universität innsbruck

SS 2024 lecture 10

Logic

Diana Gründlinger Aart Middeldorp Fabian Mitterwallner

Alexander Montag Johannes Niederhauser Daniel Rainer

Outline

1. **Summary of Previous Lecture**
2. **Symbolic Model Checking**
3. **Intermezzo**
4. **Linear-Time Temporal Logic (LTL)**
5. **Further Reading**

Definitions

boolean function f is

- **monotone** if $f(x_1,\dots,x_n) \leqslant f(y_1,\dots,y_n)$ for all $x_1 \leqslant y_1, \dots, x_n \leqslant y_n$
- **self-dual** if $f(x_1,\dots,x_n) = \overline{f(\overline{x}_1,\dots,\overline{x}_n)}$
- **affine** if $f(x_1,\dots,x_n) = c_0 \oplus c_1 x_1 \oplus \cdots \oplus c_n x_n$ for some $c_0,\dots,c_n \in \{0,1\}$

Theorem (Post's Adequacy Theorem)

set X of boolean functions is adequate if and only if following conditions hold:

1. $\exists\, f_1 \in X$ such that $f_1(0,\dots,0) \neq 0$
2. $\exists\, f_2 \in X$ such that $f_2(1,\dots,1) \neq 1$
3. $\exists\, f_3 \in X$ which is not monotone
4. $\exists\, f_4 \in X$ which is not self-dual
5. $\exists\, f_5 \in X$ which is not affine

Definitions

- CTL (computation tree logic) formulas are built from atoms, logical connectives, and temporal connectives AX, EX, AF, EF, AG, EG, AU, EU according to BNF grammar

$$\varphi \ ::= \bot \mid \top \mid p \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (\mathsf{AX}\, \varphi) \mid (\mathsf{EX}\, \varphi) \mid$$
$$(\mathsf{AF}\, \varphi) \mid (\mathsf{EF}\, \varphi) \mid (\mathsf{AG}\, \varphi) \mid (\mathsf{EG}\, \varphi) \mid \mathsf{A}[\varphi\, \mathsf{U}\, \varphi] \mid \mathsf{E}[\varphi\, \mathsf{U}\, \varphi]$$

- transition system (model) is triple $\mathcal{M} = (S, \rightarrow, L)$ with
  - set of states S
  - transition relation $\rightarrow \subseteq S \times S$ such that $\forall\, s \in S \ \ \exists\, t \in S$ with $s \rightarrow t$ ("no deadlock")
  - labelling function $L \colon S \rightarrow \mathcal{P}(\text{atoms})$
- satisfaction $\mathcal{M}, s \vDash \varphi$ of CTL formula φ in state $s \in S$ of model $\mathcal{M} = (S, \rightarrow, L)$ is defined by induction on φ

Definition

CTL formulas φ and ψ are semantically equivalent ($\varphi \equiv \psi$) if

$$\mathcal{M}, s \vDash \varphi \quad\Longleftrightarrow\quad \mathcal{M}, s \vDash \psi$$

for all models $\mathcal{M} = (S, \rightarrow, L)$ and states $s \in S$

Theorem

$$\neg\,\mathsf{AF}\,\varphi \equiv \mathsf{EG}\,\neg\varphi \qquad\qquad \mathsf{AF}\,\varphi \equiv \mathsf{A}[\top\,\mathsf{U}\,\varphi]$$

$$\neg\,\mathsf{EF}\,\varphi \equiv \mathsf{AG}\,\neg\varphi \qquad\qquad \mathsf{EF}\,\varphi \equiv \mathsf{E}[\top\,\mathsf{U}\,\varphi]$$

$$\neg\,\mathsf{AX}\,\varphi \equiv \mathsf{EX}\,\neg\varphi \qquad\qquad \mathsf{A}[\varphi\,\mathsf{U}\,\psi] \equiv \neg(\mathsf{E}[\neg\psi\,\mathsf{U}\,(\neg\varphi \wedge \neg\psi)] \vee \mathsf{EG}\,\neg\psi)$$

Theorem

satisfaction of CTL formulas in finite models is decidable

CTL Model Checking Algorithm

input: • model $\mathcal{M} = (S, \rightarrow, L)$ and CTL formula φ

output: • $\{s \in S \mid \mathcal{M}, s \vDash \varphi\}$

label each state $s \in S$ by those subformulas of φ that are satisfied in s

p label s $\iff$ $p \in L(s)$

$\neg\varphi$ label s $\iff$ s is not labelled with φ

$\varphi \wedge \psi$ label s $\iff$ s is labelled with both φ and ψ

$\mathsf{EX}\,\varphi$ label s $\iff$ t is labelled with φ for some t with $s \rightarrow t$

