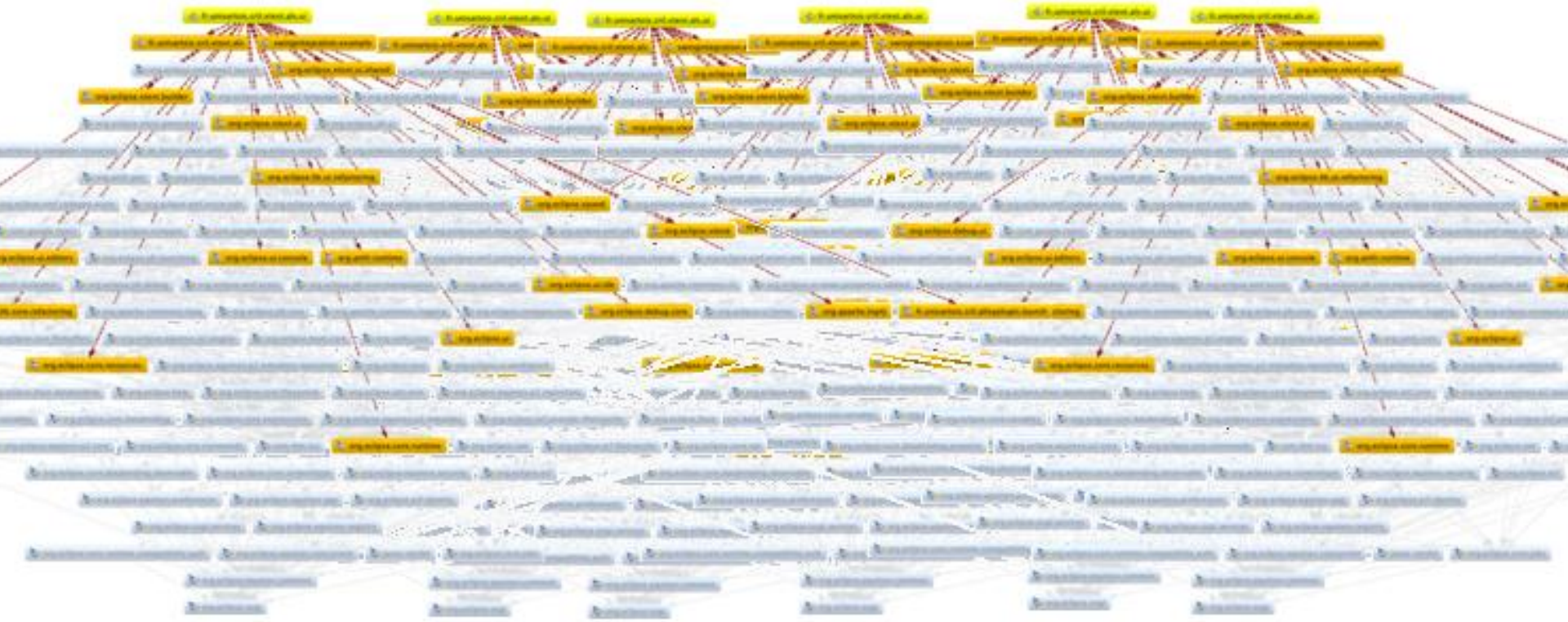
이클립스 플러그인 의존성

Alloy 4 Eclipse 플러그인의 의존성 (20개의 직접 의존성, 총 108개)



이클립스 플러그인 의존성

여러 개의 플러그인을 동시에 깔 때 문제 발생



이클립스 플러그인

- 3000개 이상의 플러그인

- 각 플러그인에는
  - 반드시 같이 깔아야 되는 플러그인 목록
  - 같이 깔면 안 되는 플러그인 목록

이클립스 플러그인

- 플러그인 집합

```
{ smt1, pico1, pico2, crypto1, crypto2 lp_solve1, glpk1}
```

- 의존성

```
smt1 -> {
   {pico1, pico2, crypto1, crypto2},
   {lp_solve1, glpk1}
}
```

