SPIN

Advanced Software Engineering

2017.04.03

KONKUK UNIVERSITY ITCS

노은방, 심우진

CONTENTS

1. Introduction

- What is SPIN?
- What is Promela?

2. Body

- How to Install SPIN?

3. Conclusion

- How to Execute SPIN?

1. Introduction

1. Introduction

- What is SPIN?
- What is Promela?

2. Main

- How to Install SPIN?

3. Conclusion

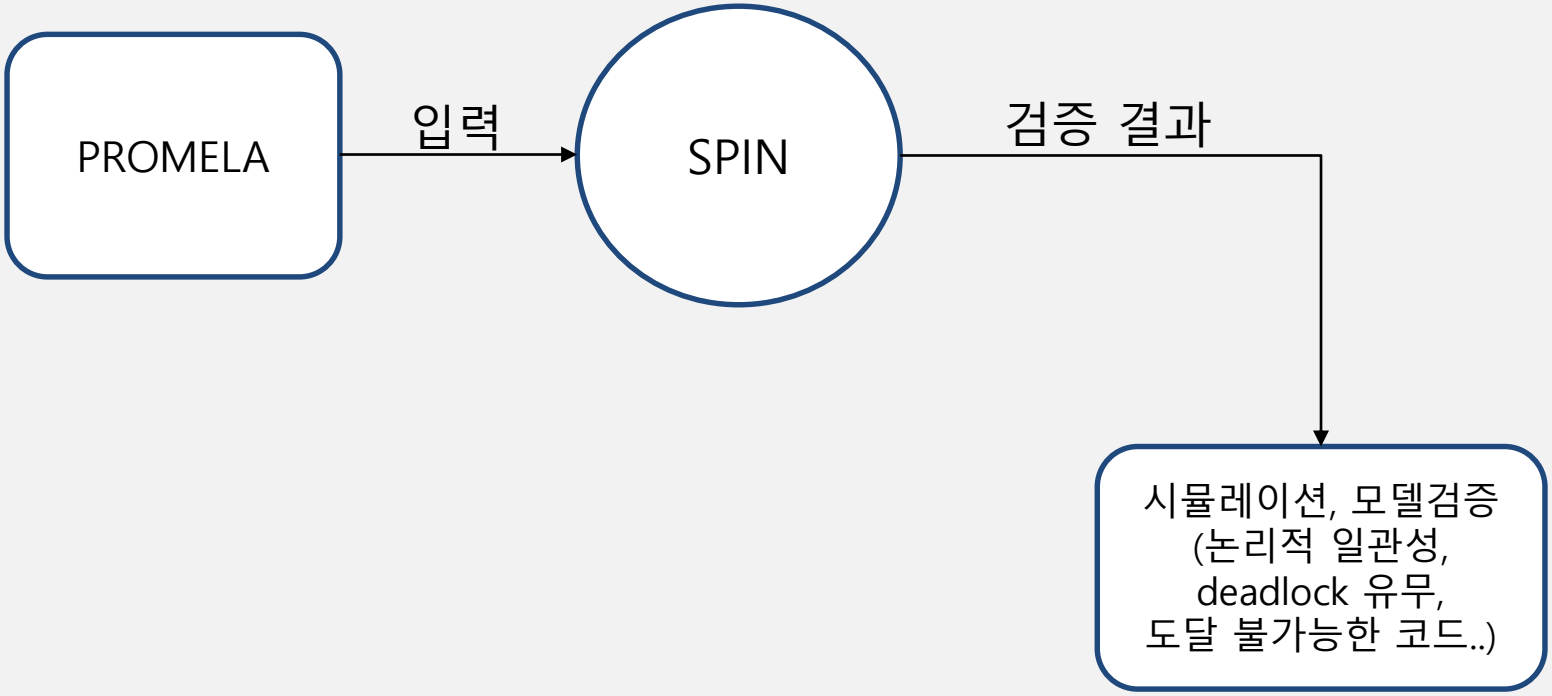
- How to Execute SPIN?