$\mathsf{EG}\,\varphi$
① label every s that is labelled with φ
② remove label from s $\iff$ t is not labelled with $\mathsf{EG}\,\varphi$ for all t with $s \rightarrow t$
③ repeat ② until no change

$\mathsf{E}[\varphi\,\mathsf{U}\,\psi]$ label s $\iff$
① s is labelled with ψ
② s is labelled with φ and t with $\mathsf{E}[\varphi\,\mathsf{U}\,\psi]$ for some t with $s \rightarrow t$
③ repeat ② until no change

Part I: Propositional Logic

algebraic normal forms, binary decision diagrams, conjunctive normal forms, DPLL, Horn formulas, natural deduction, Post's adequacy theorem, resolution, SAT, semantics, sorting networks, soundness and completeness, syntax, Tseitin's transformation

Part II: Predicate Logic

natural deduction, quantifier equivalences, resolution, semantics, Skolemization, syntax, undecidability, unification

Part III: Model Checking

adequacy, branching-time temporal logic, CTL*, fairness, linear-time temporal logic, model checking algorithms, symbolic model checking

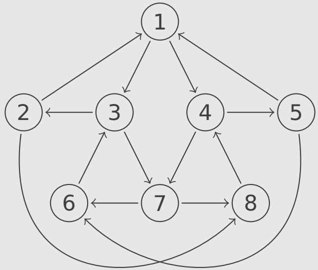
Outline

symbolic model checking = (CTL) model checking with BDDs

Questions

- how to represent sets of states?
- how to represent transition relation?
- how to implement model checking algorithm?

Example



model $\mathcal{M} = (S, \rightarrow, L)$

$$S = \{1,2,3,4,5,6,7,8\}$$

$L(1) = \{I_A, I_B\}$ $\quad$ $L(5) = \{I_A, P_B\}$

$L(2) = \{P_A, I_B\}$ $\quad$ $L(6) = \{R_A, P_B\}$

$L(3) = \{R_A, I_B\}$ $\quad$ $L(7) = \{R_A, R_B\}$

$L(4) = \{I_A, R_B\}$ $\quad$ $L(8) = \{P_A, R_B\}$

- 8 states require 3 boolean variables

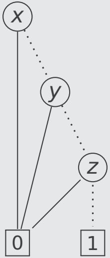
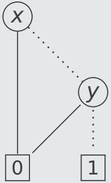
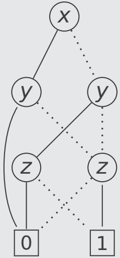
state	x	y	z		state	x	y	z	
1	0	0	0	$\overline{x}\,\overline{y}\,\overline{z}$	5	1	0	0	$x\overline{y}\,\overline{z}$
2	0	0	1	$\overline{x}\,\overline{y}z$	6	1	0	1	$x\overline{y}z$
3	0	1	0	$\overline{x}y\overline{z}$	7	1	1	0	$xy\overline{z}$
4	0	1	1	$\overline{x}yz$	8	1	1	1	xyz

Example (cont'd)

state		state		state		state	
1	$\overline{x}\,\overline{y}\,\overline{z}$	3	$\overline{x}\,y\,\overline{z}$	5	$x\,\overline{y}\,\overline{z}$	7	$x\,y\,\overline{z}$
2	$\overline{x}\,\overline{y}\,z$	4	$\overline{x}\,y\,z$	6	$x\,\overline{y}\,z$	8	$x\,y\,z$

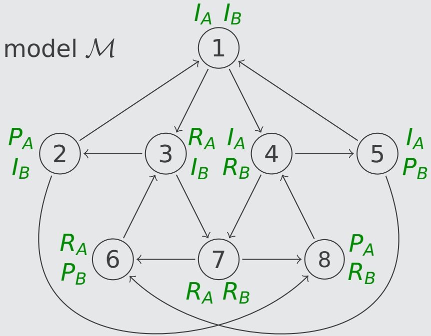
variable ordering

$[x, y, z]$

set of states	$\{1\}$	$\{1,2\}$	$\varnothing$	$\{2,3,6\}$
boolean function	$\overline{x}\,\overline{y}\,\overline{z}$	$\overline{x}\,\overline{y}$	0	$\overline{y}z + \overline{x}y\overline{z}$
reduced OBDD				

