

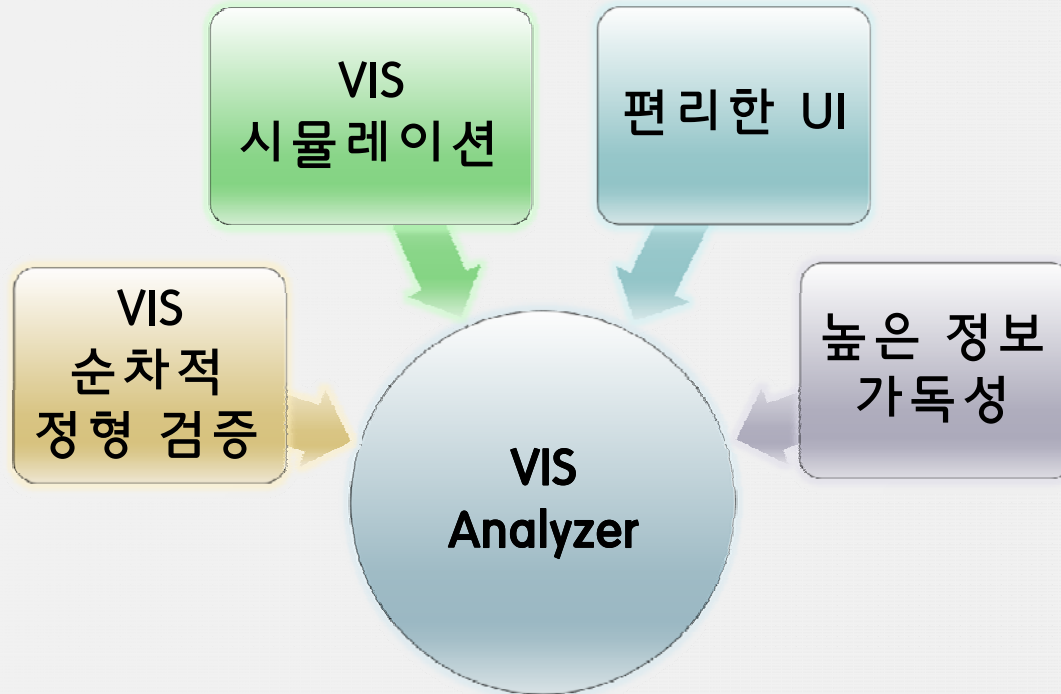
VIS 정형 검증 자동화를 위한 소프트웨어 VIS Analyzer 개발

정세훈^{o,a}, 유준범^b, 차성덕^a

a 고려대학교 컴퓨터학과, b 건국대학교 컴퓨터공학부

gifaranga@yahoo.co.kr, jbyoo@konkuk.ac.kr, scha@korea.ac.kr

1. VIS Analyzer



- I. VIS(Verification Interaction Synthesis) 정형 검증 지원 도구.
- II. VIS 순차적 일치성 검증 결과와 시뮬레이션결과를 조합하여 순차적 검증 결과의 활용성을 높임.
- III. VIS의 콘솔 기반 UI → 그래픽 기반 UI 제공

2. VIS

I. VIS (Verification Interaction Synthesis)

II. 모델 체킹, 일치성 검증, 시뮬레이션 등을 지원

III. 콘솔 기반 UI

<순차적 일치성 검증 예제>

```
--State 0:
state$NTK2:S1
state:S0
...
--Goes to state 1:
state:S1
timer$NTK2:T1
timer:T1
--On input:
f_X<0>:0
f_X<2>:1
...
Networks are NOT sequentially equivalent..
```

<시뮬레이션 예제>

```
.inputs f_X_Raw<0> f_X_Raw<1> ...
.latches AA.Prev_th_Reset_Ini AA.state ...
.outputs th_X_Pretrip
.initial 0 A0 T5 S1 0 0 1 0 0 1 1 1 ...

.start_vectors

# f_X_Raw<0> f_X_Raw<1> f_X_Raw<3> ...
.state DD.th_Prev_X_Pretrip DD.timer ...

1 0 1 1 1 1 0 0 ; 0 A0 T5 S1 0 0 1 0 0 1 1 1 ...
1 0 1 1 1 1 0 1 ; 0 A0 T5 S1 0 0 1 0 0 1 1 0 ...
1 0 1 1 1 1 0 1 ; 1 A1 T0 S0 1 1 0 0 0 1 1 1 ...
```

3. VIS 일치성 검증 기능의 활용

VIS의 순차적 일치성 검증을 활용하여 수정된 두 프로그램 간의 행동 일치성을 정형 검증 할 수 있다.