SPIN(Simple Promela Interpreter)

- 오픈 소스 기반의 소프트웨어 모델을 검증하는 도구.
  -> 주어진 모델과 속성 값을 통해 해당 모델의 검증 결과가 참인지 거짓인지 판별
- 분산 시스템 설계의 논리적 설계 오류를 검출에 활용.
  ex.) 운영체제, 데이터 통신 프로토콜, 우주선 제어 소프트웨어, 철도 신호 프로토콜 등.
- On-the-fly 검증 도구.
  -> 시스템 속성 확인을 위해 Kripke 모형과 같은 구조를 사전 구성할 필요가 없다.
- Promela(Process Meta Language) 검증 모델링 언어가 사용.

SPIN

SPIN(Simple Promela Interpreter)



SPIN_Mode

① 시뮬레이터(Simulator)
-> random : Default 옵션
-> guided : (-t) 이전 검증에서 발생한 오류 경로를 확인
-> interactive : (-i) 둘 이상의 경우가 있을 때, 사용자 입력을 받음.

② Exhaustive Verifier(철저한 검증 도구)
-> 사용자가 지정한 정확한 요구 사항을 엄격하게 증명.

③ Proof approximation system(근사 증명 시스템)
-> 모델의 크기가 큰 경우

④ Swarm Verification(스웜 검증)
-> 코어의 개수가 큰 모델 처리의 경우(클라우드 네트워크, 병렬처리 모델)

Promela(Process Meta Language)

- SPIN에서 사용되는 검증 모델링 언어
- 유한상태 전이 시스템을 정의하기 위한 언어
- 프로세스가 동적으로 생성
- 메시지 채널을 이용한 메시지의 동기적 / 비동기적 통신 모델링 가능.
- 구문의 실행 가능성(Executability)으로 동기화 수단 제공.

Promela(Process Meta Language)

- Promela Composition
  - 프로세스 : 행위 기술.
  - 채널 : 프로세스의 동작 환경 기술.
  - 변수 : 프로세스의 동작 기술(지역) / 환경 기술(전역).

```
init {
    ...
}

proctype transfer(chan i,o)
{
...
}
```

프로세스 선언, 수행

```
chan AtoB = [1] of {mtype, byte};
chan BtoA = [1] of {mtype, byte};
```

메시지 채널 선언

```
bit t,f
byte b_out,b_in;
short a_out,a_in;
int out[6],in[6]
```

변수선언

```
out!accept(i);
in?next(o);
```

메시지 채널 예시

```
mtype = {ack, nak, err};
```

타입 선언

2. Main

1. Introduction

- What is SPIN?
- What is Promela?

2. Main

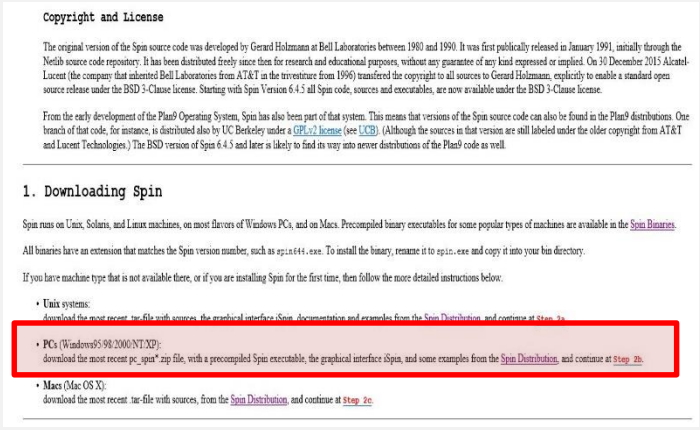
- How to Install SPIN?

3. Conclusion

- How to Execute SPIN?

How to Install Spin?

SPIN 다운로드 및 환경변수 설정



Copyright and License

The original version of the Spin source code was developed by Gerard Holzmann at Bell Laboratories between 1980 and 1990. It was first publically released in January 1991, initially through the Netlib source code repository. It has been distributed freely since then for research and educational purposes, without any guarantee of any kind expressed or implied. On 30 December 2015 Alcatel-Lucent (the company that inherited Bell Laboratories from AT&T in the trivestiture from 1996) transfered the copyright to all sources to Gerard Holzmann, explicitly to enable a standard open source release under the BSD 3-Clause license. Starting with Spin Version 6.4.5 all Spin code, sources and executables, are now available under the BSD 3-Clause license.