Example (cont'd)

state		state		state		state	
1	$\overline{x}\,\overline{y}\,\overline{z}$	3	$\overline{x}\,y\,\overline{z}$	5	$x\,\overline{y}\,\overline{z}$	7	$x\,y\,\overline{z}$
2	$\overline{x}\,\overline{y}\,z$	4	$\overline{x}\,y\,z$	6	$x\,\overline{y}\,z$	8	$x\,y\,z$



transition relation

$$
\begin{aligned}
&\overline{x}\,\overline{y}\left(\overline{z}\,\overline{x}'y' + z(\overline{x}'\,\overline{y}'\,\overline{z}' + x'y'z')\right)\\
+\,&\overline{x}y\left(\overline{z}(\overline{x}'\,\overline{y}'z' + x'y'\overline{z}') + zx'\overline{z}'\right)\\
+\,&x\overline{y}\left(\overline{z}(\overline{x}'\,\overline{y}'\,\overline{z}' + x'\overline{y}'z') + z\overline{x}'y'\overline{z}'\right)\\
+\,&xy\left(\overline{z}x'z' + z\overline{x}'y'z'\right)
\end{aligned}
$$

reduced OBDD with variable ordering $[x, y, z, x', y', z']$ has 24 nodes ($B_{\to}$)

Definition

model $\mathcal{M} = (S, \rightarrow, L) \quad X \subseteq S$

- $[\![\varphi]\!] = \{s \in S \mid \mathcal{M}, s \vDash \varphi\}$
- $\mathrm{pre}_\forall(X) = \{s \in S \mid t \in X \text{ for all } t \text{ with } s \rightarrow t\}$
- $\mathrm{pre}_\exists(X) = \{s \in S \mid s \rightarrow t \text{ for some } t \in X\}$

Lemma

$$\begin{aligned}
[\![\top]\!] &= S \\
[\![\bot]\!] &= \varnothing \\
[\![\neg\varphi]\!] &= S - [\![\varphi]\!] \\
[\![\varphi \wedge \psi]\!] &= [\![\varphi]\!] \cap [\![\psi]\!] \\
[\![\varphi \vee \psi]\!] &= [\![\varphi]\!] \cup [\![\psi]\!] \\
[\![\varphi \rightarrow \psi]\!] &= (S - [\![\varphi]\!]) \cup [\![\psi]\!]
\end{aligned}$$

$$\begin{aligned}
[\![p]\!] &= \{s \in S \mid p \in L(s)\} \\
[\![\mathrm{AX}\,\varphi]\!] &= \mathrm{pre}_\forall([\![\varphi]\!]) \\
[\![\mathrm{EX}\,\varphi]\!] &= \mathrm{pre}_\exists([\![\varphi]\!])
\end{aligned}$$

$$\mathrm{pre}_\forall(X) = S - \mathrm{pre}_\exists(S - X)$$

Symbolic Model Checking Operations

required operations		BDD representation
complement	$S - X$	$\text{apply}(\oplus, B_S, B_X)$
union	$X \cup Y$	$\text{apply}(+, B_X, B_Y)$
intersection	$X \cap Y$	$\text{apply}(\cdot, B_X, B_Y)$
	$\text{pre}_\exists(X)$	$\text{exists}(x'_1, \cdots (\text{exists}(x'_n, \text{apply}(\cdot, B_\to, \underbrace{B_{X'}}_{\text{replace } x_1, \dots, x_n \text{ by } x'_1, \dots, x'_n \text{ in } B_X}))) \cdots)$

$$\text{exists}(x', B) = \text{apply}(+, \text{restrict}(0, x', B), \text{restrict}(1, x', B))$$

Lemma

$$[\![\mathsf{AF}\,\varphi]\!] = [\![\varphi]\!] \cup \mathsf{pre}_\forall([\![\mathsf{AF}\,\varphi]\!]) \qquad [\![\mathsf{EF}\,\varphi]\!] = [\![\varphi]\!] \cup \mathsf{pre}_\exists([\![\mathsf{EF}\,\varphi]\!])$$

$$[\![\mathsf{AG}\,\varphi]\!] = [\![\varphi]\!] \cap \mathsf{pre}_\forall([\![\mathsf{AG}\,\varphi]\!]) \qquad [\![\mathsf{EG}\,\varphi]\!] = [\![\varphi]\!] \cap \mathsf{pre}_\exists([\![\mathsf{EG}\,\varphi]\!])$$

$$[\![\mathsf{A}[\varphi\,\mathsf{U}\,\psi]]\!] = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \mathsf{pre}_\forall([\![\mathsf{A}[\varphi\,\mathsf{U}\,\psi]]\!])) \qquad [\![\mathsf{E}[\varphi\,\mathsf{U}\,\psi]]\!] = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \mathsf{pre}_\exists([\![\mathsf{E}[\varphi\,\mathsf{U}\,\psi]]\!]))$$

Remark

- $[\![\mathsf{AF}\,\varphi]\!]$ is least fixed point of function $F_{\mathsf{AF}}(X) = [\![\varphi]\!] \cup \mathsf{pre}_\forall(X)$
- $[\![\mathsf{EG}\,\varphi]\!]$ is greatest fixed point of function $F_{\mathsf{EG}}(X) = [\![\varphi]\!] \cap \mathsf{pre}_\exists(X)$