- 충돌

pico1 -> {pico2, crypto1, crypto2}

pico2 -> {crypto1, crypto2}

SAT 문제로 바꾸기

- 의존성은 clause로 표현 가능

```
smt1 -> {
  {pico1, pico2, crypto1, crypto2},
  {lp_solve1, glpk1}
}
```

$\neg smt1 \vee pico1 \vee pico2 \vee crypto1 \vee crypto2$

$\neg smt1 \vee lp\_solve1 \vee glpk1$

SAT 문제로 바꾸기

- 충돌은 binary clause로 표현 가능

```
pico1 -> {pico2, crypto1, crypto2}

¬pico1 ∨ ¬pico2
¬pico1 ∨ ¬crypto1
¬pico1 ∨ ¬crypto2
```

더 나아가서

- 설치된 패키지 숫자를 최소로 만드는 것
- 설치된 패키지의 용량을 줄이기
- 가장 최신의 패키지들을 주로 설치하기

- 리눅스(데비안, 3만개 패키지)
- Maven(20만개 라이브러리)의 패키지 분석

사례2. 테스트 입력 자동으로 만들기

테스트 입력 자동으로 만들기

- 수동으로 넣기
  - 모든 경우를 다 생각하기 힘듦
  - 같은 상황을 만드는 불필요한 입력이 있을 수 있음
- 목표
  - 프로그램에서 가능한 모든 경우를 다 포함할 수 있게

테스트 입력 자동으로 만들기

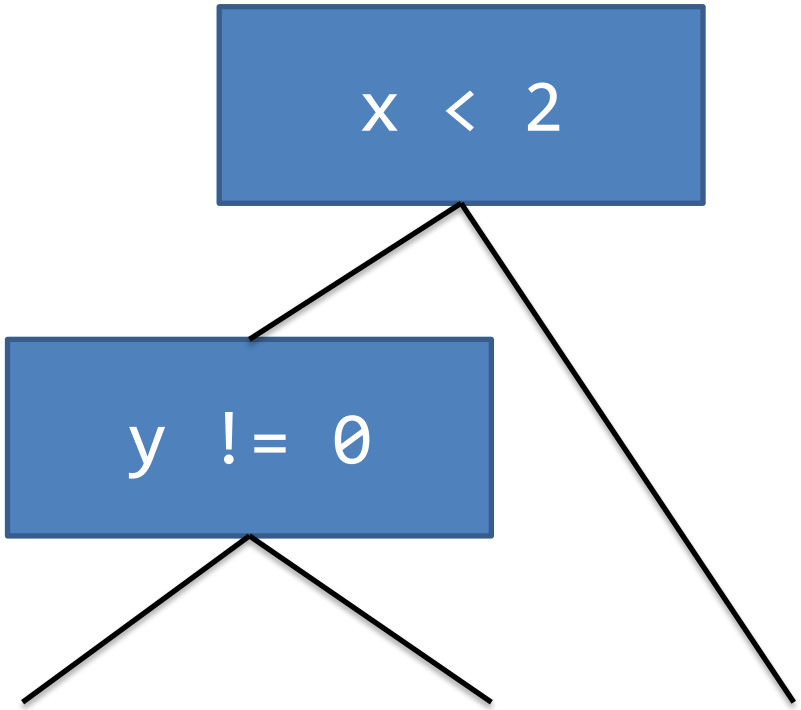
- 방법
  - 중첩된 조건식을 SMT의 입력으로 만듦
  - SMT solver로 풀리면 그 지점으로 접근할 수 있는 입력 예제가 만들어짐

테스트 입력

```
if (x < 2)
{
    if (y != 0)
      assert(false);
    else
      printf (“OK”);
}
else
    printf (“OK”);
```