From the early development of the Plan9 Operating System, Spin has also been part of that system. This means that versions of the Spin source code can also be found in the Plan9 distributions. One branch of that code, for instance, is distributed also by UC Berkeley under a GPLv2 license (see UCB). (Although the sources in that version are still labeled under the older copyright from AT&T and Lucent Technologies.) The BSD version of Spin 6.4.5 and later is likely to find its way into newer distributions of the Plan9 code as well.

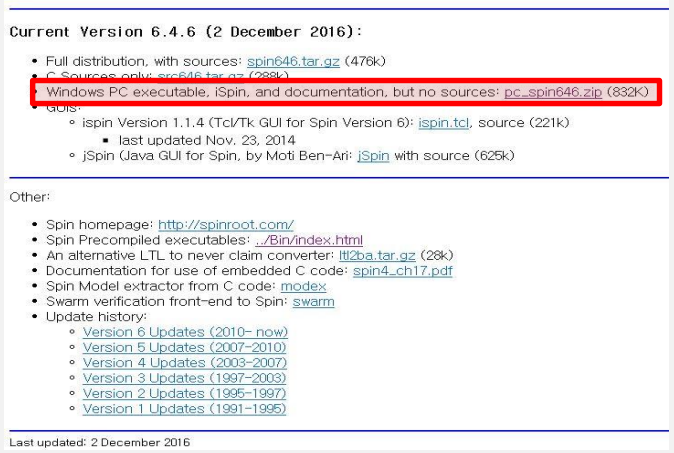
1. Downloading Spin

Spin runs on Unix, Solaris, and Linux machines, on most flavors of Windows PCs, and on Macs. Precompiled binary executables for some popular types of machines are available in the Spin Binaries.

All binaries have an extension that matches the Spin version number, such as spin644.exe. To install the binary, rename it to spin.exe and copy it into your bin directory.

If you have machine type that is not available there, or if you are installing Spin for the first time, then follow the more detailed instructions below:

- Unix systems:
  download the most recent .tar-file with sources, the graphical interface iSpin, documentation and examples from the Spin Distribution, and continue at Step 2a.
- PCs (Windows95/98/2000/NT/XP):
  download the most recent pc_spin*.zip file, with a precompiled Spin executable, the graphical interface iSpin, and some examples from the Spin Distribution, and continue at Step 2b.
- Macs (Mac OS X):
  download the most recent .tar-file with sources, from the Spin Distribution, and continue at Step 2c.



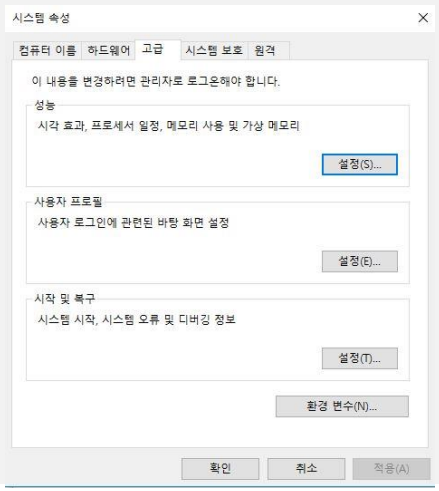
Current Version 6.4.6 (2 December 2016):

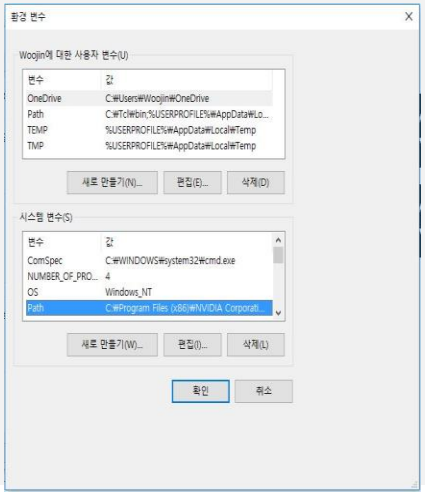
- Full distribution, with sources: spin646.tar.gz (476k)
- C Sources only: src646.tar.gz (288k)
- Windows PC executable, iSpin, and documentation, but no sources: pc_spin646.zip (832K)
- GUIs:
  - ispin Version 1.1.4 (Tcl/Tk GUI for Spin Version 6): ispin.tcl, source (221k)
    - last updated Nov. 23, 2014
  - jSpin (Java GUI for Spin, by Moti Ben-Ari: iSpin with source (625k)