Theorem (Knaster–Tarski)

every monotone function $F\colon \mathcal{P}(S) \to \mathcal{P}(S)$ with $|S| = n$ admits

- least fixed point $\mu F = F^n(\varnothing)$
- greatest fixed point $\nu F = F^n(S)$

Theorem (Knaster–Tarski)

every monotone function $F\colon \mathcal{P}(S) \to \mathcal{P}(S)$ with $|S| = n$ admits

- least fixed point $\mu F = F^n(\varnothing)$
- greatest fixed point $\nu F = F^n(S)$

function $F\colon \mathcal{P}(S) \to \mathcal{P}(S)$ is monotone if $F(X) \subseteq F(Y)$ whenever $X \subseteq Y \subseteq S$

Proof

see overlay version of slides

Definition

function $F_{\mathsf{AF}} : \mathcal{P}(S) \to \mathcal{P}(S)$: $\quad F_{\mathsf{AF}}(X) = [\![\varphi]\!] \cup \mathsf{pre}_{\forall}(X)$

Example

$$\varphi = I_B \qquad X = \varnothing$$

$$F_{\mathsf{AF}}(X) = [\![\varphi]\!] = \{1,2,3\}$$

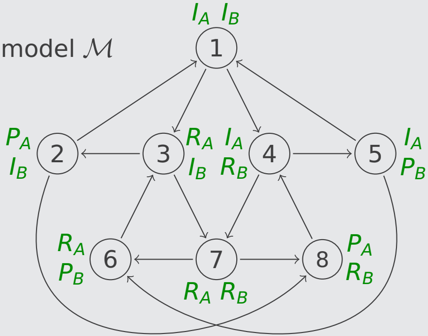
$$F^2_{\mathsf{AF}}(X) = F_{\mathsf{AF}}(F_{\mathsf{AF}}(X)) = \{1,2,3\} \cup \{6\}$$

$$F^3_{\mathsf{AF}}(X) = \{1,2,3\} \cup \{5,6\}$$

$$F^4_{\mathsf{AF}}(X) = \{1,2,3\} \cup \{5,6\}$$

$$[\![\mathsf{AF}\, I_B]\!] = \{1,2,3,5,6\}$$

model $\mathcal{M}$



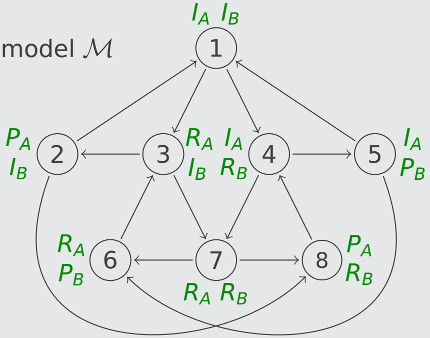
Definition

function $F_{\mathrm{EG}} \colon \mathcal{P}(S) \to \mathcal{P}(S)$: $\quad F_{\mathrm{EG}}(X) = [\![\varphi]\!] \cap \mathrm{pre}_{\exists}(X)$

Example

$$
\begin{aligned}
\varphi &= P_A \vee I_B \\
X &= \{1,2,3,4,5,6,7,8\} = \mathrm{pre}_{\exists}(X) \\
F_{\mathrm{EG}}(X) &= [\![\varphi]\!] = \{1,2,3,8\} \\
F^2_{\mathrm{EG}}(X) &= \{1,2,3,8\} \cap \{1,2,3,5,6,7\} \\
F^3_{\mathrm{EG}}(X) &= \{1,2,3,8\} \cap \{1,2,3,5,6\} \\
[\![\mathrm{EG}\,(P_A \vee I_B)]\!] &= \{1,2,3\}
\end{aligned}
$$

model $\mathcal{M}$



Definition

function $F_{\mathsf{EU}} : \mathcal{P}(S) \to \mathcal{P}(S)$: $\quad F_{\mathsf{EU}}(X) = [\![\psi]\!] \cup ([\![\varphi]\!] \cap \mathsf{pre}_{\exists}(X))$

Lemma

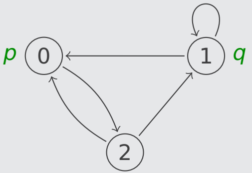
$[\![\mathsf{E}[\varphi \,\mathsf{U}\, \psi]]\!]$ is least fixed point of monotone function F_{EU}

Algorithm

```
W := [[φ]];
X := ∅;
Y := [[ψ]];
repeat until X = Y
     X := Y;
     Y := Y ∪ (W ∩ pre∃(Y))
return Y
```

