NuSCR의 검사를 위한 Quick Checker 개선

건국대학교 컴퓨터공학부
조재연, 윤상현, 유준범

차례

1. 서론

- 안전 필수 소프트웨어는 높은 안전성을 위해서 요구사항 명세에 대한 모델 체킹과 안전성 분석을 한다.
- 안전성 분석과 모델 체킹을 위해서 명세의 정확성을 확보해야 한다.
- 이전의 연구에서 디지털 원자로 보호 시스템의 소프트웨어 요구사항을 명세하기 위한 언어인 NuSCR과 이를 지원하기 위한 도구인 NuSRS가 제시되었다.
- NuSRS에서는 NuSCR로 명세 된 요구사항을 정적인 관점에서 검사하기 위해 Quick Check 모듈을 사용하였다.
- 기존 Quick Check를 개선하여 Quick Checker를 제시한다.

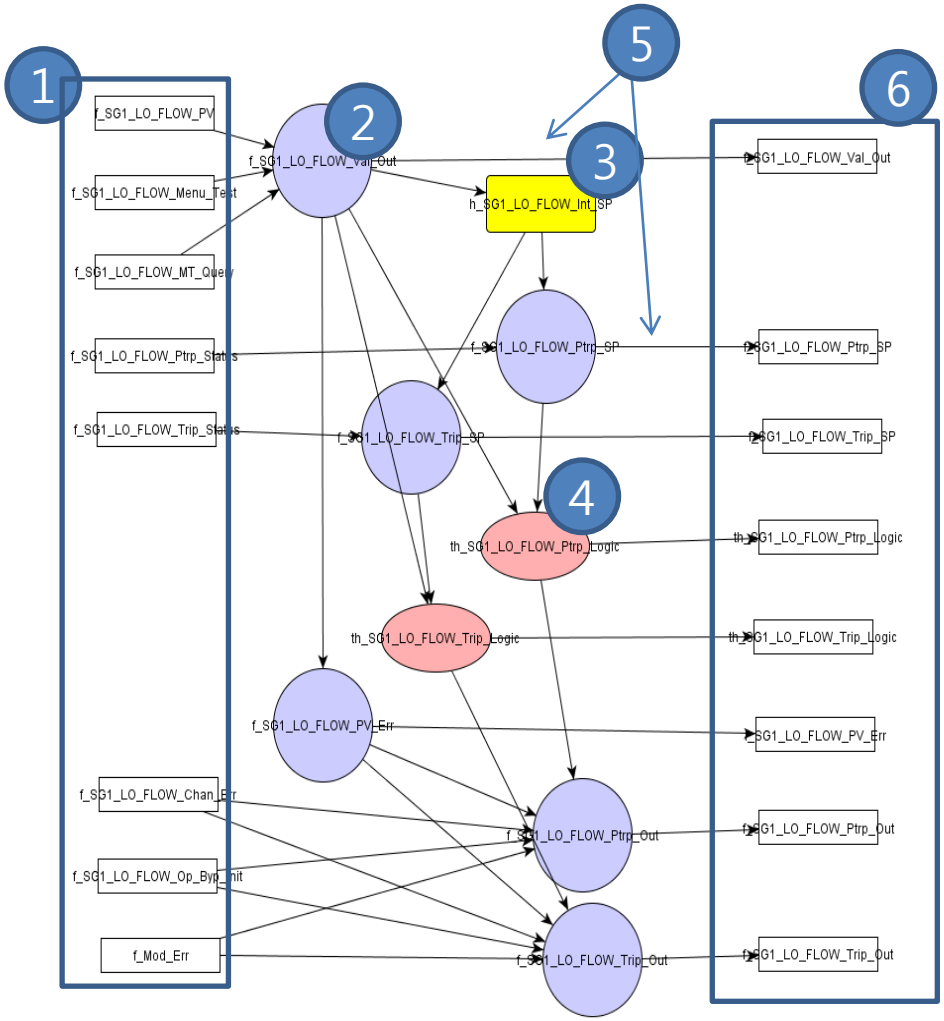
2.1 NuSCR

- 디지털 원자로 보호 시스템의 소프트웨어 요구사항을 명세하기 위한 언어이다.
- SCR(Software Cost Reduction)을 확장한 언어이다.
- APR-1400 원자로 보호시스템의 소프트웨어 프로토타입을 개발하는데 사용되었다.
- NuSCR로 작성된 명세는 여러 개의 FOD로 구성된다.

- FOD(Function Overview Diagram)
  - 노드들 간의 관계를 다루는 구조를 표현한다.
  - 노드들은 Function variable node, Timed history variable node, History variable node, input node, output node 로 이루어져 있다.

