
Esterel에서 회로중복사용 문제의 모듈단위 분석

윤정한 황준형 한태숙
한국과학기술원 프로그래밍언어연구실
2010년 10월 21일

차례

- Esterel 언어 소개
- Schizophrenia 문제 소개
- Esterel 언어의 Control Flow Graph 생성
- Schizophrenia 모듈 단위 분석 기법 제안

Esterel 언어

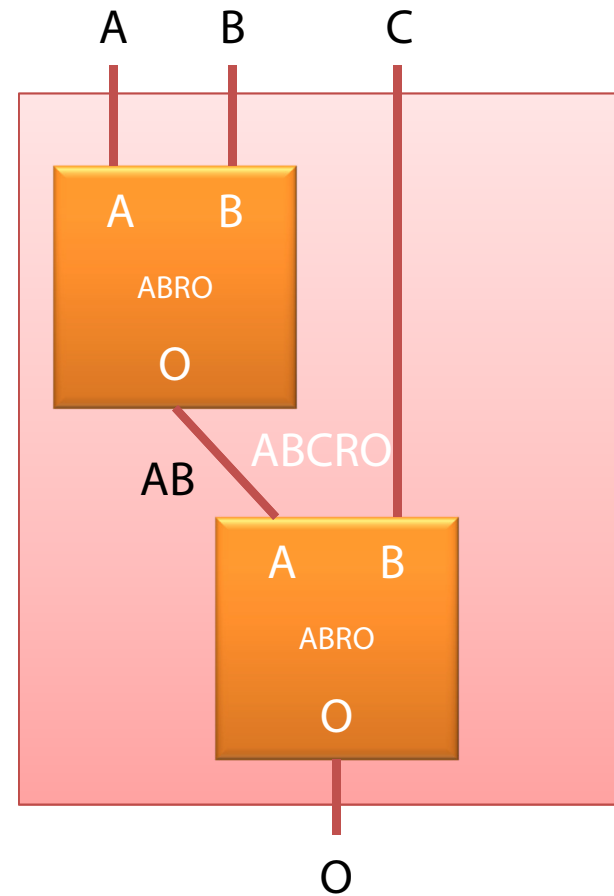
- 동기식 synchronous
 - 실행을 단위 시간으로 끊어서 생각
- 반응형 reactive
 - 외부에서 주어진 자극에 대해 정해진 시간 안에 반응
 - 대부분의 반응형 시스템은 결정적 deterministic
- 명령형 imperative
 - 시스템의 실행은 상태 변화를 통해 이루어짐

Esterel 프로그램의 구조

- Esterel 프로그램은 모듈`module`로 구성
- 모듈 선언은 프로그램 문장과 인터페이스 신호 선언으로 구성
- 한 모듈에서 다른 모듈을 사용하려면 `run` 문장으로 모듈을 호출하고 인터페이스 신호를 연결

Esterel 언어로 작성한 프로그램의 예

```
module ABCRO:  
input A, B, C, R;  
output O;  
signal AB in  
    run ABRO [signal AB / O]  
||  
    run ABRO [signal AB / A, C / B]  
end signal  
end module
```



회로중복사용 문제

- 회로중복사용schizophrenia 문제
 - 한 문장이 한 단위시간에 두 번 이상 수행될 때 발생
 - 문제가 발생할 수 있는 회로를 복제해서 치료

회로중복사용 사례

```
loop
  signal S in
    present S then emit 0 else nothing end;
    pause;
    emit S
  end
end
```

| 회로의 종류 | 첫 단위시간의 0 | 두 번째 단위시간의 0 |
|--------|-----------|--------------|
| 올바른 회로 | 꺼짐 | 꺼짐 |
| 잘못된 회로 | 꺼짐 | 켜짐 |

회로중복사용 치료

```
loop
  signal S in
    present S then emit 0 else nothing end;
    pause;
    emit S
  end;
  signal S in
    present S then emit 0 else nothing end;
    pause;
    emit S
  end
end
```