Example (Huth and Ryan, Exercise 6.12.2(a))

model $\mathcal{M} = (S, \rightarrow, L)$



state	x	y
0	1	0
1	0	1
2	0	0
–	1	1

$\rightarrow$: $\overline{x}\,\overline{x}'y' + \overline{x}x'\overline{y}' + x\overline{x}'\overline{y}\,\overline{y}'$

p: $x\overline{y}$ $\quad$ $S, \top$: $x\overline{y} + \overline{x}y + \overline{x}\,\overline{y} = \overline{x} + \overline{y}$

q: $\overline{x}y$ $\quad$ $\neg p \wedge q$: $((\overline{x} + \overline{y}) \oplus x\overline{y}) \cdot \overline{x}y = \overline{x}y$

W: $\overline{x} + \overline{y}$

$$\mathrm{AG}(p \vee \neg q) \equiv \neg \mathrm{E}[\top \,\mathrm{U}\, \neg p \wedge q]$$

```
W := ⟦⊤⟧;
X := ∅;
Y := ⟦¬p ∧ q⟧;
repeat until X = Y
    X := Y;
    Y := Y ∪ (W ∩ pre∃(Y))
return Y
```

X_0 0 $\quad$ X_1 $\overline{x}y$ $\quad$ X_2 $\overline{x}$ $\quad$ X_3 $\overline{x} + \overline{y}$

Y_0 $\overline{x}y$ $\quad$ Y_1 $\overline{x}$ $\quad$ Y_2 $\overline{x} + \overline{y}$ $\quad$ Y_3 $\overline{x} + \overline{y}$ $\quad$ $X_3 = Y_3$

$\mathrm{E}[\top \,\mathrm{U}\, \neg p \wedge q]$: $\overline{x} + \overline{y}$

$\mathrm{AG}(p \vee \neg q)$: $(\overline{x} + \overline{y}) \oplus (\overline{x} + \overline{y}) = 0$

Outline

Question

Which of the following statements about symbolic model checking are true ?

A For a model with 2 states the reduced BDD $B_{\rightarrow}$ has at most 5 nodes.

B The set $[\![p \vee \neg p]\!]$ corresponds to the reduced BDD $\boxed{0}$.

C Every monotone function $F\colon \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ with $|S| = n$ admits a least fixed point $\mu F = F^n(S)$.

D $[\![\varphi \rightarrow \bot]\!] = (S - [\![\varphi]\!])$

Outline

Definitions

- LTL (linear-time temporal logic) formulas are built from
  - atoms $p,\ q,\ r,\ p_1,\ p_2,\ \ldots$
  - logical connectives $\bot,\ \top,\ \neg,\ \wedge,\ \vee,\ \rightarrow$
  - temporal connectives $\mathsf{X},\ \mathsf{F},\ \mathsf{G},\ \mathsf{U},\ \mathsf{W},\ \mathsf{R}$

  according to following BNF grammar:

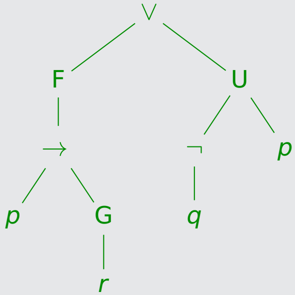
$$\varphi \ ::= \bot \mid \top \mid p \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid$$
$$(\mathsf{X}\,\varphi) \mid (\mathsf{F}\,\varphi) \mid (\mathsf{G}\,\varphi) \mid (\varphi\,\mathsf{U}\,\varphi) \mid (\varphi\,\mathsf{W}\,\varphi) \mid (\varphi\,\mathsf{R}\,\varphi)$$

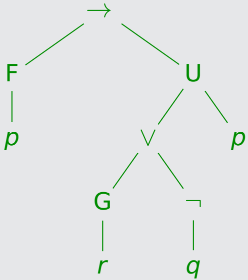
- notational conventions:
  - binding precedence $\neg,\ \mathsf{X},\ \mathsf{F},\ \mathsf{G} \ > \ \mathsf{U},\ \mathsf{W},\ \mathsf{R} \ > \ \wedge,\ \vee \ > \ \rightarrow$
  - omit outer parentheses
  - $\rightarrow,\ \wedge,\ \vee$ are right-associative

Example

formula: $\mathsf{F}\,(p \to \mathsf{G}\,r) \vee \neg q\;\mathsf{U}\;p$ $\qquad$ $\mathsf{F}\,p \to (\mathsf{G}\,r \vee \neg q)\;\mathsf{U}\;p$

parse tree





X	next state	F	$\exists$ future state	W	weak until
U	until	G	$\forall$ states globally	R	release