2.1 NuSCR - Function Overview Diagram(FOD)

① input node
② function variable node – SDT(Structural Decision Table)
③ history variable node - FSM(Finite State Machine)
④ timed history variable node - TTS(Time Transition System)
⑤ transition
⑥ output node



2.1 NuSCR - Structural Decision Table(SDT)

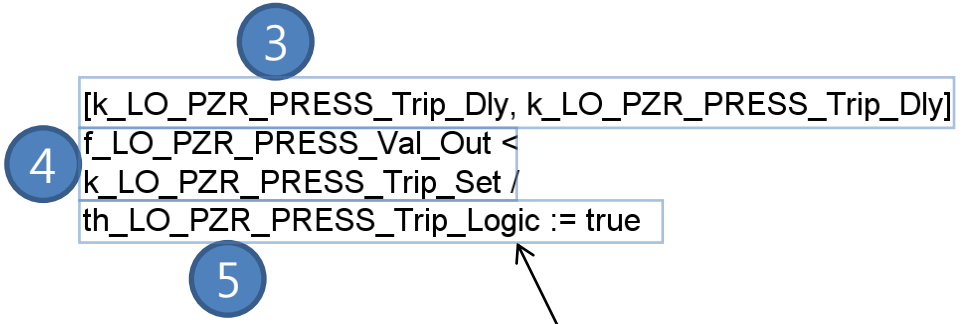
①조건 구문
②행동 구문
③조건 기술
　　해당 조건의 참/거짓을 기술한다. (T(true), F(False), -(Don't Care))
④행동 기술
　　해당 행동의 발생 여부를 기술한다.

Conditions	1	2	3
f_SG1_LO_FLOW_Val_Out > k_SG1_LO_FLOW_PV_Max	T	-	F
f_VAR_OVER_PWR_Val_Out < k_VAR_OVER_PWR_PV_Min	-	T	F
Action	**1**	**2**	**3**
f_SG1_LO_FLOW_PV_Err := true	O	O	
f_SG1_LO_FLOW_PV_Err := false			O

