

반도체 생산 제어 코드를 고칠 때 영향 지점의 의미를 더 잘 보이기

SMT Solver 응용

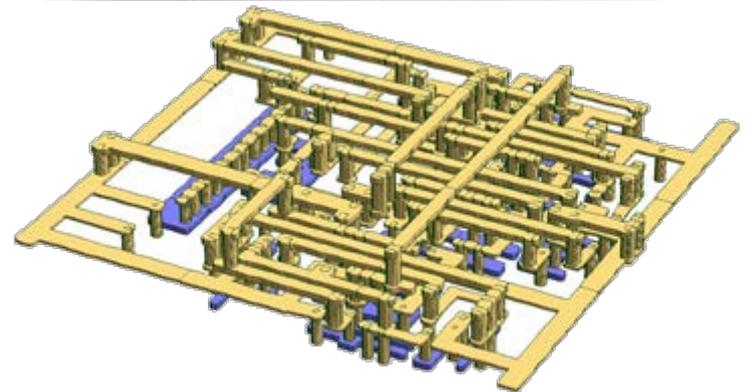
12th ROSAEC 번개발표
서울대학교 프로그래밍연구실

김용기

2014.01.27

반도체 생산 제어 시스템

- 최적 생산이 목적
 - 비싼 반도체 설비를
 - 복잡한 제조 공정에 맞게
 - 효율적으로 운영
- 비즈니스 규칙이 핵심
 - 주어진 기준값에 대해
 - 어떤 조건에서
 - 어떻게 행동할지를 정의



비즈니스 규칙 코드를 고치기란 쉽지 않음...

- PL/SQL 레거시 코드
 - 이십년 가까운 유산(빚?)
 - 수많은 IF/THEN/ELSE
 - 폭탄 제거 같은 위험성
- 전문가 수작업에 의존
 - 코드 변경지점의 영향을
 - 눈으로 / 감으로
 - 많은 노력과 비용
 - 그럼에도 불안



고쳤을 때 영향을 자동으로 찾는 분석기가 필요

- 전문가 대신...
 - PL/SQL 코드를 고쳤을 때
 - 영향 받는 (예상) 지점을
 - 자동으로 찾아주는.
 - 그래서, 코드 리뷰 할 곳을
 - 빠짐없이 알려주는
- 그래서,
우리가 만든 분석기 →
(+ 사용성 개선 된 버전)

The image shows a side-by-side comparison of PL/SQL code. The left pane shows the original code, and the right pane shows the modified code. A yellow highlight in the original code marks a conditional statement: `if {"tag":13,"var":"Y","val":1,"cond":{"val":0}} then`. Below the code, a control flow graph (CFG) visualizes the execution paths. The graph starts with a node 'Field:11 cond: 1' which branches to 'F:13-12 eq_v(1,1)'. From there, it proceeds through a series of nodes: 'F:14-13 eq_v(phi, phi)', 'F:14-13 eq_v(y, y)', 'F:15-14 eq_v(x, x)', 'G:23-22 eq_v(x, x)', and finally 'G:24-23 eq_v(sql, sql)'. A node 'Finew:10 cond: 0' also branches into the main flow at the 'F:13-12' node.

사용자들은 여전히 어려워 함... 해결 방법은?

- “어떤 영향” 인지?
 - 심볼릭 실행 결과로는
 - 파악이 힘들다.
- 실행하지 않고
- 알아낼 수 없을까?

```
Var RESERVATION;1/new/1.sql:MAIN:1226:
  Ins (
    Sym (Var RESERVATION;None, ),
    Sym (Var RESERVETYPE;None, ),
    Sym (Var LOTID;None, ),
    Sym (Var EQPID;None, ),
    Sym (Var PROCESSID;None, ),
    Sym (Var STEPSEQ;None, ),
    Sym (Var UPDATETIME;None, ),
    Sym (Var COMMENT_;None, ),
    , Con "EQPFIX",
    Sym (Fld (Var V_W_ASIMAXWIPR;None, SUBLOTID);1/new/1.sql:MAIN:783, ),
    Sym (Fld (Var V_W_ASIMAXWIPR;None, STATE);None, ),
    Sym (Fld (Var V_W_ASIMAXWIPR;None, PROCESSID);1/new/1.sql:MAIN:562, ),
    Sym (Fld (Var V_W_ASIMAXWIPR;None, STEPSEQ);1/new/1.sql:MAIN:562, ),
    Sym (Var SYSDATE;None, ),
    Con "BACKUPSAMPLE",
  )
Val Con "";
Val Grd Lop
(And,
  Rop (Eq,
    Sym (Fld (Var V_W_ASIMAXWIPR;None, TRACKINGTYPE);None, ),
    Con "BACKUPSAMPLE"),
  Rop (Neq,
    Fun (.ret, NVL,
      Sym (Fld (Var V_W_ASIMAXWIPR;None, STATE);None, ),
      Con "-"),
    ),
    Con "-")
);
...
```

