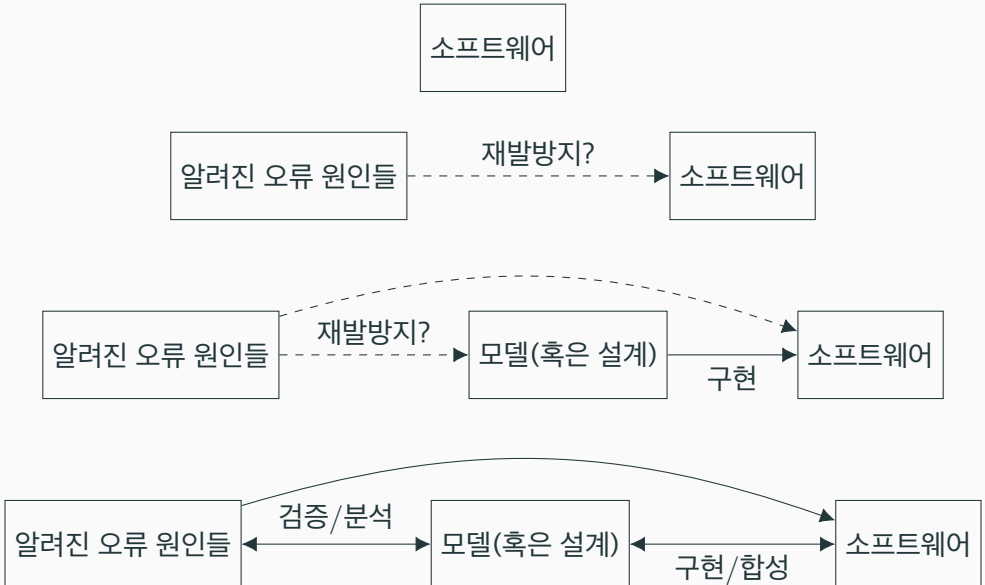
3그룹 연구목표 소개

배경민

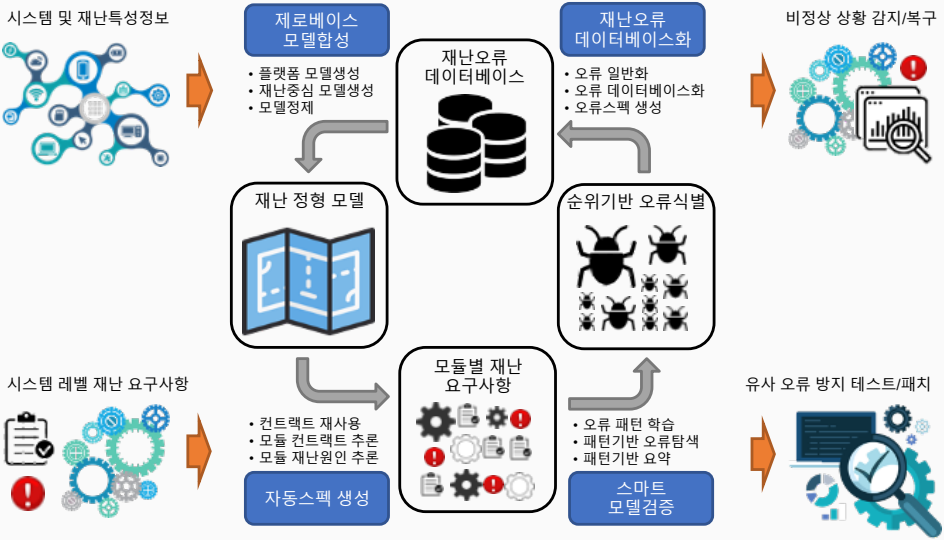
2024년 1월 30일

POSTECH 컴퓨터공학과

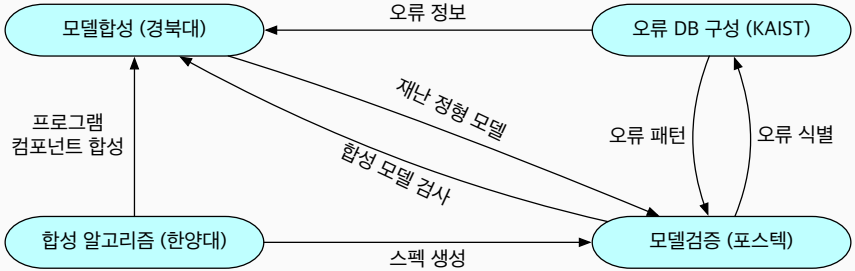
그룹3 연구목표: 소프트웨어재난 재발방지



그룹3 연구내용 개요


시스템 및 재난특성정보
제로베이스 모델합성
• 플랫폼 모델생성
• 재난중심 모델생성
• 모델정제
재난오류 데이터베이스
재난오류 데이터베이스화
• 오류 일반화
• 오류 데이터베이스화
• 오류스펙 생성
비정상 상황 감지/복구
재난 정형 모델
순위기반 오류식별
시스템 레벨 재난 요구사항
• 컨트랙트 재사용
• 모듈 컨트랙트 추론
• 모듈 재난원인 추론
자동스펙 생성
모듈별 재난 요구사항
• 오류 패턴 학습
• 패턴기반 오류탐색
• 패턴기반 요약
스마트 모델검증
유사 오류 방지 테스트/패치