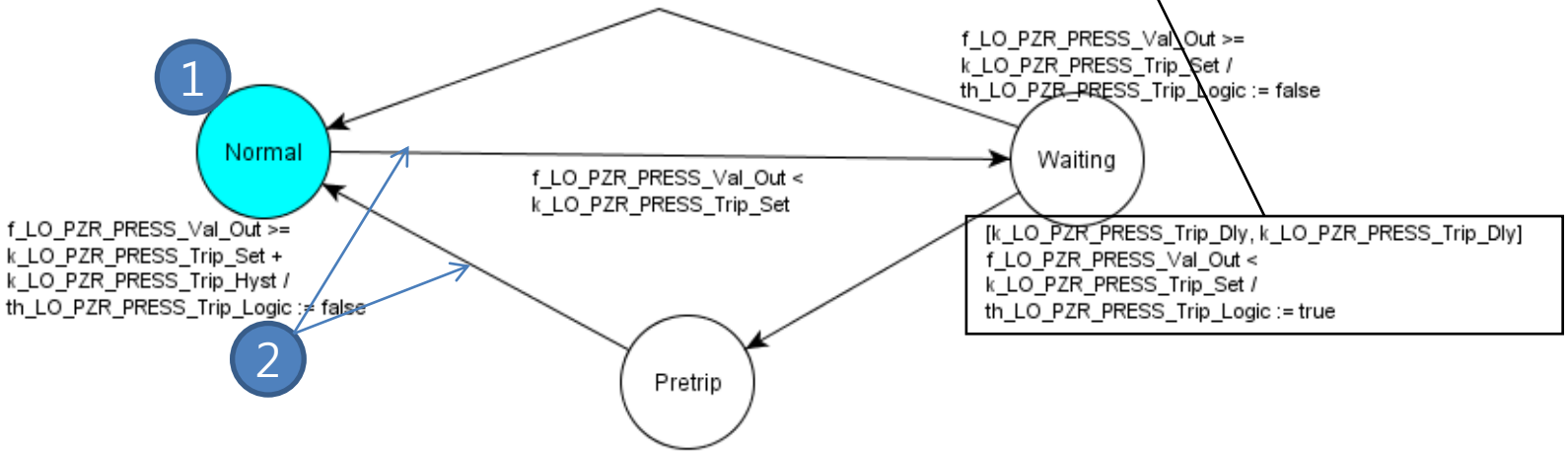
1
2
3
4

2.1 NuSCR - Finite State Machine(FSM) Timed Transition System(TTS)

① 초기 상태
② transition
③ transition의 시간 조건
④ transition의 조건
⑤ transition의 행동

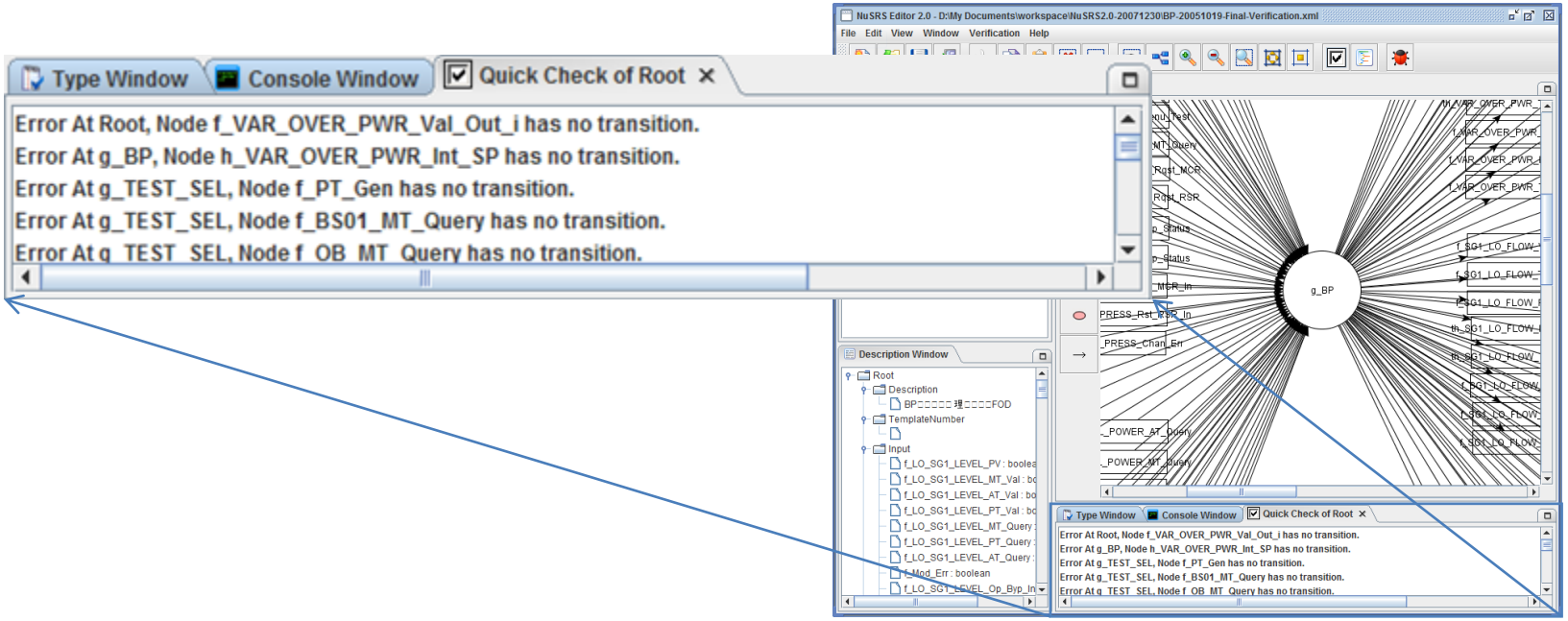


- TTS는 FSM에 시간조건이 있는 transition를 추가할 수 있다.



2.1 NuSCR - NuSRS

- NuSRS는 NuSCR의 작성 및 검증을 위한 도구이다.
  - 모델 체킹을 위해서 NuSCR언어를 NuSCRtoSMV를 통하여 SMV model checker 의 입력 언어로 자동으로 변환한다.
  - NuFTA를 이용하여 Software Fault Tree를 생성한다.
- Quick Check 모듈은 NuSCR을 정적인 관점에서 검사한다.



2.2 Constraint Satisfaction Problem(CSP)

- 여러 개의 수식이 있을 경우, 각 수식이 동시에 참이 될 수 있는 경우를 찾는 문제이다.
- 변수의 집합, 값들의 영역, 제약사항들의 집합으로 이루어져 있다.
- NuSCR 명세를 각 수식이 동시에 참이 될 수 없게 기술하였다면, 제시한 수식에 문제가 있다고 판단할 수 있다.
- CSP를 해결하기 위해 CSP Solver를 이용하였다.
  - Quick Checker에서 CSP Solver 라이브러리를 사용하였다.
  - 조건구문의 모순여부를 검사하기 위해 사용하였다.