기회: “비즈니스” 규칙 프로그램의 특수성

- 규칙: DB 컬럼 값들의 조건식
- DB 컬럼은 일종의 “값 도메인”
- 대부분 열거형 또는 범위형
 - 열거형 : 제품 타입 등
 - 범위형 : 온도, 횟수 등
 - 거의 모든 경우 정수형

- 예: EQUIPMENT (설비) 테이블의 컬럼 값 조건

EQPID	SUBAREA	EQPTYPE	RUNCOUNT

- EQPID: E% (앞자리가 생산라인)
 - EQPID LIKE “E%” & LINEID=13
- SUBAREA: enum
 - PHOTO, ETCH, IMP, ...
- EQPTYPE: P or E
- SUBAREA: PHOTO인 경우
 - RUNCOUNT < 10

목표: 고친 코드의 영향 의미를 더 잘 보이기

- 컬럼 값의 스펙을 잘 설계
 - 컬럼값 조건식을 체계화
- 스펙으로 심볼릭 값 풀기
 - SMT Solver 이용
- 영향지점의 값 요약/응용
 - 변경 전/후 심볼릭 값의 의미를 쉽게 파악 가능
 - 경계값 테스트 등 테스트 케이스 자동 생성에 응용

```
create package body p as
  procedure f (x IN OUT number)
  is
    y number;
  begin
    y := x;
    if 0 then
      y := 1;
    end if;
    x := y;
  end f;

  function g
  is
    x number;
  begin
    x := 0;
    f(x);
  end g;
end p;
```

변경 지점

```
create package body p as
  procedure f (x IN OUT number)
  is
    y number;
  begin
    y := x;
    if 1 then
      y := 1;
    end if;
    y;
  end f;

  function g () return number
  is
    x number;
  begin
    x := 0;
    f(x);
  end g;
end p;
```

영향 지점

Given Spec: $0 \leq t.x \leq 100$
Before: $0 \leq t.x \leq 100$
After: $t.x = 1$

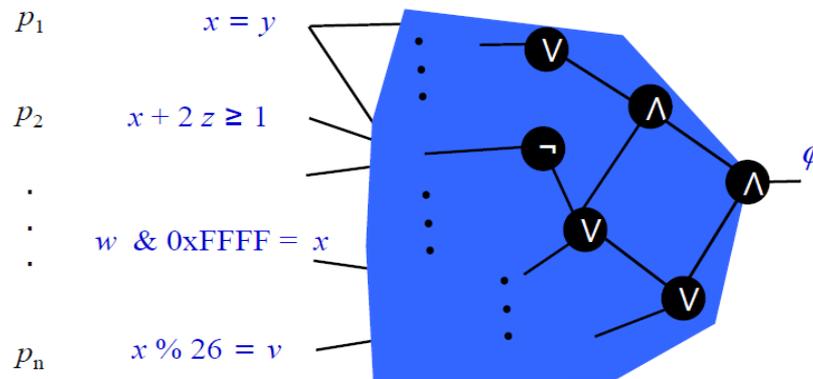
```
insert into t values (x);
return x;
```

SMT Solver

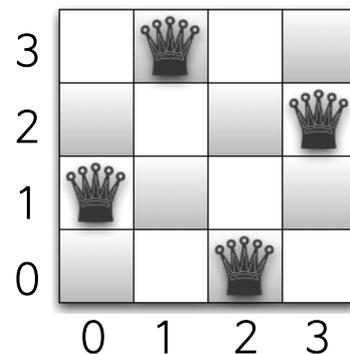
- SAT solver의 확장
- 주어진 (부)등식, 방정식을 만족시키는 모델 찾기
 - SMT-LIB.org
 - YICES, Z3, ALT-ERGO, ...
- 예: 4 Queen 문제 풀기

(참고: <https://kldp.org/node/106972>)

```
(define-type nat4 (subtype (n::nat) (< n 4)))
(define abs::(-> int nat) (lambda (x::int)
  (if (>= x 0) x (- 0 x))))
(define q::(-> nat4 nat4)
  (assert (forall (y::nat4)
    (exists (x::nat4) (= y (q x))))))
  (assert (/= (abs (- (q 0) (q 1))) 1))
  (assert (/= (abs (- (q 0) (q 2))) 2))
  (assert (/= (abs (- (q 0) (q 3))) 3))
  ...
```



참고: C. Barrett & S. A. Seshia, ICCAD 2009 Tutorial



$q(0) = 1$
 $q(1) = 3$
 $q(2) = 0$
 $q(3) = 2$

결론 및 기대 효과

- 비즈니스 규칙 시스템의 특수성을 이용하면 SMT Solver로 영향 지점의 의미를 더 잘 분석 가능
- 비슷한 산업분야에 일반적으로 응용 가능
- (잘되면) 삼성 반도체에 수십억 테스트 비용 절감 기대

감사합니다. 😊