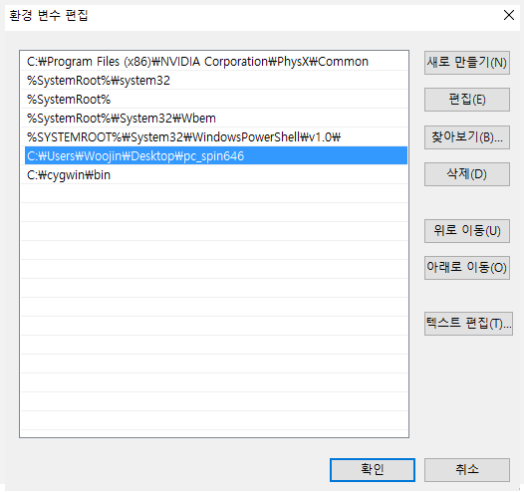
Other:

- Spin homepage: http://spinroot.com/
- Spin Precompiled executables: ../Bin/index.html
- An alternative LTL to never claim converter: ltl2ba.tar.gz (28k)
- Documentation for use of embedded C code: spin4_ch17.pdf
- Spin Model extractor from C code: modex
- Swarm verification front-end to Spin: swarm
- Update history:
  - Version 6 Updates (2010- now)
  - Version 5 Updates (2007-2010)
  - Version 4 Updates (2003-2007)
  - Version 3 Updates (1997-2003)
  - Version 2 Updates (1995-1997)
  - Version 1 Updates (1991-1995)