2.2 CSP 예

① n_variable_1 = 1
② n_variable_2 < 1
③ n_variable_1 = n_variable_2
④ -6 < n_variable_1 < 5
⑤ -3 < n_variable_2 < -1

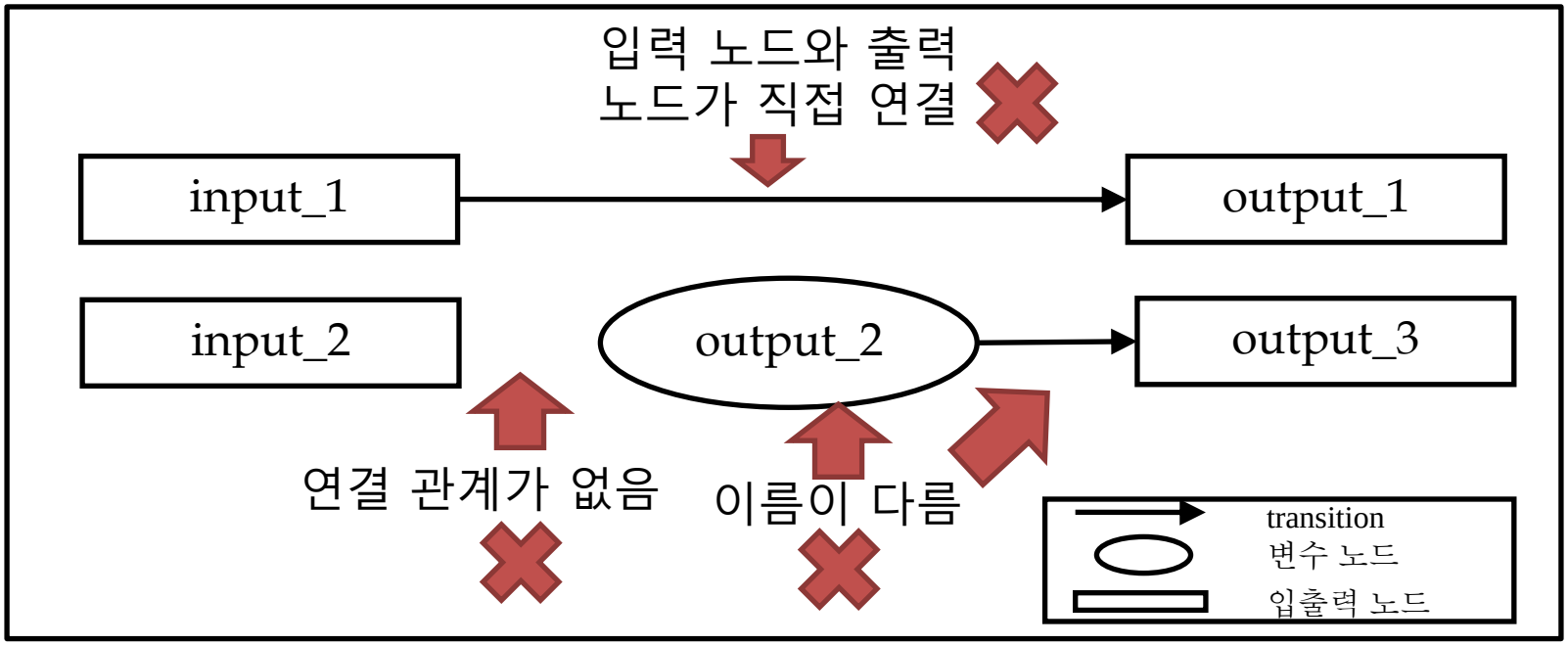
CSP Solver로 검사

동시에 만족하는 해를 찾을 수 없음
①, ②, ③

CSP를 만족하지 못한 것으로 결론

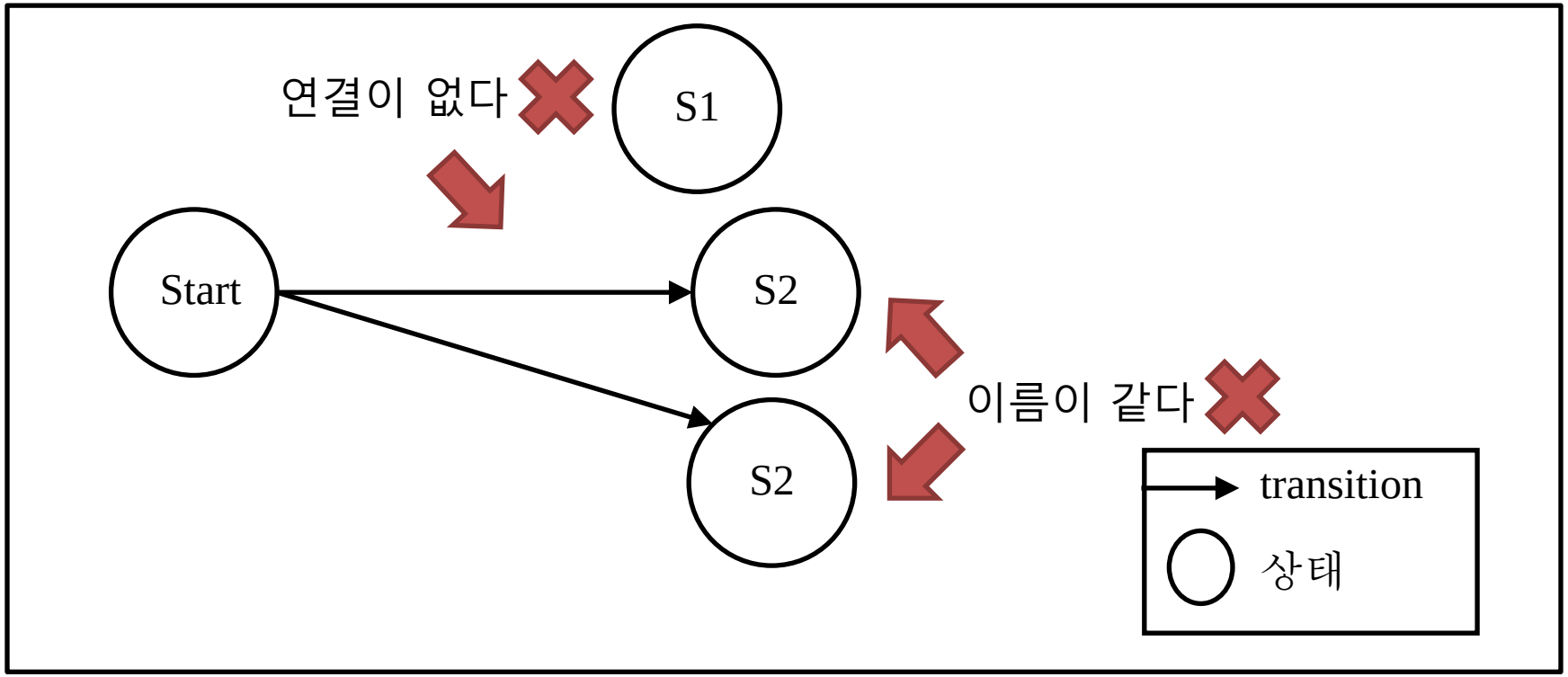
3.1 NuSCR 검사에서 고려해야 할 점 - FOD

- 변수 노드와 출력 노드는 서로 이름이 같아야 한다.
- 입출력 노드는 반드시 변수 노드와 연결되어 있어야 한다.
  - Quick Checker에서 개선된 부분
- 입력 노드와 출력 노드가 직접 연결이 되면 안 된다.