회로중복사용 문제의 종류

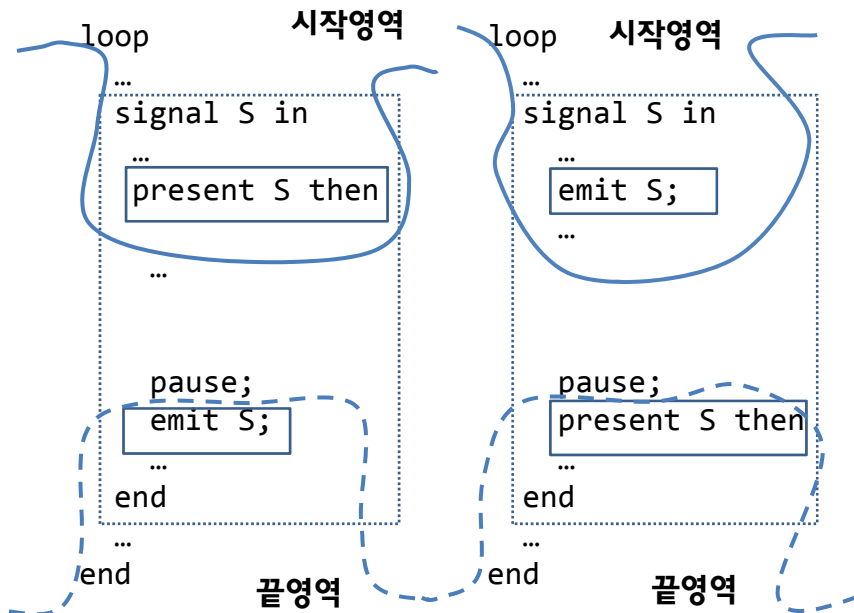
- 지역신호 중복사용 schizophrenic signal declaration
- 예외 중복사용 schizophrenic trap statement
- 출력문 중복출력 schizophrenic emit statement
- 동기장치 중복사용 schizophrenic trap synchronizer

회로중복사용 문제의 탐지

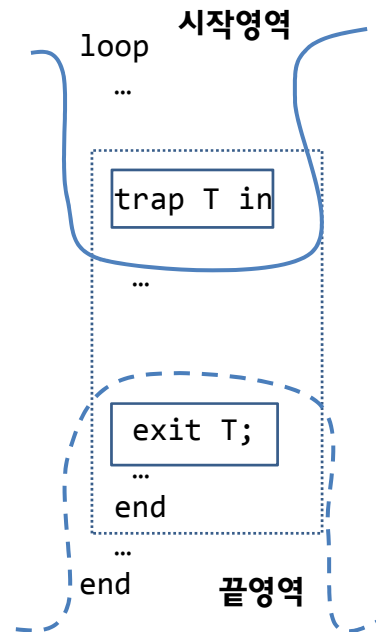
- CFG(Control Flow Graph) 상에서 시작영역(first-surface)과 끝 영역(last-surface)을 이용해서 탐지

회로중복사용 문제의 탐지

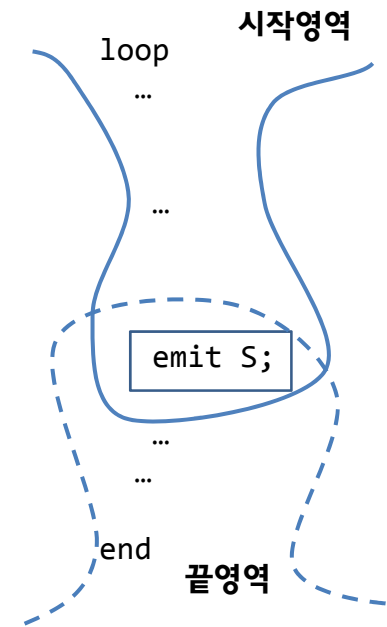
지역신호 중복사용



예외 중복사용



출력문 중복사용



관련 논문

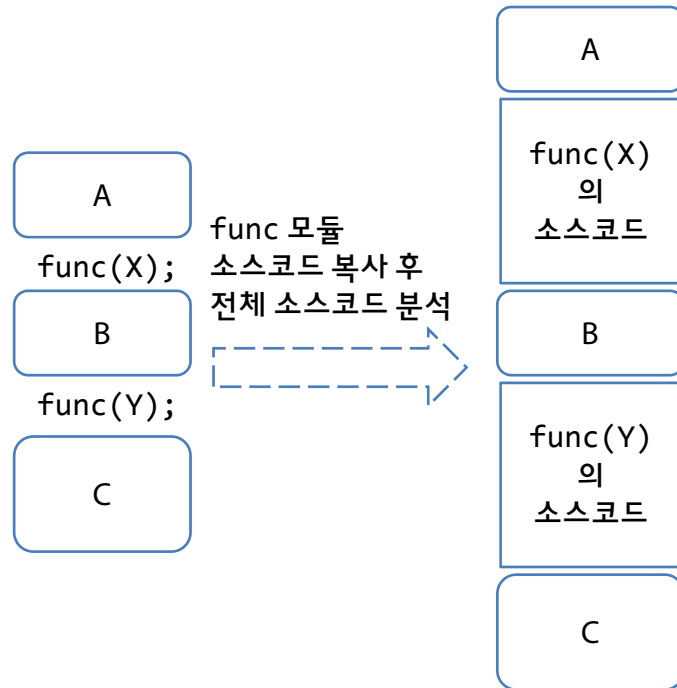
- CFG 생성
 - C. Kim, J. Yun, S. Kim, T. Han, and K. Choe, "Over-approximated control flow graph construction on pure Esterel", *IEICE Transactions on Information and Systems*, 2010
- 회로중복사용 문제
 - J. Yun, C. Kim, S. Seo, T. Han, and K. Choe, "Refining schizophrenia via graph reachability in Esterel", *7th ACM-IEEE International Conference on Formal Methods and Models for Codesign*, 2009
 - J. Yun, C. Kim, S. Kim, K. Choe, and T. Han, "New schizophrenia patterns on Esterel caused by control/data signals", *Journal of KIISE: Software and Application*, 2010
 - J. Yun, C. Kim, S. Kim, and T. Han, "Reducing False Alarms in Schizophrenic Parallel Synchronizer Detection for Esterel", *Journal of KIISE: Software and Application*, 2010