Outline

Definition

- path in model $\mathcal{M} = (S, \to, L)$ is infinite sequence $s_1 \to s_2 \to \cdots$
- $\forall$ paths $\pi = s_1 \to s_2 \to \cdots \quad \forall\, i \geqslant 1 \quad \pi^i = s_i \to s_{i+1} \to \cdots$

Definition

satisfaction of LTL formula φ with respect to path $\pi = s_1 \to s_2 \to \cdots$ in model $\mathcal{M} = (S, \to, L)$

$$\pi \vDash \varphi$$

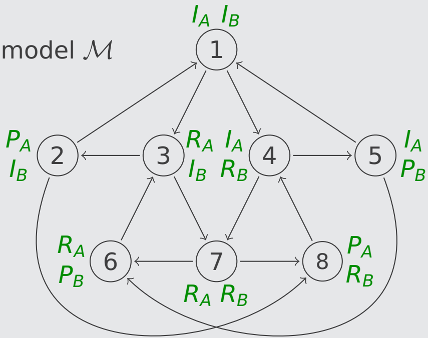
is defined by induction on φ:

$$\pi \vDash \top \qquad \pi \nvDash \bot \qquad \pi \vDash \varphi \wedge \psi \iff \pi \vDash \varphi \text{ and } \pi \vDash \psi$$

$$\pi \vDash p \iff p \in L(s_1) \qquad \pi \vDash \varphi \vee \psi \iff \pi \vDash \varphi \text{ or } \pi \vDash \psi$$

$$\pi \vDash \neg\varphi \iff \pi \nvDash \varphi \qquad \pi \vDash \varphi \to \psi \iff \pi \nvDash \varphi \text{ or } \pi \vDash \psi$$

Example



$$\pi_1 = 1 \to 3 \to 2 \to 1 \to 3 \to 2 \to \cdots$$

$$\pi_2 = 7 \to 6 \to 3 \to 7 \to 6 \to 3 \to \cdots$$

special notation for infinite paths:

$$\pi_1 = (1\,3\,2)^\omega \qquad \pi_2 = (7\,6\,3)^\omega$$

$\pi_1 \vDash I_A$ $\qquad$ $\pi_1 \nvDash R_A \wedge I_B$ $\qquad$ $\pi_1 \nvDash I_B \to P_A \vee R_B$

$\pi_2 \nvDash I_A$ $\qquad$ $\pi_2^6 \vDash R_A \wedge I_B$ $\qquad$ $\pi_2 \vDash I_B \to P_A \vee R_B$

Definition

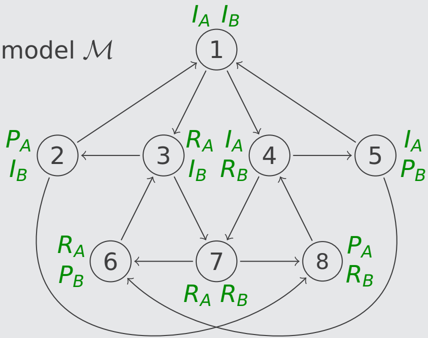
satisfaction of LTL formula φ with respect to path $\pi = s_1 \rightarrow s_2 \rightarrow \cdots$ in model $\mathcal{M} = (S, \rightarrow, L)$

$$\pi \vDash \varphi$$

is defined by induction on φ:

$$\begin{aligned}
\pi \vDash \mathsf{X}\,\varphi &\iff \pi^2 \vDash \varphi \\
\pi \vDash \mathsf{F}\,\varphi &\iff \exists\, i \geqslant 1 \;\; \pi^i \vDash \varphi \\
\pi \vDash \mathsf{G}\,\varphi &\iff \forall\, i \geqslant 1 \;\; \pi^i \vDash \varphi \\
\pi \vDash \varphi\,\mathsf{U}\,\psi &\iff \exists\, i \geqslant 1 \;\; \pi^i \vDash \psi \;\text{ and }\; \forall\, j < i \;\; \pi^j \vDash \varphi \\
\pi \vDash \varphi\,\mathsf{W}\,\psi &\iff \big(\exists\, i \geqslant 1 \;\; \pi^i \vDash \psi \;\text{ and }\; \forall\, j < i \;\; \pi^j \vDash \varphi\big) \;\text{ or }\; \forall\, i \geqslant 1 \;\; \pi^i \vDash \varphi \\
\pi \vDash \varphi\,\mathsf{R}\,\psi &\iff \big(\exists\, i \geqslant 1 \;\; \pi^i \vDash \varphi \;\text{ and }\; \forall\, j \leqslant i \;\; \pi^j \vDash \psi\big) \;\text{ or }\; \forall\, i \geqslant 1 \;\; \pi^i \vDash \psi
\end{aligned}$$