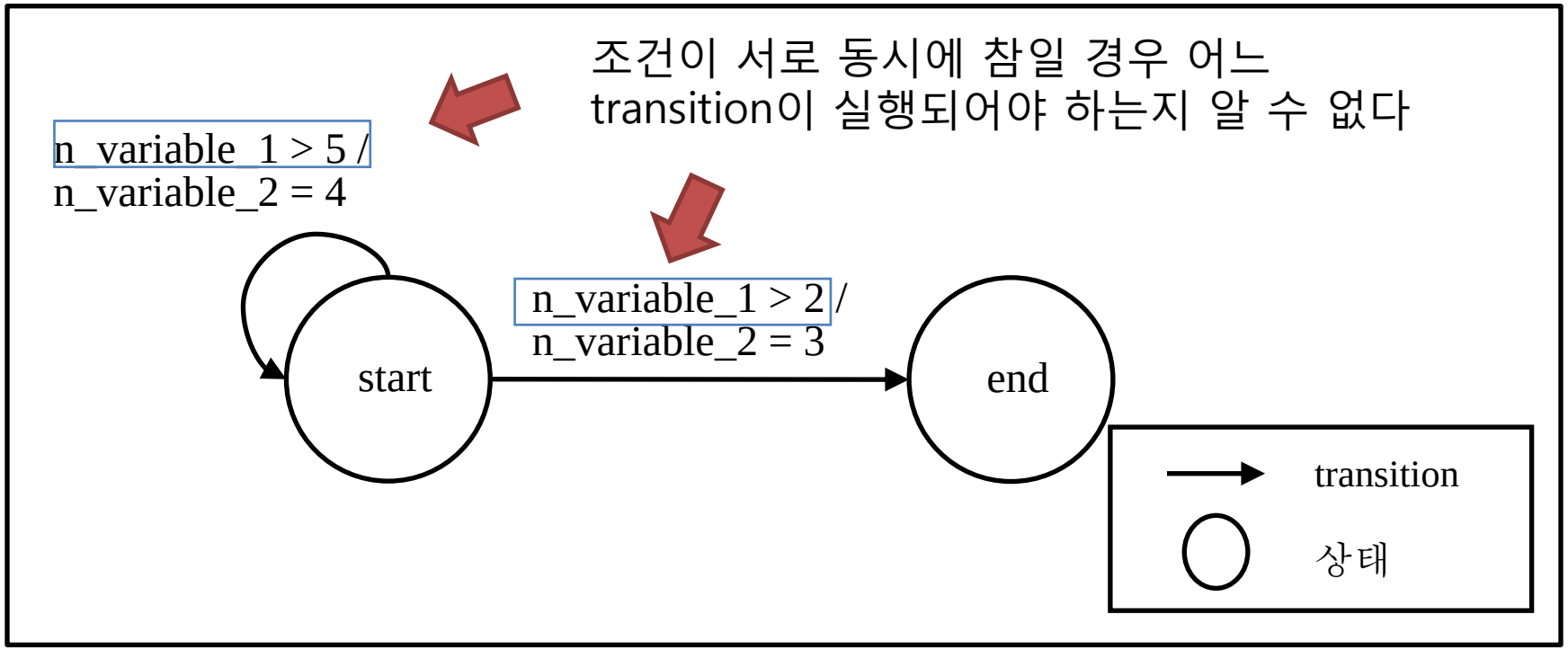
3.1 NuSCR 검사에서 고려해야 할 점 - FSM, TTS

- 상태는 반드시 transition으로 연결이 되어 있어야 한다.
- 이름이 같은 상태가 2개 이상 있으면 안 된다.



3.1 NuSCR 검사에서 고려해야 할 점 - FSM, TTS

- 하나의 상태에서 연결된 transition의 조건이 서로 모순이 없어야 한다.