Program₁과 Program_N의 행동이 일치함을 검증

4. VIS 순차적 검증기능의 문제점

검증 결과가 불충분함

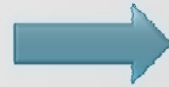
- 각 상태 별 출력 값이 나오지 않음
- 각 상태 별 상태 값이 나오지 않음
- 입력 값의 출력이 불충분 함

수행 과정이 불편함

- 콘솔 기반의 UI임
- 모든 명령어를 수동 입력해야 함
- 검증 결과 해석이 어려움

5. VIS Analyzer의 솔루션 (1/2)

검증 결과가
불충분함



VIS 시뮬레이션
활용

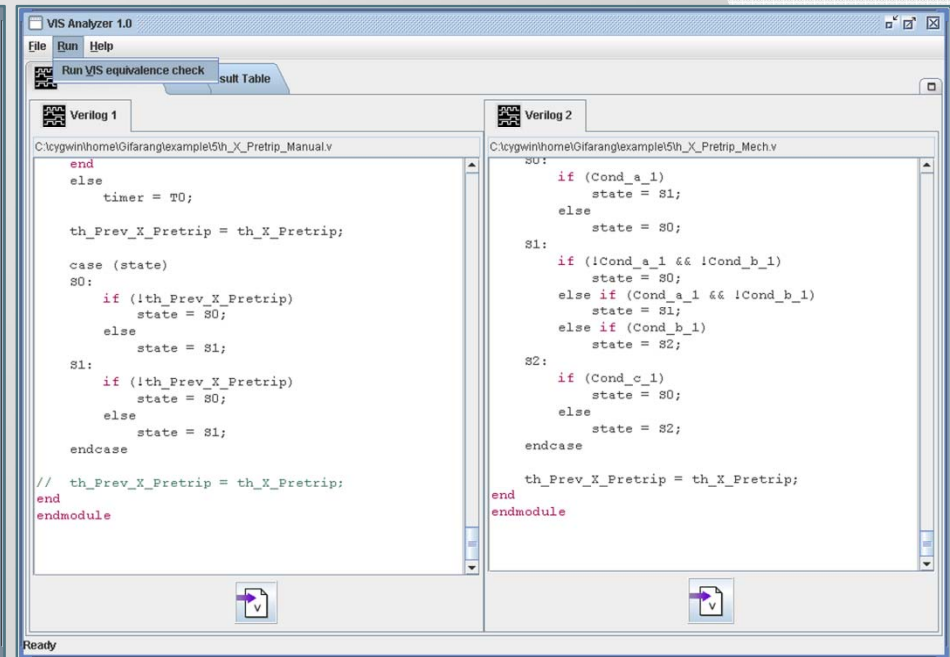
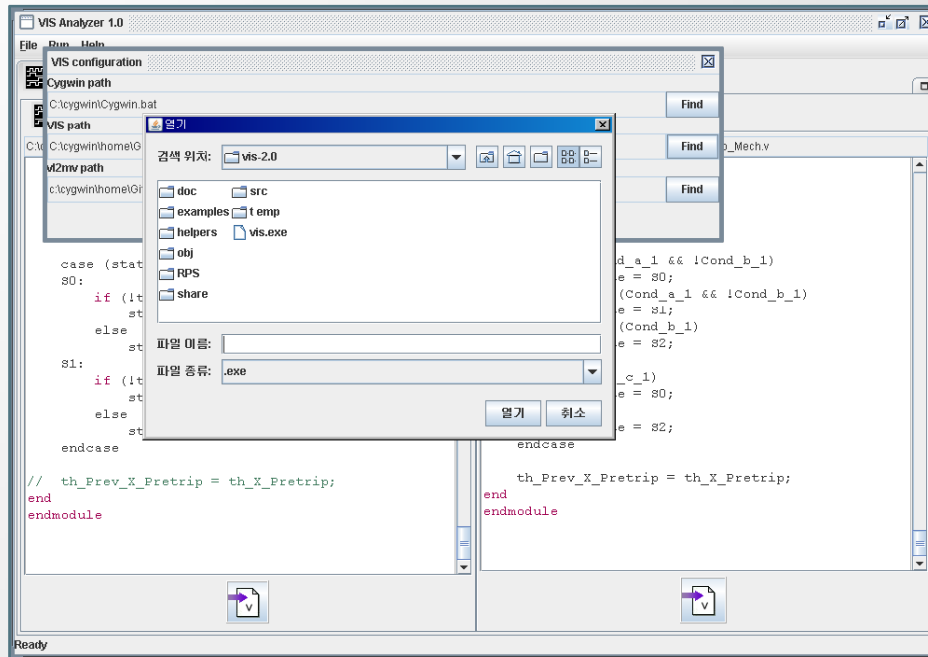
수행 과정이
불편함



그래픽 기반 UI
출력 형식 개선

- I. VIS 순차적 일치성 검증 결과를 바탕으로 VIS 시뮬레이션을 수행하여 순차적 일치성 검증에서 제공하지 않는 출력, 상태 변화 값을 보완함
- II. 그래픽 기반의 UI를 제공하여 사용이 편리
- III. 각종 VIS 명령어와 검증에 필요한 파일 생성을 자동으로 수행하여 검증 절차를 자동화함
- IV. 검증 및 시뮬레이션 결과가 표로 출력되어 가독성이 높음

6. VIS Analyzer의 솔루션 (2/2)



The screenshot shows the 'Result Table' window in VIS Analyzer 1.0. The table displays the results of the equivalence check, including state numbers, inputs, and outputs for two files.