연구실 별 연구주제 및 인터페이스



- 모델합성 ⇔ 모델검증
- 합성 알고리즘 ⇔ 모델합성, 모델검증
- 오류 데이터베이스 ⇔ 모델검증, 모델합성

POSTECH 모델검증 연구 소개

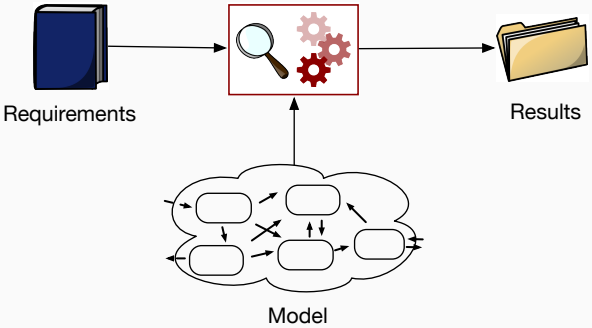
배경민

2024년 1월 30일

POSTECH 컴퓨터공학과

모델검증 (Model Checking)

- 시스템의 오류를 자동으로 찾는 기술
  - 소프트웨어/하드웨어 디자인,
    프로토콜 디자인, 소스 코드, …
  - 다양한 모델검증 도구 존재
- 특징
  - 시스템의 모든 가능한 상태를
    확인하여 “오류 없음” 증명 가능
  - 자동적으로 복잡한 성질을 검증 가능



접근방법: 논리 기반 모델검증

- 모델검증 기법 적용의 장애물
  - Algorithmic challenge: 상태 폭발 문제 (state space explosion)
  - Modeling challenge: 다양한 소프트웨어 시스템의 정형명세
- 논리 기반 모델검증

Model		Logic System		Verification
시스템 명세		수학적 모델		
M	$\Longrightarrow$	$\mathcal{R}_M$	$\Longrightarrow$	모델검증
성질 명세		논리식		알고리즘
$spec$	$\Longrightarrow$	φ_{spec}		

논리 기반 모델검증 도구 예제

- Boolean logic
  - CBMC, NuSMV, ⋯
- Satisfiability modulo theories (SMT)
  - nuXmv, MCMT, ⋯
- Rewriting logic
  - Maude, KEVM, RV-Predict, CafeOBJ, ⋯
- Temporal logic of actions
  - TLA+
- ⋯

- Algorithmic challenge
  - 논리 시스템의 알고리즘 및 최적화 기법 연구: Rewriting logic 및 SMT
  - STAAR: 패턴 기반 모델검증 알고리즘
- Modeling challenge
  - 대상 성질/오류 및 시스템에 최적화된 모델링 및 정형명세 기술 연구
  - STAAR: 모델검증 인터페이스 개발 및 모델검증 응용

패턴을 활용한 모델검증

- 목적: 모델검증 시 패턴에 기반한 상태공간 탐색
  - 오류 패턴 등을 활용하여 (알려진) 오류를 효율적으로 탐색
  - 오류와 관계가 적은 상태공간을 효과적으로 요약
- 필요 기술
  - 패턴의 정의 및 패턴 기반 실행 방법
  - 패턴 공간의 탐색 및 요약
- 연구 방법
  - Rewriting logic: 높은 표현력을 가진 명세 언어로 다양한 모델링 언어의 의미 정의 가능
  - Rewriting logic으로 명세된 시스템의 패턴 기반 모델검증 연구

Rewriting Logic Specification

- State

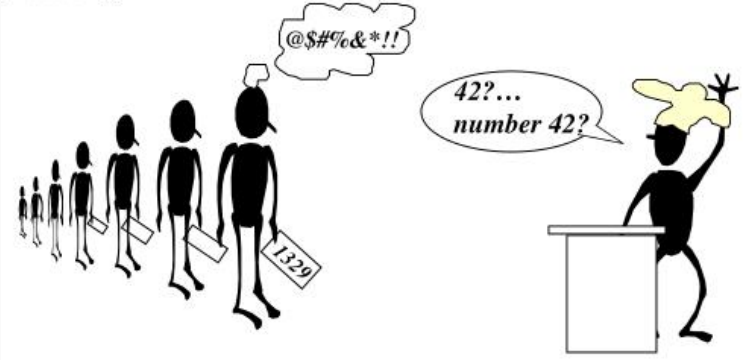
  term t (algebraic data type)

- Transition

  rewrite rule $t \longrightarrow t'$ **if** ψ (patterns t and t', and condition ψ)

- Example
  - By rule $f(N) \longrightarrow f(N+1)$ **if** $N < 10$, term $f(5)$ is rewritten to $f(5+1)$

Example: Lamport's Bakery Algorithm



- Each process receives a ticket number to enter the critical section.
- Process with the smallest ticket number enters the critical section.