  - 하나의 transition이 실행되기 위해서 실행되는 transition의 조건이 참, 나머지 transition의 조건은 거짓이어야 한다.



CSP를 이용한 해결 - FSM, TTS

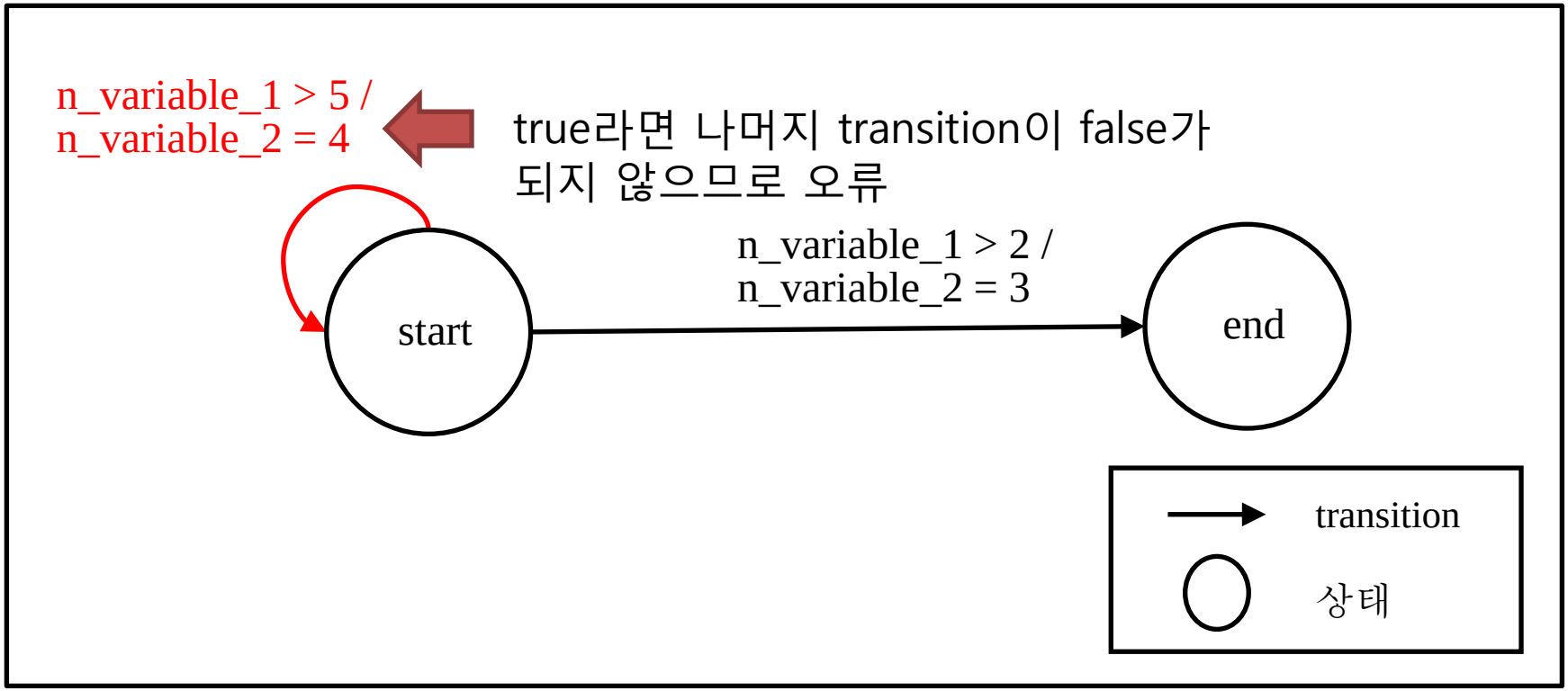
- 하나의 상태 에서 여러 transition이 있을 경우 하나의 transition의 조건을, 나머지는 조건의 inverse를 하나의 case로 놓고 CSP Solver로 검사한다.