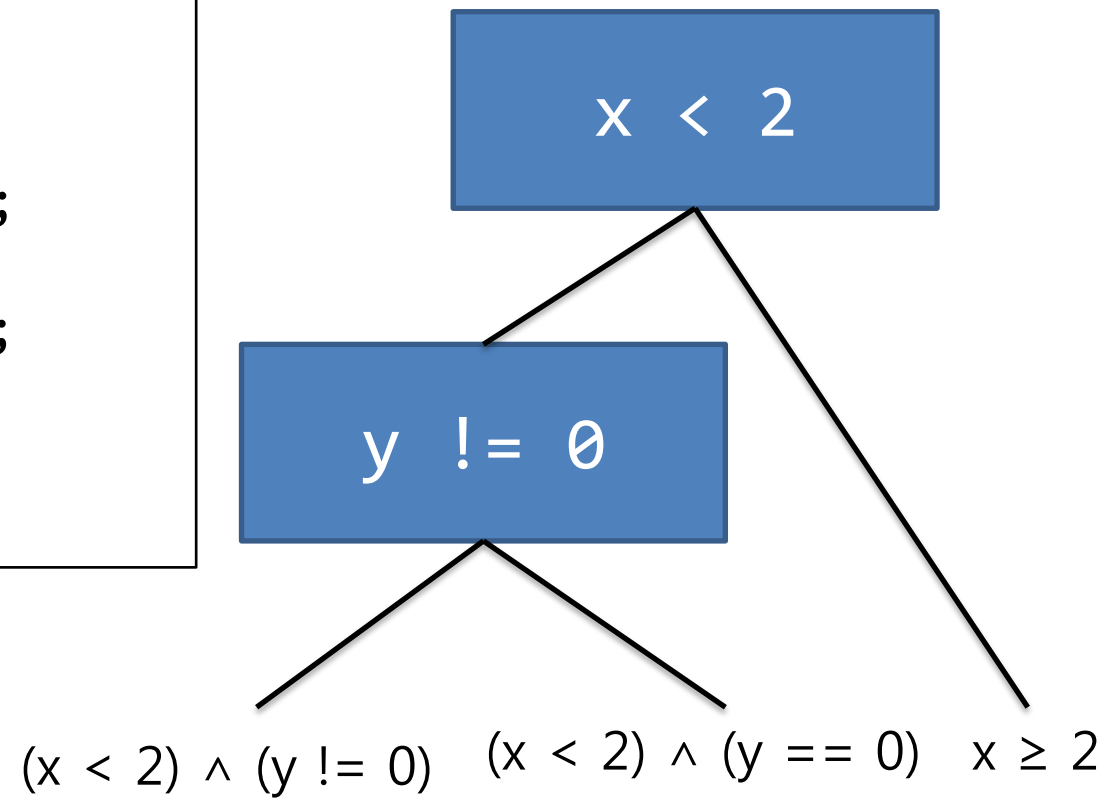
테스트 입력

```
if (x < 2)
{
    if (y != 0)
      assert(false);
    else
      printf (“OK”);
}
else
    printf (“OK”);
```