Example



$\pi_1 = (1\,3\,2)^\omega \qquad \pi_2 = (7\,6\,3)^\omega$

$\pi_1 \vDash \mathrm{X}\,(R_A \vee R_B)$ $\qquad \pi_1 \vDash \mathrm{F}\,P_A$ $\qquad \pi_1 \nvDash \mathrm{X}\,\mathrm{X}\,P_B$

$\pi_2 \nvDash \mathrm{F}\,P_A$ $\qquad \pi_2 \vDash \mathrm{G}\,\neg I_A$ $\qquad \pi_2 \vDash \mathrm{G}\,\mathrm{F}\,P_B$

$\pi_1 \nvDash I_A\,\mathrm{U}\,P_A$ $\qquad \pi_2 \vDash \neg I_A\,\mathrm{W}\,P_A$ $\qquad \pi_2 \nvDash P_B\,\mathrm{R}\,R_B$

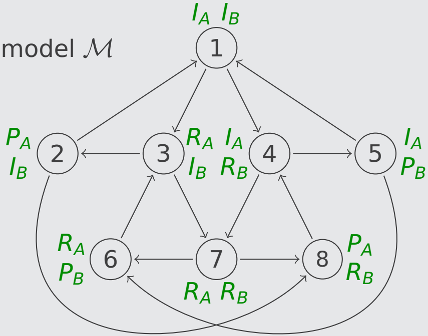
Definition

model $\mathcal{M} = (S, \rightarrow, L)$, state $s \in S$, LTL formula φ

$\mathcal{M}, s \vDash \varphi \iff \forall$ paths $\pi = s \rightarrow \cdots \quad \pi \vDash \varphi$ "formula φ holds in state s of model $\mathcal{M}$"

Example

model $\mathcal{M}$



$\mathcal{M}, 1 \nvDash \mathsf{G}\,(R_A \rightarrow \mathsf{F}\,P_A)$

$\mathcal{M}, 4 \nvDash \neg(R_B \,\mathsf{U}\, P_B)$

$\mathcal{M}, 4 \nvDash R_B \,\mathsf{U}\, P_B$

$\mathcal{M}, 6 \vDash \mathsf{X}\,(\mathsf{F}\,I_B \wedge ((\mathsf{X}\,\neg P_B)\,\mathsf{R}\,R_A))$

Definition

LTL formulas φ and ψ are semantically equivalent $(\varphi \equiv \psi)$ if

$\forall$ models $\mathcal{M} = (S, \to, L)$

$\forall$ paths π in $\mathcal{M}$

$$\pi \vDash \varphi \iff \pi \vDash \psi$$

Theorem

$$\neg \mathsf{X}\, \varphi \equiv \mathsf{X} \neg \varphi \qquad \varphi \,\mathsf{U}\, \psi \equiv \neg(\neg \psi \,\mathsf{U}\, (\neg \varphi \wedge \neg \psi)) \wedge \mathsf{F}\, \psi$$

$$\neg \mathsf{F}\, \varphi \equiv \mathsf{G} \neg \varphi \qquad \mathsf{F}\,(\varphi \vee \psi) \equiv \mathsf{F}\, \varphi \vee \mathsf{F}\, \psi$$

$$\neg \mathsf{G}\, \varphi \equiv \mathsf{F} \neg \varphi \qquad \mathsf{G}\,(\varphi \wedge \psi) \equiv \mathsf{G}\, \varphi \wedge \mathsf{G}\, \psi$$

$$\neg(\varphi \,\mathsf{U}\, \psi) \equiv \neg \varphi \,\mathsf{R}\, \neg \psi \qquad \mathsf{F}\, \varphi \equiv \top \,\mathsf{U}\, \varphi$$

$$\neg(\varphi \,\mathsf{R}\, \psi) \equiv \neg \varphi \,\mathsf{U}\, \neg \psi \qquad \mathsf{G}\, \varphi \equiv \bot \,\mathsf{R}\, \varphi$$

$$\varphi \,\mathsf{U}\, \psi \equiv \varphi \,\mathsf{W}\, \psi \wedge \mathsf{F}\, \psi \qquad \varphi \,\mathsf{W}\, \psi \equiv \psi \,\mathsf{R}\, (\varphi \vee \psi)$$

$$\varphi \,\mathsf{W}\, \psi \equiv \varphi \,\mathsf{U}\, \psi \vee \mathsf{G}\, \varphi \qquad \varphi \,\mathsf{R}\, \psi \equiv \psi \,\mathsf{W}\, (\varphi \wedge \psi)$$