Case 1	Case 2
n_variable_1 > 5	!(n_variable_1 > 5)
!(n_variable_1 > 2)	n_variable_1 > 2

CSP Solver Library

Case 1	Case 2
n_variable_1 > 5	!(n_variable_1 > 5)
!(n_variable_1 > 2)	n_variable_1 > 2

3.1 NuSCR 검사에서 고려해야 할 점 - FSM, TTS



3.1 NuSCR 검사에서 고려해야 할 점 - SDT

- 각 조건의 참/거짓에 대한 행동이 명확해야 한다.
  - 조건이 일어날 수 있는 모든 조합(아래의 경우 (T,T),(T,F),(F,T)(F,F))을 기준으로 조건의 참/거짓의 조합을 기술하지 않은 경우를 검사하였다.
  - 아래의 예에서 (C1,C2)의 (F,F), (T,T), (F,T)를 기술하지 않았다.
  - 조건을 기술하지 않은 경우도 검사하였다.
- 조건의 경우가 서로 중복인 경우가 없어야 한다.
  - 중복인 부분을 검사하였다.
  - 아래의 예에서 (T,F)가 중복이다.

Condition이 없다

	Condition			3
C1	n_variable_1>3		T	T
C2	n_variable_2>3	T	F	F
	Action	1	2	3
A1	n_variable_1=6		O	O
A2	n_variable_1=8			

Action이 2개 이상

3.1 NuSCR 검사에서 고려해야 할 점 - SDT

- 조건 구문을 동시에 만족시킬 수 없는 경우에 요구사항에 문제가 있다.
  - C1과 C2의 참/거짓을 동시에 만족시킬 수 없는 경우가 생기면 해당 행동은 실행할 수 없다.

	Condition				
C1	n_variable_1>3	T	T	F	F
C2	n_variable_1>4	T	F	T	F
	Action				
A1	n_variable_1=6	O	O		
A2	n_variable_1=8			O	O

T,F : True, False
O : 행동 지정

3.1 NuSCR 검사에서 고려해야 할 점 - SDT

- 각 조건이 참/거짓이 될 수 있는 모든 경우를 case로 놓고 CSP Solver library로 검사한다.

Case 1	Case 2	Case 3	Case 4
n_variable_1>3	n_variable_1>3	!(n_variable_1>3)	!(n_variable_1>3)
n_variable_1>4	!(n_variable_1>4)	n_variable_1>4	!(n_variable_1>4)

CSP Solver Library

Case 1	Case 2	Case 3	Case 4
n_variable_1>3	n_variable_1>3	!(n_variable_1>3)	!(n_variable_1>3)
n_variable_1>4	!(n_variable_1>4)	n_variable_1>4	!(n_variable_1>4)

3.1 NuSCR 검사에서 고려해야 할 점 - SDT

- C1 = F, C2 = T (n_variable_1>4, n_variable_1<=3)일 경우에 모순이 발생한다.

	Condition				
C1	n_variable_1>3	T	T	F	F
C2	n_variable_1>4	T	F	T	F
	Action				
A1	n_variable_1=6	O	O		
A2	n_variable_1=8			O	O

C1,C2 : 조건
A1,A2: 행동
T,F : True, False
O : 행동 지정

3.2 Parser

- 이전 Quick Check에서 조건 구문과 행동구문을 구별하지 않고 검사하였다.
  - 이를 해결하기 위해서 검사하는 문법을 변경하였다.
  - 조건 구문과 행동 구문을 구분하였다.
- JavaCC를 이용하여 Parser를 구현하였다.
  - 문법을 검사하였다.
  - CSP Solver 사용을 위해서 파싱을 하여 나온 파스트리를 CSP Solver 에 입력이 가능한 형태로 변환하였다.