Example: Lamport's Bakery Algorithm

- Each state with N processes:

$$n\ ;\ m\ ;\ [i_1, d_1] \ldots [i_N, d_N]$$

  - n: the current number in the bakery's number dispenser
  - m: the number currently served
  - $[i_l, d_l]$: process with id d_l in status $d_l \in \{\texttt{idle}, \texttt{wait}(\mathit{ticket}), \texttt{crit}(\mathit{ticket})\}$

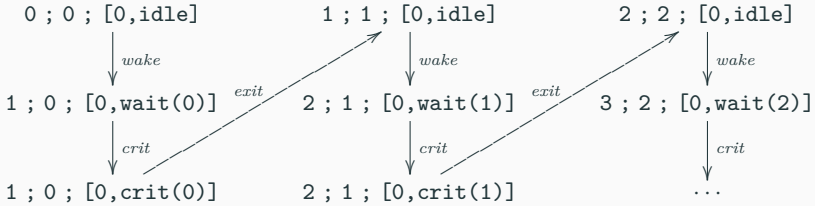
- Rewrite rules (in the Maude syntax)

```
rl [wake]: N ; M ; [K,    idle] PS  =>  N + 1 ; M ; [K,wait(N)] PS .
rl [crit]: N ; M ; [K,wait(M)] PS  =>  N ;     M ; [K,crit(M)] PS .
rl [exit]: N ; M ; [K,crit(M)] PS  =>  N ; M + 1 ; [K,    idle] PS .
```

  - Variables: `N`, `M`, `K` for numbers, and `PS` for process sets

Example: Lamport's Bakery Algorithm

- Rewrite sequences with one process



Symbolic Representation Using Logical Terms

- Symbolic state

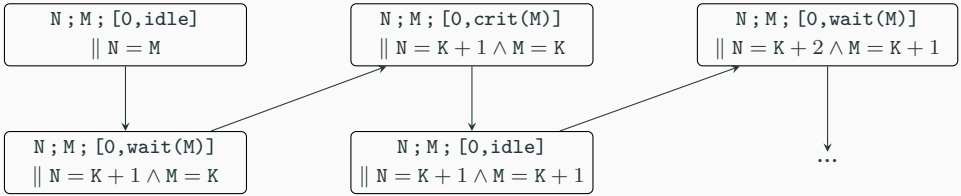
  $t \parallel \phi$ a pattern t with variables constrained by a formula ϕ

  expresses a set of (potentially infinitely many) concrete states

- Example
  - $\mathtt{N}\,;\,\mathtt{M}\,;\,\mathtt{PS} \parallel \mathtt{N} \geq \mathtt{M}$
  - $0\,;\,0\,;\,[0, \mathit{idle}]$, $1\,;\,1\,;\,[0, \mathit{idle}]\,[1, \mathit{idle}]$, $3\,;\,2\,;\,[0, \mathit{idle}]\,[2, \mathit{wait}(0)]$, $\cdots$

Computing Symbolic State Space

- Symbolic transition $\rightsquigarrow$
  - $t \parallel \phi \rightsquigarrow t' \parallel \phi'$ if an instance of $t \parallel \phi$ can be rewritten to t' by some rewrite rule
  - a.k.a. constrained narrowing
- Example: constrained narrowing sequences with one process



Research Topics

- 우선순위 기반 패턴 탐색
  - Heuristic search for rewriting and (constrained) narrowing
  - Learning heuristics from previous model checking attempts
- 패턴 기반 상태 공간 축소
  - State-space reduction using patterns
  - Learning patterns from previous mode checking attempts
- 패턴 기반 정형명세
  - Formal specification using constrained patterns
  - Programming languages, system APIs, ⋯

POSTECH 모델검증 연구

- 모델검증 알고리즘: (패턴을 활용한) 모델검증 알고리즘 연구
  - 모델검증 탐색전략 학습(포스터) (손병호)
  - Maude-SE 도구 개발 및 벤치마크(포스터) (류근열,장혁순)
- 모델검증 인터페이스: 프로그래밍 언어의 의미구조 정형명세
  - PLC 언어의 모델검증(포스터) (이재서)
  - PROMELA 언어 및 AADL 언어의 모델검증 (손병호/이재훈)
- 모델검증 응용: 모델검증 사례연구 및 도구 개발
  - TEE API의 정형명세 및 모델검증(포스터) (류근열,채승현)
  - TLS의 정형명세 및 모델 기반 테스팅(포스터) (이재훈)
- 사이버물리시스템 및 심층신경망 모델검증
  - Signal Temporal Logic 모델검증 연구(포스터) (이지아,류근열)
  - 심층신경망 모델 검증 연구(포스터/포스터) (연주은/채승현)

Thank you!