Theorem

$$\varphi \mathsf{U} \psi \equiv \neg(\neg\psi \mathsf{U} (\neg\varphi \wedge \neg\psi)) \wedge \mathsf{F} \psi$$

see overlay version for proof

Outline

Mutual Exclusion

- concurrent processes sharing resource
- identify critical sections (including access to shared resource) in each process' code
- at most one process can be in its critical section at any time

desired:

protocol for determining which process is allowed to enter its critical section at which time

expected properties:

safety only one process is in its critical section at any time

liveness whenever process requests to enter its critical section, it will eventually be permitted to do so

non-blocking each process can always request to enter its critical section

Mutual Exclusion (first modeling attempt)

- two processes have three states each:

  (n) non-critical state (t) trying to enter critical state (c) critical state

- each process undergoes transitions in cycle $n_i \to t_i \to c_i \to n_i \to \cdots$ $(n_i t_i c_i)^\omega$

- asynchronous interleaving

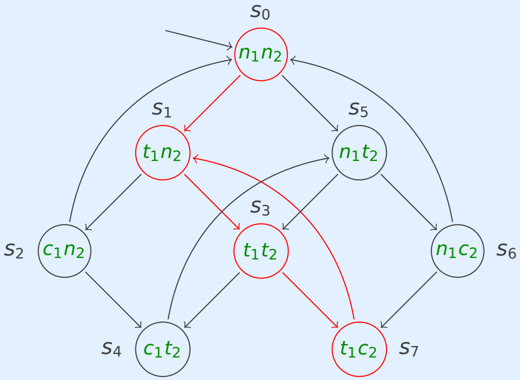
- model (protocol):

- safety: $\mathsf{G}\,\neg(c_1 \wedge c_2)$ ✓

- liveness: $\mathsf{G}\,(t_1 \to \mathsf{F}\,c_1)$ ✗

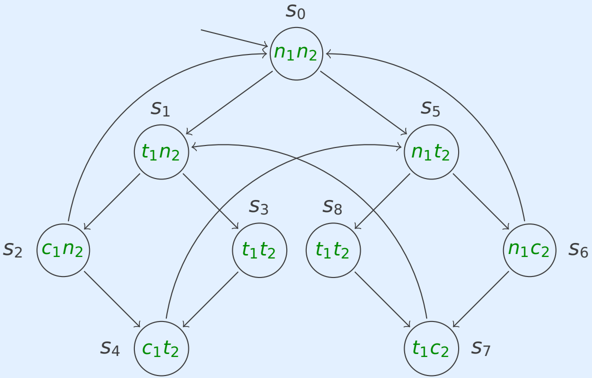
- non-blocking holds but is not expressible in LTL

  $\mathsf{AG}\,(n_1 \to \mathsf{EX}\,t_1)$ in CTL



Mutual Exclusion (second modeling attempt)

- safety: $\mathsf{G}\,\neg(c_1 \wedge c_2)$ ✓
- liveness: $\mathsf{G}\,(t_1 \rightarrow \mathsf{F}\,c_1)$ ✓



NuSMV (New Symbolic Model Verifier)

provides language for describing models and checks satisfaction of LTL and CTL formulas

Mutual Exclusion Protocol in NuSMV

```
MODULE main
VAR
  pr1 : process prc ( pr2.st, turn, FALSE ) ;
  pr2 : process prc ( pr1.st, turn, TRUE ) ;
  turn : boolean ;
ASSIGN
  init ( turn ) := FALSE ;

LTLSPEC G ! (( pr1.st = c ) & ( pr2.st = c ))   -- safety
LTLSPEC G (( pr1.st = t ) -> F ( pr1.st = c ))  -- liveness
LTLSPEC G (( pr2.st = t ) -> F ( pr2.st = c ))  -- liveness

MODULE prc ( other-st, turn, myturn )
VAR st : { n, t, c } ;
ASSIGN
  init ( st ) := n ;
  next ( st ) := case
    ( st = n )                        : { n, t } ;
    ( st = t ) & ( other-st = n ) : c ;
    ( st = t ) & ( other-st = t ) : c ;
    ( st = c )                        : st ;
    TRUE                              : st ;
  esac ;
  next ( turn ) := case
    turn = myturn & st = c          : ! turn ;
    TRUE                              : turn ;
  esac ;

FAIRNESS running
FAIRNESS ! ( st = c )
```

Outline

Huth and Ryan

- Section 3.1
- Section 3.2
- Section 3.3
- Section 3.7
- Section 6.3

Model Checking Tools

- NuSMV
- Spin

Important Concepts

- $[\![\varphi]\!]$
- F
- fixed point
- G
- greatest fixed point
- Knaster–Tarski
- least fixed point
- linear-time temporal logic
- liveness
- LTL
- non-blocking
- path
- $\text{pre}_\forall$
- $\text{pre}_\exists$
- R
- safety
- symbolic model checking
- U
- W
- X

homework for June 6

next week (June 3): online evaluation in presence $\Longrightarrow$ bring device