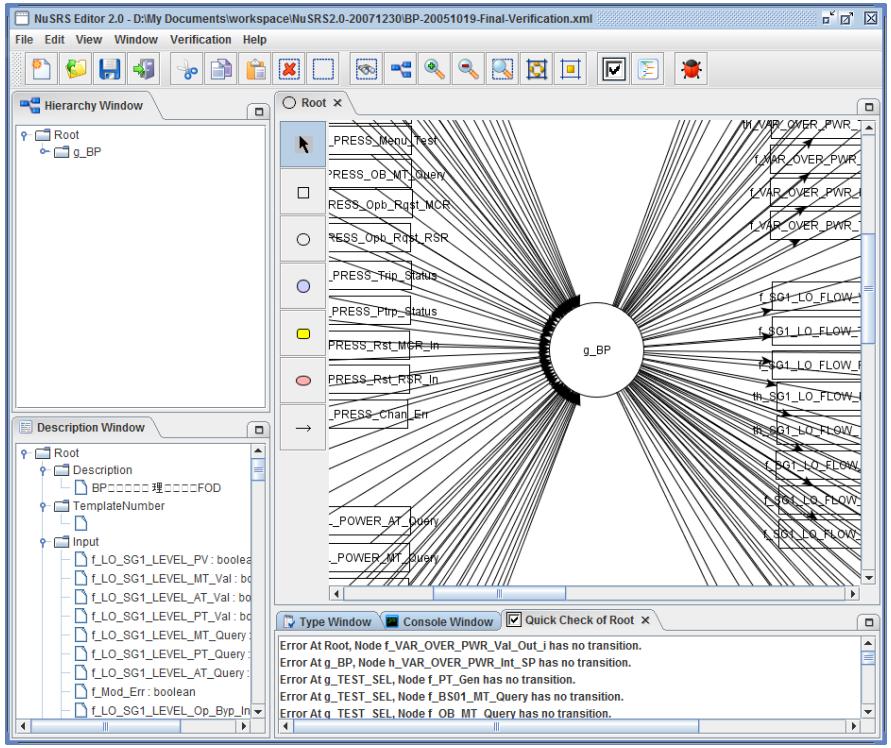
3.3 Quick Checker - 보완점

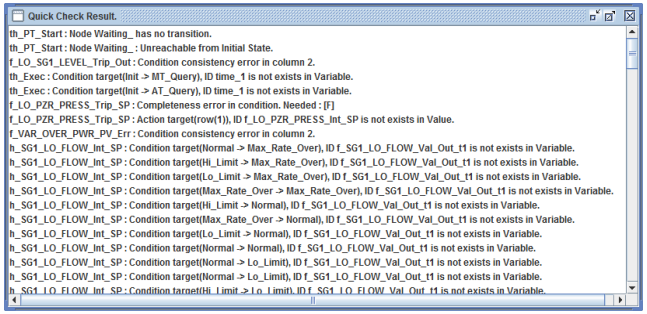
- FOD
  - 변수 노드로 가는 입력transition이 없는 경우를 검사하였다.

- FSM, TTS
  - TTS에서 최소 시간보다 최대 시간이 크지 않은지 검사하였다.
  - 조건부의 문법과 행동부의 문법을 구분하여 검사하였다.
  - 각 transition의 조건이 참이고 나머지 transition의 조건이 거짓인 경우를 만족하는 변수를 찾아서 해당 transition이 결정될 수 있는지를 검사하였다.
    - CSP 해결 라이브러리를 사용하여 검사하였다.
- SDT
  - 조건기술부가 모든 경우를 기술하는지, 중복된 경우를 기술하는지를 검사하였다.
  - 조건부의 문법과 행동부의 문법을 구분하여 검사하였다.
  - 각 조건 구문이 기술된 대로 참/거짓이 결정된 경우를 만족하는 변수를 찾아서 해당 기술이 유효한지를 검사하였다.
    - CSP 해결 라이브러리를 사용하여 검사하였다.

4. 사례연구

- APR-1400 원자로 보호 시스템의 프로토 타입 명세를 검사한 예이다.
- 입출력노드가 아닌 노드로 transition이 들어오지 않는 경우와 SDT가 제대로 기술되었는지의 여부를 새로 검사하였다.





Quick Checker

NuSRS(Quick Check)

결과 비교

이전 Quick Check

오류 유형	개수
transition없음	11
정의되지 않은 변수	66
초기상태로부터 도달할 수 없음	2

구현된 Quick Checker

오류 유형	개수
transition없음	11
정의되지 않은 변수	66
초기상태로부터 도달할 수 없음	2
입력이 없는 노드	1
Completeness 오류	14
Consistency 오류	18

5. 결론 및 향후 계획

- Quick Checker
  - 기존 Quick Check의 기능을 보완하였다.
    - TTS, FOD, SDT, FSM의 검사범위를 확장하였다.
  - NuSRS가 지원하는 기능인 NuSCRtoSMV를 이용한 모델 체킹과 NuFTA를 이용한 안전성 분석을 위해 NuSCR specification의 Completeness와 Consistency을 확보하였다.

- CSP Solver를 이용하여 검사하는 기능을 보완할 예정이다.