모듈단위 분석의 필요성

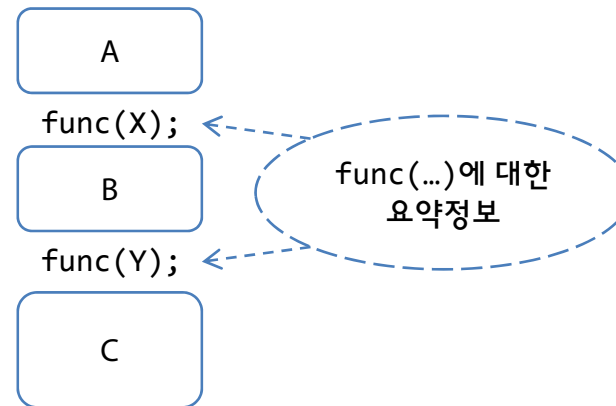
- 기존 분석 방법은 각 모듈의 내용을 복사한 다음 분석을 수행
 - 동일한 분석을 중복 수행
- 소스 코드 없이 분석해야 할 때가 있음
 - 컴파일을 통해 생성한 회로만 있을 때
 - 다른 사람이 만든 회로를 이용해야 할 때

모듈단위 분석

기존 분석 방법



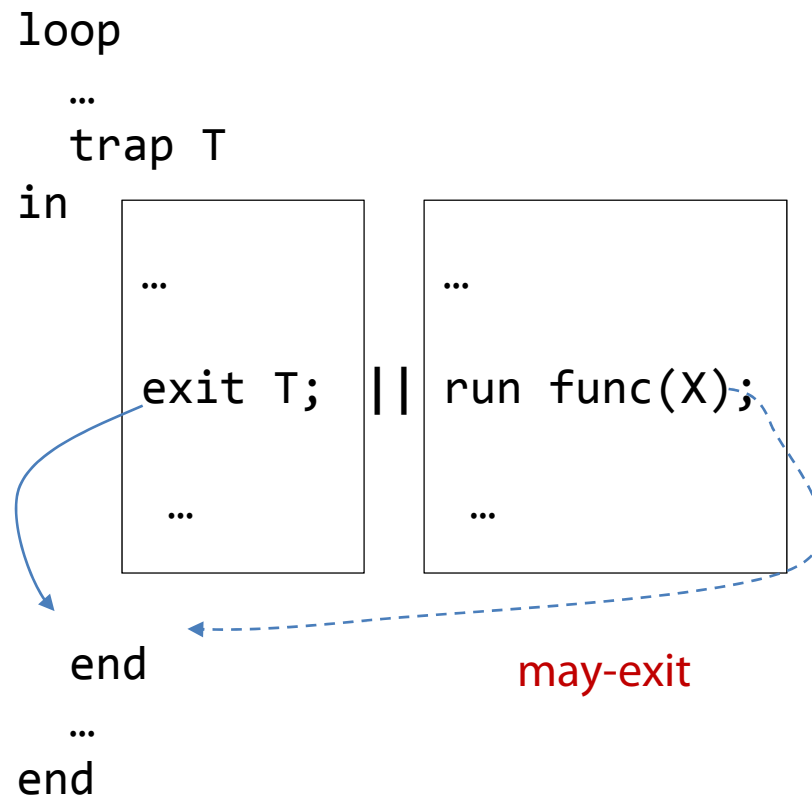
모듈단위 분석



모듈단위 분석을 위해 요약하는 정보

- 시작영역
 - emit, test, trap_start, module_end 노드
- 끝영역
 - emit, test, exit, module_start 노드
- 코드 전체
 - emit, test, exit 노드

병렬문에서의 예외 발생



모듈단위 분석 방법

1. CFG 생성

2. 시작영역, 끝영역 계산

- normal 모듈과 may-exit 모듈 구분

3. 시작영역, 끝영역 요약

- 시작영역 내에 모듈 노드가 있으면 시작영역 정보를 포함시킴
- 끝영역 내에 normal 모듈 노드가 있으면 끝영역 정보를 포함시킴
- 끝영역 내에 may-exit 모듈 노드가 있으면 전체 정보를 포함시킴

4. 회로중복사용 문제 검출

향후 과제

- 실제로 구현하여 예상한 결과를 얻는지를 확인
 - 중복 분석을 줄여 큰 프로그램에 대한 분석 속도 향상에 기여할 것으로 기대
 - 모듈 단위 컴파일에도 적용 가능
- 다양한 정적 분석 기법을 모듈 단위로 적용

고맙습니다

- 질문 부탁드립니다