# A Practical Demonstration of the Model Checker NuSMV[1]

Viraj Wijesuriya

Computer-Aided Formal Verification
Week 6, Michaelmas term 2018

[1] The slides are provided, courtesy of Nathalie Cauchi

# What is NuSMV

NuSMV: a symbolic model checker

- the first model checker based on BDDs
- open architecture for model checking, which can be reliably used for the verification of industrial designs, as a core for custom verification tools, as a testbed for formal verification techniques, and applied to other research areas. [2]

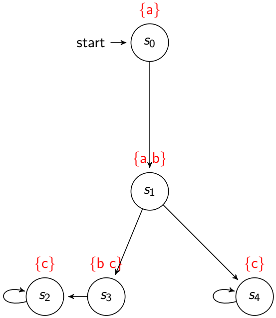
[2] nusmv.fbk.eu

# Application

- We will perform two tasks:

  1. We will first use the tool to encode transition systems and LTL and CTL formulas to be model checked.
  2. We will use the tool to perform bounded model checking.

# Transition systems in NuSMV




```
MODULE main
VAR
state :{s0,s1,s2,s3,s4};
ASSIGN
init(state) := {s0};
next(state) := case
state=s0 :  s1;
state=s1 :  {s3, s4};
state=s2 :  s2;
state=s3 :  s2;
state=s4 :  s4;
esac;
DEFINE
a := state=s0 | state=s1;
b := state=s1 | state=s3;
c := state=s2 | state=s3 | state=s4;
```

# Remark

- The NuSMV code is saved in a text file with extension `.smv`

```
TS1.smv
```

- Unlike SPIN, NuSMV can handle multiple initial states in the verification process. Hence, we only need to run the verification once.
- Can model check both `LTL` and `CTL` properties.

# NuSMV specification for LTL and CTL formulae

- An LTL formula consists of atomic proposition(s), boolean operator(s) and temporal operator(s)
- A CTL formula consists of atomic proposition(s), boolean operator(s), temporal operators and path quantifier(s)

| operator | math | NuSMV |
|---|---|---|
| not | $\neg$ | ! |
| and | $\wedge$ | & |
| or | $\vee$ | \| |
| implies | $\rightarrow$ | -> |
| equivalent | $\leftrightarrow$ | <-> |
| always | $\square$ | G |
| eventually | $\diamond$ | F |
| until | $U$ | U |
| next | $\bigcirc$ | X |
| for all | $\forall$ | A |
| exist | $\exists$ | E |

# Examples

- Some examples of the translation of LTL /CTL formula from mathematical notations to NuSMV commands

| LTL/CTL formula | NuSMV |
|---|---|
| $\Diamond\Box c$ | `FG c` |
| $\Box\Diamond c$ | `GF c` |
| $(\bigcirc\neg c) \rightarrow (\bigcirc\bigcirc c)$ | `(X ! c) -> (X X c)` |
| $\Box a$ | `G a` |
| $aU\Box(b \vee c)$ | `a U (G (b \| c))` |
| $(\bigcirc\bigcirc b)U(b \vee c)$ | `(X X b) U (b \| c)` |
| $\exists\Diamond\forall\Box c$ | `EF AG c` |
| $\forall\Box\exists\Diamond\neg c$ | `AG EF !c` |

# Preparing a NuSMV file `TS1.smv`

- Attach to the file `TS1.smv` the following code:

```
LTLSPEC F G a
CTLSPEC EF AG c
```

# Verification using NuSMV

- To verify the transition system against the given specification(s), execute the NuSMV with the parameter name of the smv file:

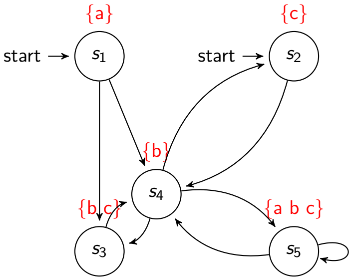
```
NuSMV TS1.smv
```

- NuSMV automatically generates a counter-example when a specification is not satisfied

# Exercise 1

- Verify the transition system used in example (`TS1.smv`) against the following properties:
  - $\Box \Diamond c$
  - $\Diamond \Box \neg b$
  - $\forall \Diamond \forall \Box c$
  - $\exists \Diamond (a \wedge b \wedge \forall \bigcirc b)$
  - $\forall \Box (b \rightarrow \forall \bigcirc c)$
  - $\forall \Box (a \leftrightarrow \neg c)$
- In each case, explain why the property was satisfied or not.

# Exercise 2




- Consider the transition system on the left
- Encode the transition system (e.g. `TS2.smv`)

# Exercise 2

- Verify the transition system (`TS2.smv`) against the following properties:
  - $\Diamond\Box c$
  - $\Box\Diamond c$
  - $(\bigcirc\neg c) \rightarrow \bigcirc(\bigcirc c)$
  - $a U(\Box(b \vee c))$
  - $\exists\Diamond(\forall\Box c)$
  - $\forall\bigcirc(\exists\Box b)$
- In each case, explain why the property was satisfied or not.

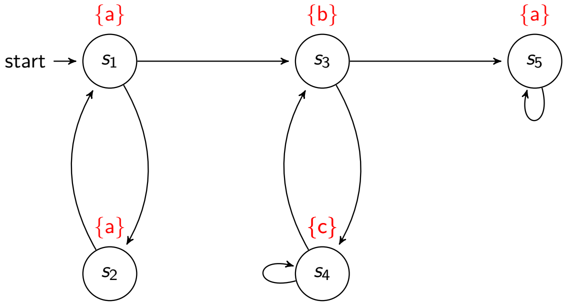
# Bounded Model Checking

Recall:

- employs a SAT solver for model checker
- focuses on counterexample generation (up to a certain length)

We will now perform bounded model checking on a transition system.

# Bounded Model Checking: Exercise




- Consider the above transition system
- Encode the transition system (e.g. `TS3.smv`)

# Bounded Model Checking: Exercise

- Verify the transition system (e.g. `TS3.smv`) against the following properties using bounded model checking
  - $\square \diamond a$
  - $\diamond\square(a \rightarrow (b \rightarrow \diamond c))$
  - $\square(a \wedge (\bigcirc c \rightarrow \diamond a))$
- To do bounded model checking:

```
NuSMV -bmc -bmc_length 2 TS3.smv
```

- Run bounded model checking with different maximum counterexample length and comment on result

# Bounded Model Checking: Extra Reading

Read the tutorial on bounded model checking using NuSMV found in the below link (pages 20 - 28):

http://nusmv.fbk.eu/NuSMV/tutorial/v26/tutorial.pdf

# Bonus Exercise

Determine whether the two formulas are equivalent:

$$\exists\Diamond(\exists\Box\ p) \text{ and } \exists\Box(\exists\Diamond\ p)$$