Last updated: 2 December 2016

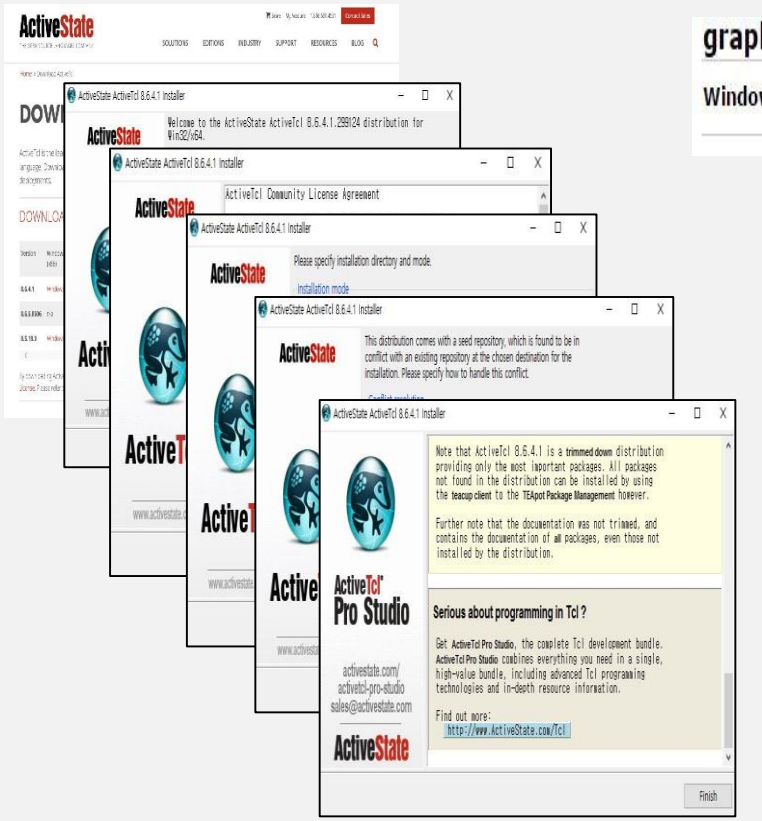


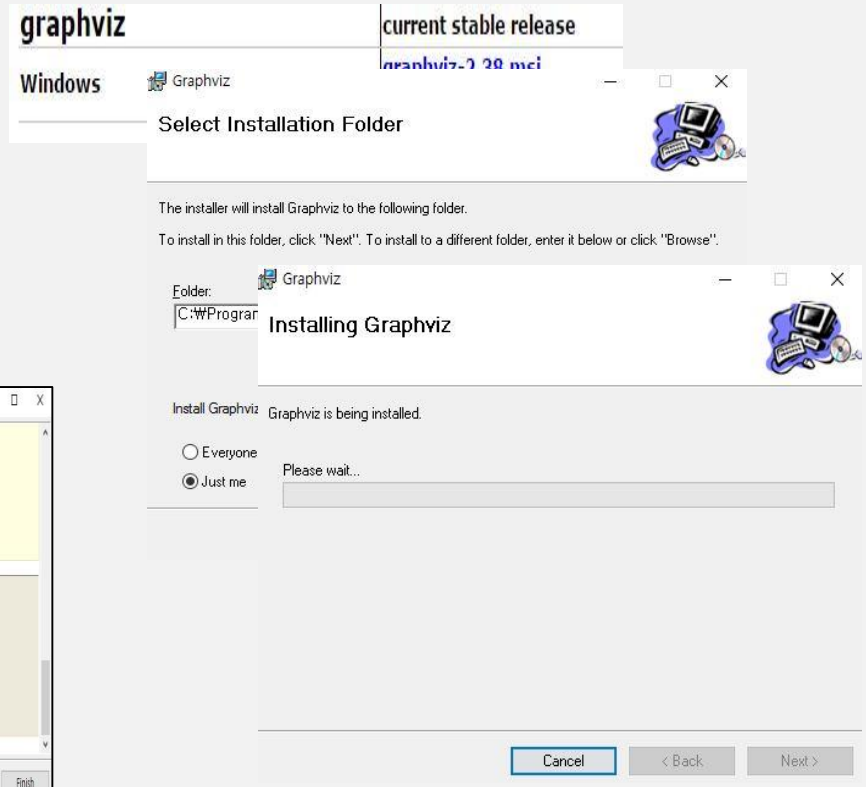




How to Install Spin?