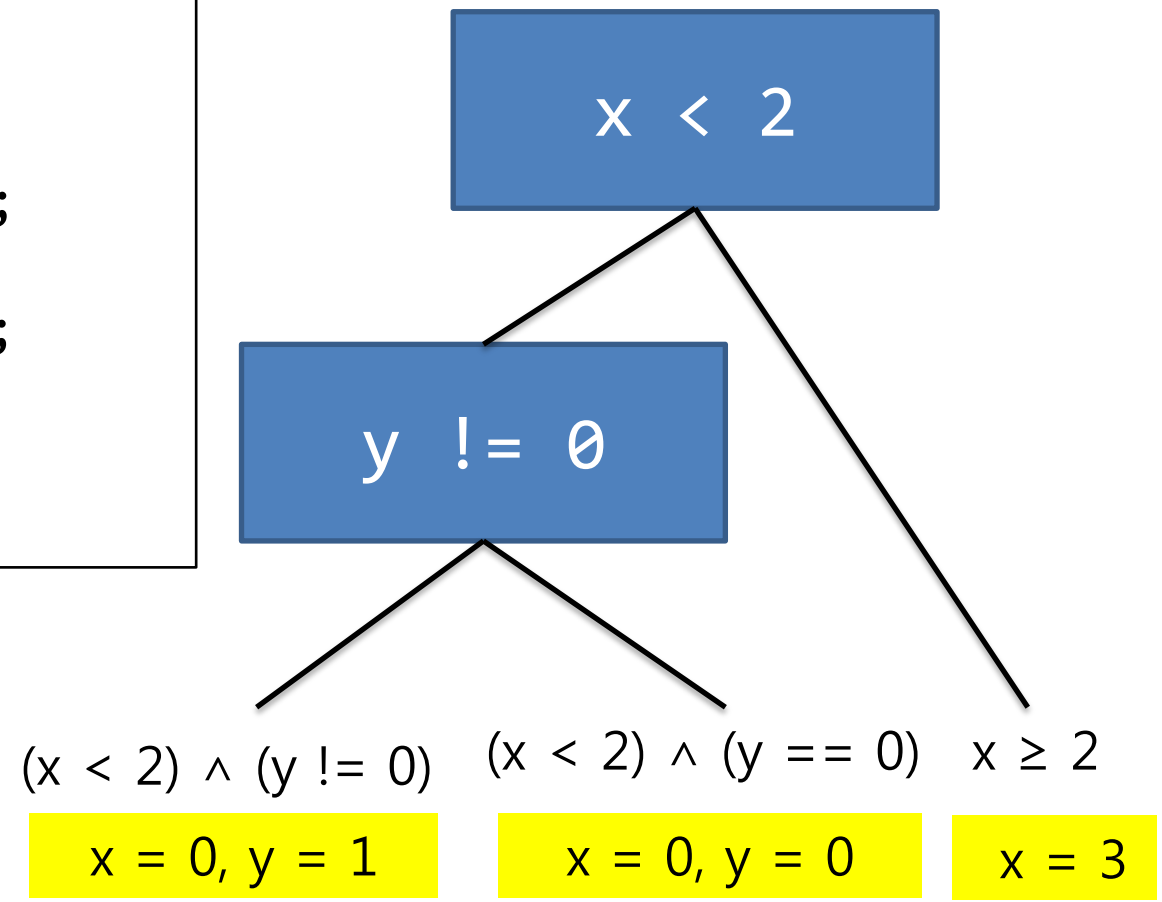
테스트 입력

```
if (x < 2)
{
    if (y != 0)
      assert(false);
    else
      printf (“OK”);
}
else
    printf (“OK”);
```



