CURS 09.
NUSMV. EXEMPLU

Verificarea și validarea sistemelor soft
[28 Aprilie 2020]

Lector dr. Camelia Chisăliță-Creţu
Universitatea Babeş-Bolyai

Conţinut

- **Tool-uri utilizate pentru verificarea modelelor**
- **NuSMV**
  - **Comenzi**
  - **Exemplu**
  - **Operatori LTL. Operatori CTL**
  - **Proprietăţi**

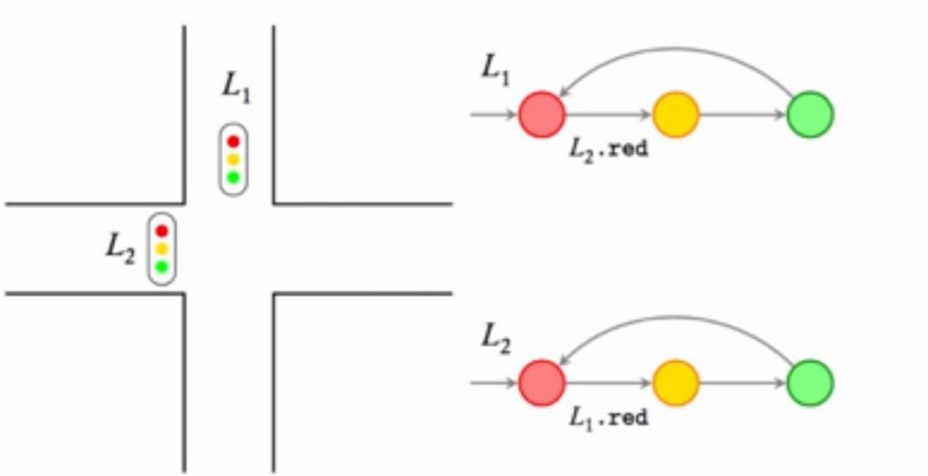
Tool-uri utilizate pentru verificarea modelelor

- **tool-uri folosite pentru verificarea modelelor:**
  - JSpin
    - limbajul Promela;
    - http://research.cs.queensu.ca/home/cisc853/readings/slides/tutorial2Slides.pdf;
  - NuSMV (New Symbolic Model Verifier);
    - dezvoltat de centre de cercetare din SUA şi Italia;
    - Foloseşte limbajul NuSMV pentru descrierea modelului şi a specificaţiilor (proprietăţi care trebuie verificate);
    - http://nusmv.fbk.eu/NuSMV/papers/sttt_j/html/paper.html;

NuSMV. Comenzi

- **NuSMV -int <filename.smv>**
- **go**
  - **read_model –i <filename.smv>**
  - **flatten_hierarchy**
  - **encode_variables sau build_variables**
  - **build_model**
  - **compute_fairness**
- **pick_state –i**
- **simulate –i –k <number of steps>**
- **check_ltlspec**
  - verificarea tuturor proprietăţilor descrise prin LTL în fişierul .svm;
- **check_ltlspec – p "<proprietate>"**
  - verifică proprietatea indicată presupunând că este descrisă prin LTL;
- **check_ctlspec**
  - verificarea tuturor proprietăţilor descrise prin CTL în fişierul .svm;
- **check_ctlspec – p "<proprietate>"**
  - verifică proprietatea indicată presupunând că este descrisă prin CTL.

NuSMV. Exemplu (1)



If a light is **red**, it can **stay red** for an **arbitrary period**

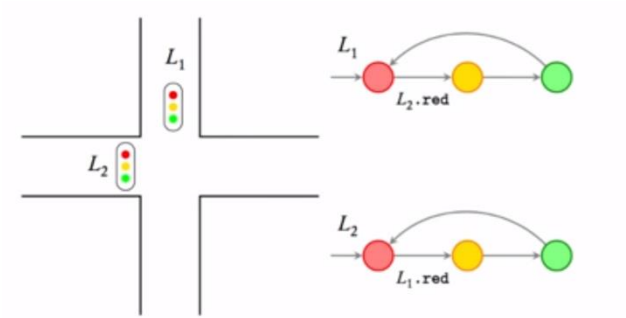
If it goes **yellow**, it should become **green within one cycle**

If it is **green**, it can **stay green** for an **arbitrary period**

NuSMV. Exemplu (2)

- MODULE light(other_state)
- VAR
  - state : {r, y, g};
- ASSIGN
  - init(state) := r;
  - next(state) :=
    - case
      - state = r & other_state=r : {r,y};
      - state = y : g;
      - state = g : {g,r};
      - TRUE : state;
    - esac;
- FAIRNESS running

- MODULE main
- VAR
  - tl1: process light(tl2.state);
  - tl2: process light(tl1.state);



If a light is **red**, it can **stay red** for an **arbitrary period**

If it goes **yellow**, it should become **green within one cycle**

If it is **green**, it can **stay green** for an **arbitrary period**

NuSMV. Operatori LTL. Operatori CTL

- **pentru descrierea proprietăţilor se folosesc LTL şi CTL; derivate din TL.**

Logica	Operator	Semnificaţie	NuSMV
propoziţiilor	¬ ∧∨→↔	Not, And, Or, Imply, Equivalence	!, &, |, ->, <->
TL	□	Always/ Globally	G
TL	◊	Eventually/ Finally/Future	F
LTL	○	Next/ NeXt State	X
LTL	U	Until/ Until	U
CTL	∃	Exists/ Exists	E
CTL	∀	For All/ For All	A

Proprietăţi (1)

- **Safety, Fairness**
  - *niciodata (cândva, în viitor) A si B vor fi simultan verde;*
    - **check_ltlspec -p “! F tl1.state=g & tl2.state=g“**

- **Liveness, Deadlock**
  - *semaforul va fi (cândva) verde;*
    - **check_ltlspec -p "F tl1.state=g”**
  - *(întotdeauna) culoarea roşie a semaforului nu se schimbă (imediat) în verde;*
    - **check_ltlspec -p "G tl1.state=r -> X(tl1.state!=g)"**
  - *culoarea roşie a semaforului se poate schimba în verde;*
    - **check_ltlspec -p "tl1.state=r -> X(tl1.state=r U (tl1.state=y & X(tl1.state=y U tl1.state=g)))"**

Proprietăţi (2)

- **Alte proprietăţi**
  - *(întotdeauna), dacă A este galben va deveni (cândva) roşu;*
    - **check_ltlspec -p "G tl1.state=y -> F tl1.state=r“**
  - *(întotdeauna) B nu va fi verde până când A nu va fi roşu;*
    - **check_ltlspec -p "G (tl2.state!=g U tl1.state=r)"**