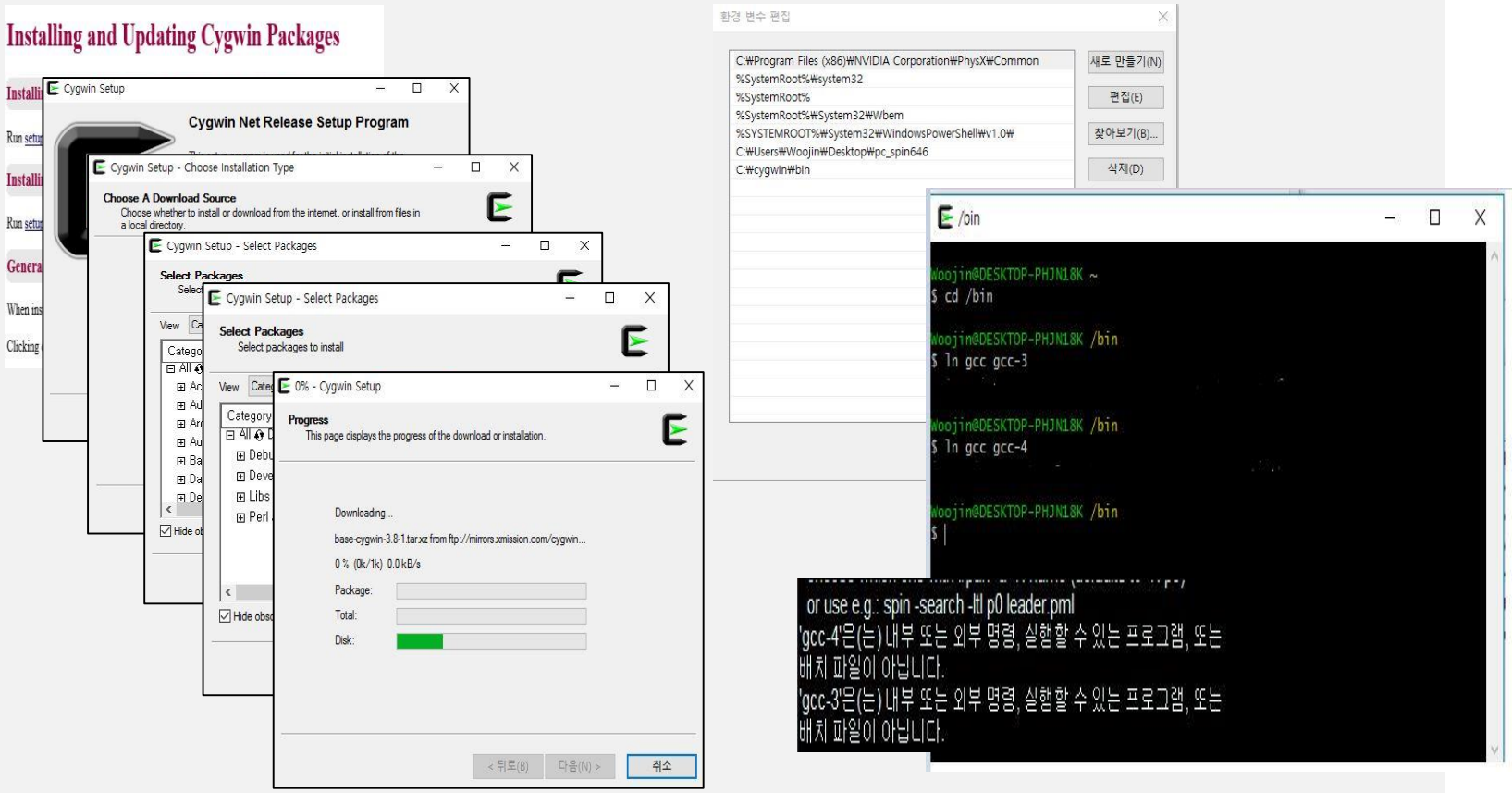
Active Tcl 및 Graphviz 설치





How to Install Spin?

Cygwin 및 gcc 설정



3. Couclusion

1. Introduction

- What is SPIN?
- What is Promela?

2. Main

- How to Install SPIN?

3. Conclusion

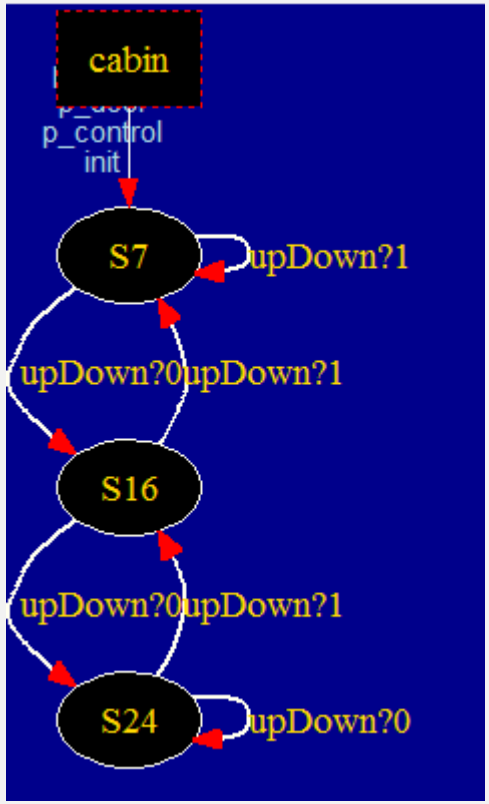
- How to Execute SPIN?

3. Couclusion

How to Execute SPIN

Smallish Elevator

```
proctype cabin(chan upDown)
{
do
:: atomic{upDown?UP} goto floor1
:: atomic{upDown?DOWN} goto floor()
Od
floor1 :
if
:: atomic{upDown?UP} goto floor2
:: atomic{upDown?DOWN} goto floor()
fi
floor2 :
do
:: atomic{upDown?UP} goto floor2
:: atomic {upDown?DOWN} goto floor1
od
}
```