테스트 입력

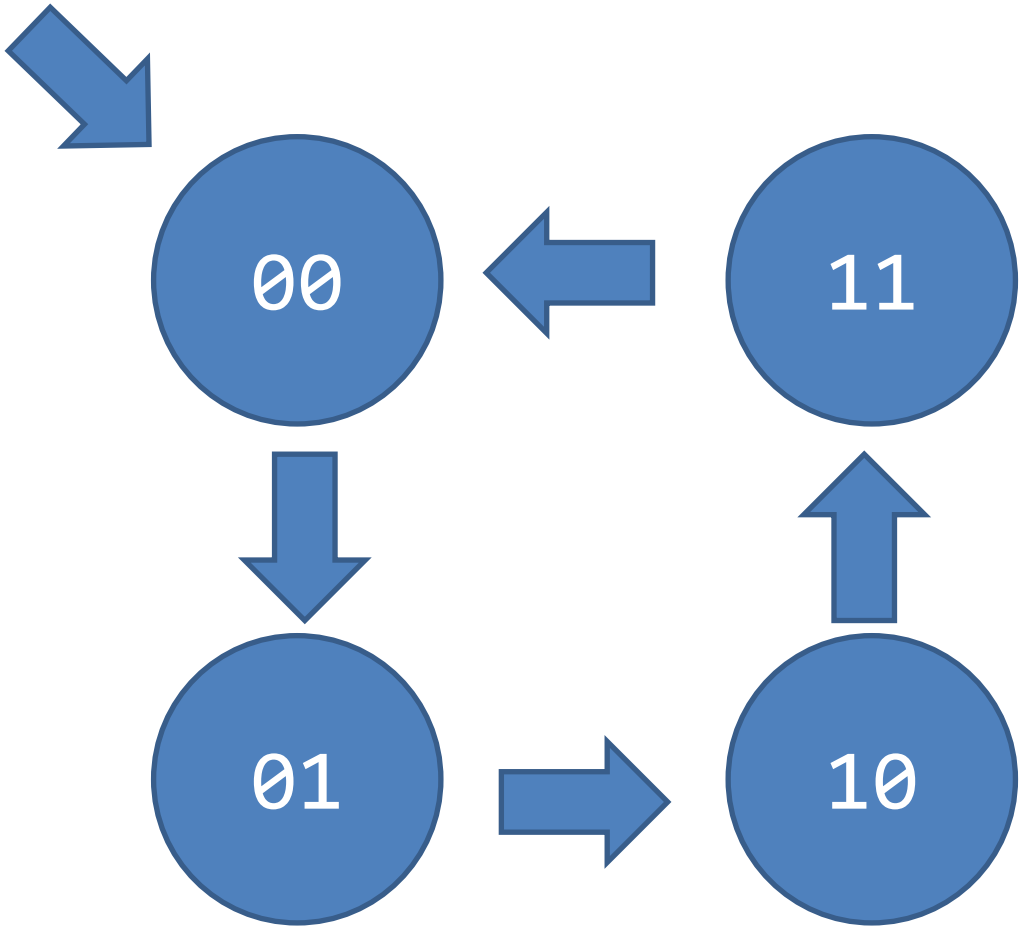
```
if (x < 2)
{
    if (y != 0)
      assert(false);
    else
      printf (“OK”);
}
else
    printf (“OK”);
```