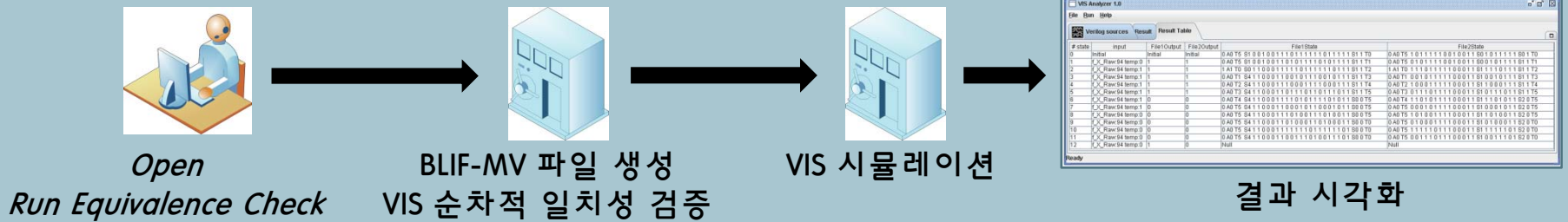
# state	input	File1Output	File2Output	File1State	File2State
0	Initial	Initial	Initial	0 A0 T5 S1 0010011101111110111111S11 T0	0 A0 T5 10111110010011S01011111S01 T0
1	f_X_Raw:94 temp:0	1	1	0 A0 T5 S1 0010011010111101011111S11 T1	0 A0 T5 01011110010011S0010101111S11 T1
2	f_X_Raw:94 temp:1	1	1	1 A1 T0 S0 1100011111011111101111S11 T2	1 A1 T0 11101111100011S11110111S11 T2
3	f_X_Raw:94 temp:1	1	1	0 A0 T1 S4 1100011001011100101111S11 T3	0 A0 T1 00101111100011S10010111S11 T3
4	f_X_Raw:94 temp:1	1	1	0 A0 T2 S4 1100011100011110001111S11 T4	0 A0 T2 10001111100011S11000111S11 T4
5	f_X_Raw:94 temp:1	1	1	0 A0 T3 S4 110001101110110110111011S11 T5	0 A0 T3 01110111100011S10111011S11 T5
6	f_X_Raw:94 temp:1	0	0	0 A0 T4 S4 1100011110101111101011S00 T5	0 A0 T4 11010111100011S11101011S20 T5
7	f_X_Raw:94 temp:1	0	0	0 A0 T5 S4 110001100010110001011S00 T5	0 A0 T5 00010111100011S10001011S20 T5
8	f_X_Raw:94 temp:0	0	0	0 A0 T5 S4 110001110100111010011S00 T5	0 A0 T5 10100111100011S11010011S20 T5
9	f_X_Raw:94 temp:0	0	0	0 A0 T5 S4 110001101000110100011S00 T0	0 A0 T5 01000111100011S10100011S20 T0
10	f_X_Raw:94 temp:0	0	0	0 A0 T5 S4 11000111111101111101S00 T0	0 A0 T5 11111011100011S11111101S20 T0
11	f_X_Raw:94 temp:0	0	0	0 A0 T5 S4 110001100111010011101S00 T0	0 A0 T5 00111011100011S10011101S20 T0
12	f_X_Raw:94 temp:0	1	0	Null	Null

7. 검증 과정 비교

VIS



VIS Analyzer



8. 결론

- I. 안전이 최우선시 되는 프로그램을 개발 할 때 정형 기법을 도입하는 것은 이상적이다.
- II. VIS는 정형 기법에 활용될 수 있는 도구로서 두 프로그램 간의 순차적 일치성을 정형적으로 검증할 수 있는 기능을 제공한다.
- III. VIS는 실행환경과 검증 결과에 문제점을 가지고 있다.
- IV. VIS Analyzer는 VIS가 가지는 어려움을 VIS 시뮬레이션 기능과 GUI 및 표를 이용한 출력으로 극복하고자 하였다.

9. 앞으로 연구 방향

- I. VIS가 제공하는 CTL 모델체킹을 효율적으로 실행하고 결과를 시각화 할 수 있는 기능을 구현한다.
- II. 조합적 일치성 검증을 효율적으로 실행하고 결과를 분석해 시각화 하는 기능을 구현한다.
- III. VIS 순차적 일치성 검증 결과에 보여지지 않는 마지막 상태의 입력 값을 알아내서 검증 결과를 완전히 하는 기능을 구현한다.