3. Couclusion

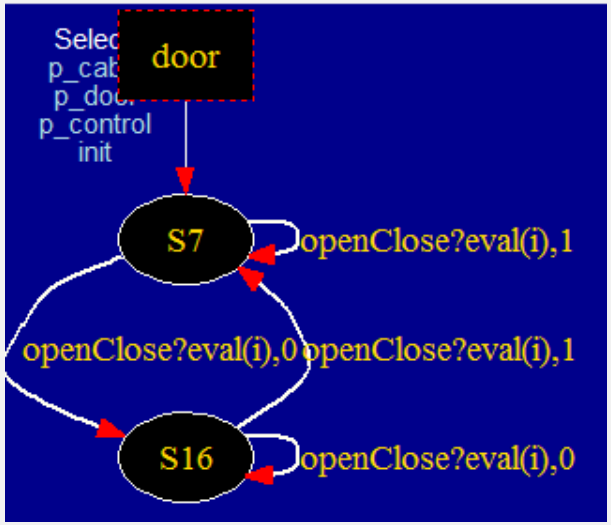
How to Execute SPIN

Smallish Elevator

```
proctype door(byte i; chan
openClose)
{
 bit curDoor = CLOSE;

 doorClose;
 do
  ::atomic{openClose?eval(i),OPEN}
goto doorOpen
  ::atomic{openClose?eval(i), CLOSE}
goto doorClose
 od

 doorOpen:
 do
  ::atomic{openClose?eval(i),OPEN}
goto doorOpen
  ::atomic {openClose?eval(i),CLOSE}
goto doorClose
 od
}
```

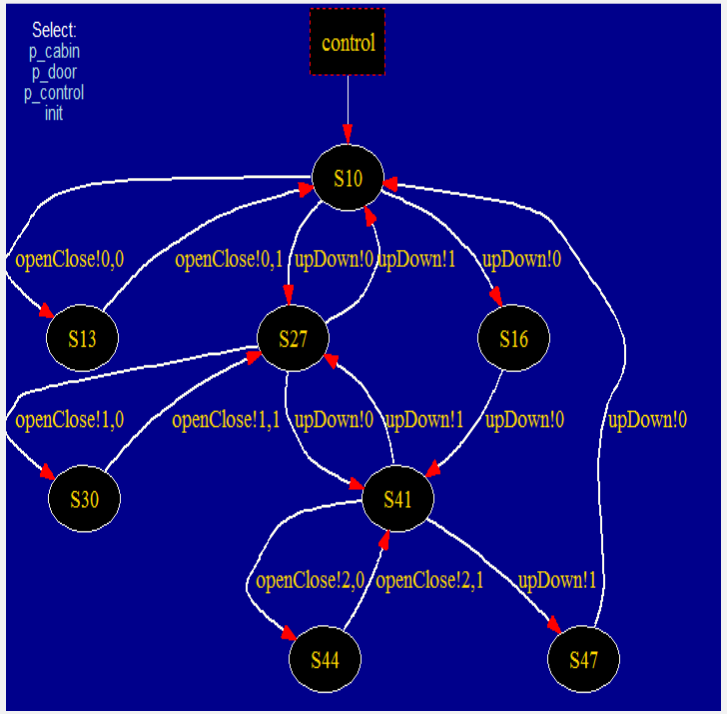


3. Couclusion

How to Execute SPIN

Smallish Elevator

```
proctype control(chan openClose, upDown)
{
On():
 if
 ::atomic{openClose!0,OPEN} goto free()
 ::atomic{upDown!UP} goto on1
 ::atomic{upDown!UP} goto to02
 fi
 free();
 atomic{openClose!1,CLOSE} goto on()
 to02
 atomic{upDown!UP} goto on2
 on1:
 if
 ::atomic{openClose!1,OPEN} goto free1
 ::atomic{upDown!UP} goto on2
 ::atomic{upDown!DOWN} goto on()
 fi
 on2:
 if
 ::atomic{openClose!2,OPEN} goto free2
 ::atomic{upDown!DOWN} goto on1
 ::atomic{upDown!DOWN} goto to2()
 fi
 free2:
 atomic{openClose!2,CLOSE} goto on2
 to 20:
 atomic{upDown!UP} goto on()}
 }
```



THANK YOU

PRESENTATIONPRESENTATIO