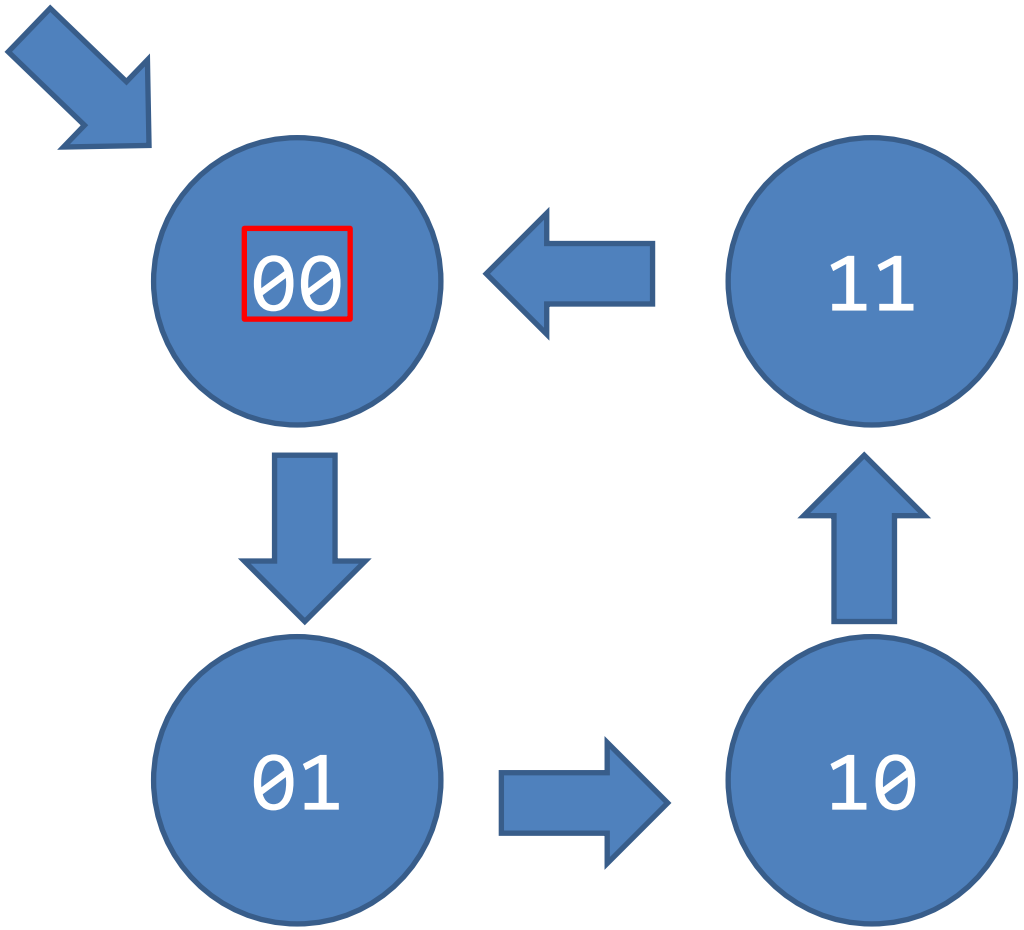
사례3. Bounded Model Checking

Bounded Model Checking

2비트 카운터의 상태 다이어그램



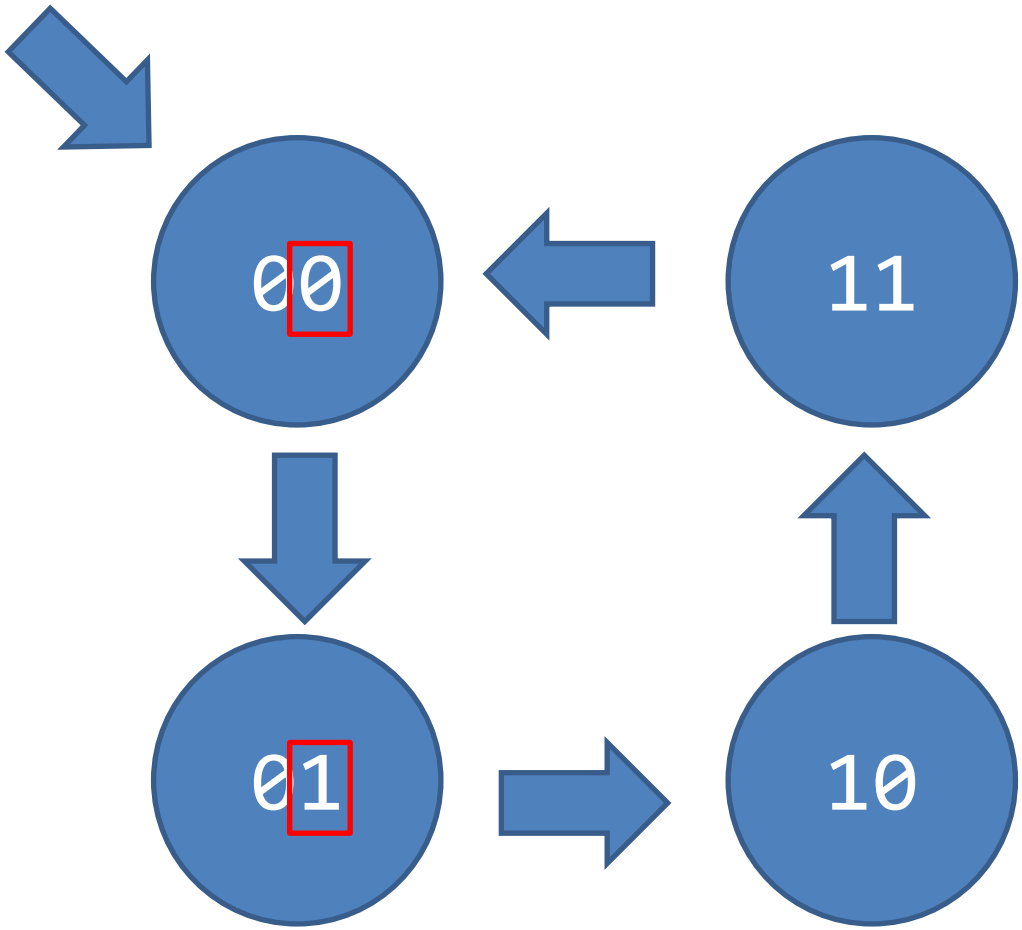
Bounded Model Checking



초기 조건 I

$\neg s[1] \wedge \neg s[0]$

Bounded Model Checking



초기 조건 I

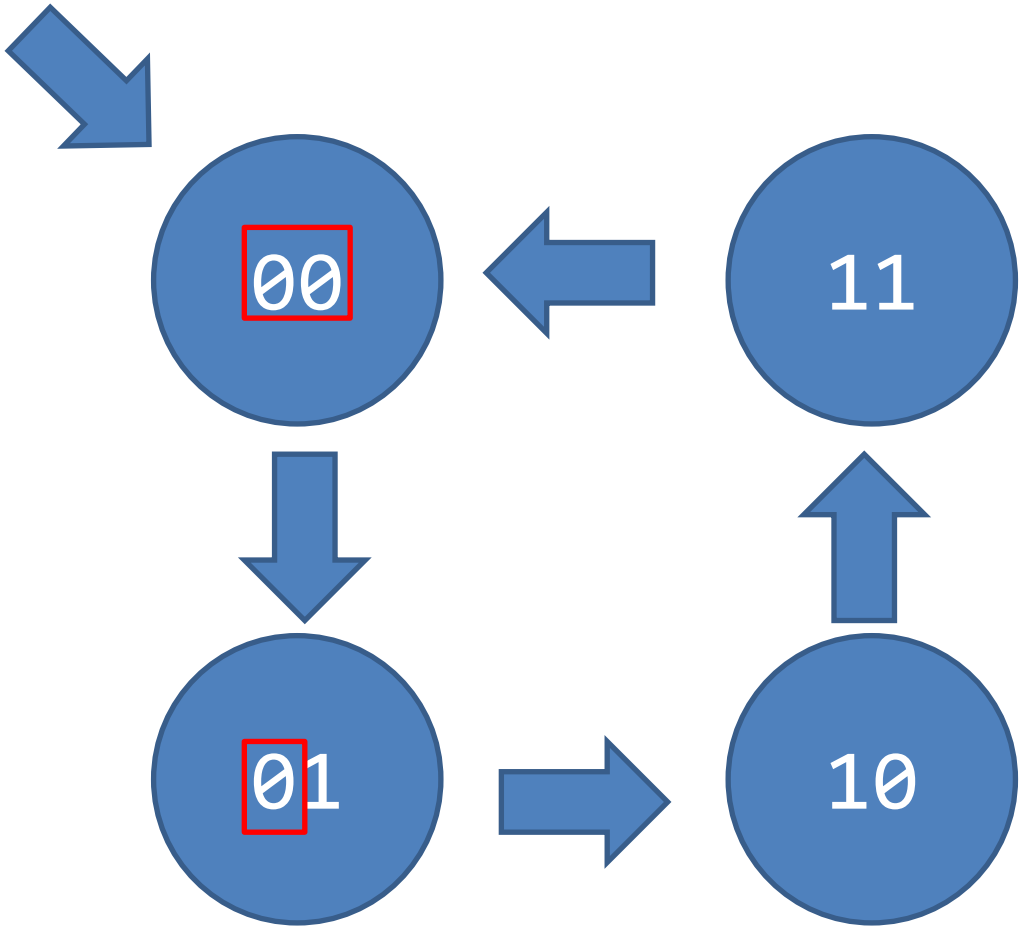
$\neg s[1] \wedge \neg s[0]$

전이 조건 T(s, s')

$(s'[0] = \neg s[0]) \wedge$

$s'[1] = (s[1] \neq s[0])$

Bounded Model Checking



초기 조건 I

$\neg s[1] \wedge \neg s[0]$

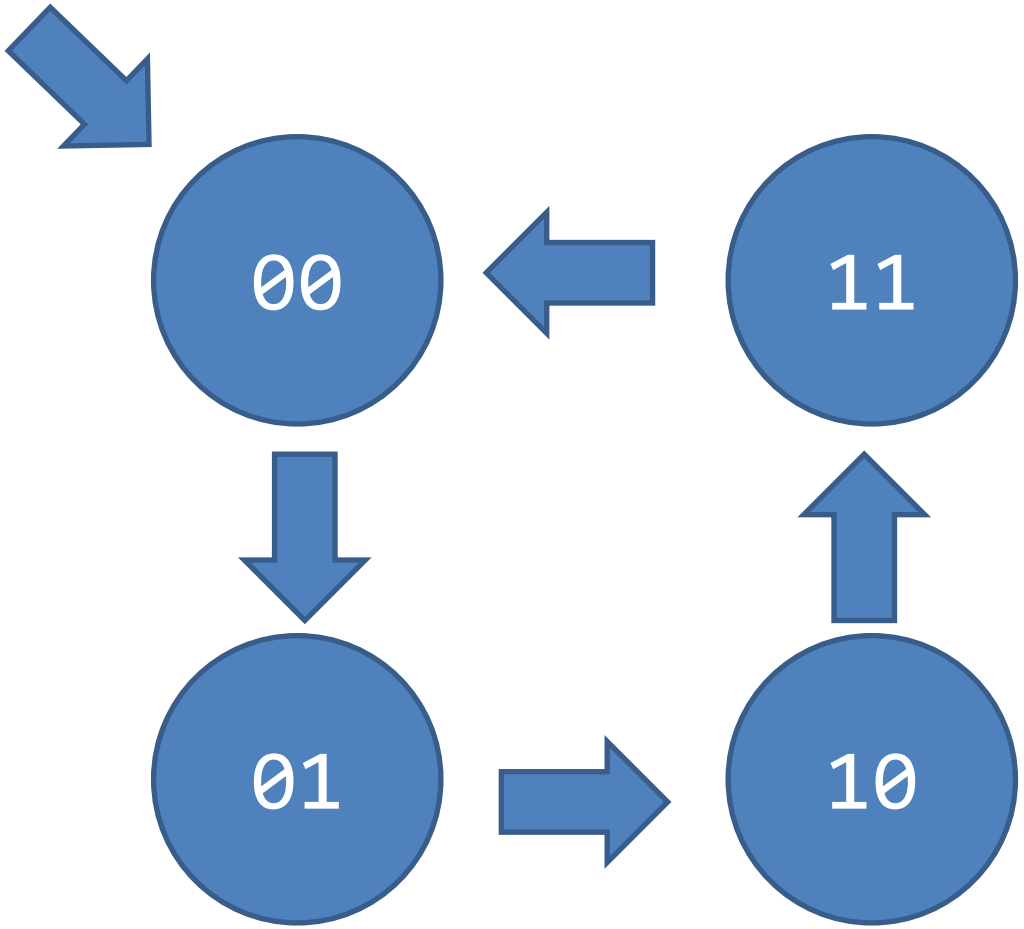
전이 조건 T(s, s')

$(s'[0] = \neg s[0]) \wedge$

$s'[1] = (s[1] \neq s[0])$

Bounded Model Checking

모든 경우에 조건P를 만족하는가



초기 조건 I

$\neg s[1] \wedge \neg s[0]$

전이 조건 T(s, s')

$(s'[0] = \neg s[0]) \wedge$

$s'[1] = (s[1] \neq s[0])$

확인하고 싶은 조건 P

$\neg s[1] \wedge \neg s[0]$

(L, R 둘중에 하나는 0이어야 한다)

Bounded Model Checking

- 안전성 검증
  - k번 전이되었을 때 조건을 만족하는 지 확인

$$I \wedge T(s_0, s_1) \wedge T(s_1, s_2) \wedge \ldots \wedge T(s_{k-1}, s_k) \wedge P(s_0) \wedge P(s_1) \wedge \ldots \wedge P(s_k)$$

참고

- Eclipse
  https://wikis.mit.edu/confluence/display/satsmtschool11/Sat4j
- Automated Testcase Generating
  https://wikis.mit.edu/confluence/display/satsmtschool11/Klee
- Bounded Model Checking
  https://wikis.mit.edu/confluence/display/satsmtschool11/SatFormalVerification