



Logic

Diana Gründlinger Aart Middeldorp Fabian Mitterwallner
Alexander Montag Johannes Niederhauser Daniel Rainer

Definitions

boolean function f is

- ▶ **monotone** if $f(x_1, \dots, x_n) \leq f(y_1, \dots, y_n)$ for all $x_1 \leq y_1, \dots, x_n \leq y_n$
- ▶ **self-dual** if $f(x_1, \dots, x_n) = \overline{f(\overline{x}_1, \dots, \overline{x}_n)}$
- ▶ **affine** if $f(x_1, \dots, x_n) = c_0 \oplus c_1 x_1 \oplus \dots \oplus c_n x_n$ 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:

- ❶ $\exists f_1 \in X$ such that $f_1(0, \dots, 0) \neq 0$
- ❷ $\exists f_2 \in X$ such that $f_2(1, \dots, 1) \neq 1$
- ❸ $\exists f_3 \in X$ which is not monotone
- ❹ $\exists f_4 \in X$ which is not self-dual
- ❺ $\exists f_5 \in X$ which is not affine

Outline

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

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

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

- ▶ **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 $L: S \rightarrow \mathcal{P}(\text{atoms})$
- ▶ **satisfaction** $\mathcal{M}, s \models \varphi$ 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

$$\mathcal{M}, s \models \varphi \iff \mathcal{M}, s \models \psi$$

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

Theorem

$$\begin{array}{ll} \neg AF \varphi \equiv EG \neg \varphi & AF \varphi \equiv A[\top U \varphi] \\ \neg EF \varphi \equiv AG \neg \varphi & EF \varphi \equiv E[\top U \varphi] \\ \neg AX \varphi \equiv EX \neg \varphi & A[\varphi U \psi] \equiv \neg(E[\neg \psi U (\neg \varphi \wedge \neg \psi)] \vee EG \neg \psi) \end{array}$$

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

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

EX φ label $s \iff t$ is labelled with φ for some t with $s \rightarrow t$

EG φ ① label every s that is labelled with φ

② remove label from $s \iff t$ is not labelled with EG φ for all t with $s \rightarrow t$

③ repeat ② until no change

$E[\varphi U \psi]$ label $s \iff$ ① s is labelled with ψ

② s is labelled with φ and t with $E[\varphi U \psi]$ 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

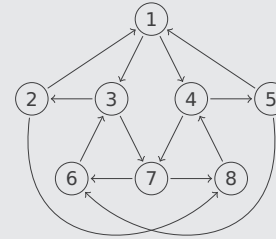
1. Summary of Previous Lecture
2. Symbolic Model Checking
3. Intermezzo
4. Linear-Time Temporal Logic (LTL)
5. Further Reading

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

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

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

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

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

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

- ▶ 8 states require 3 boolean variables

state	x	y	z	state	x	y	z
1	0	0	0	5	1	0	0
2	0	0	1	6	1	0	1
3	0	1	0	7	1	1	0
4	0	1	1	8	1	1	1

Example (cont'd)

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

variable ordering
[x, y, z]

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

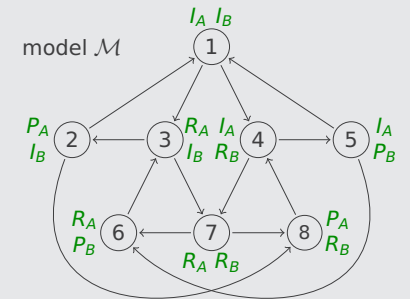
Example (cont'd)

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

transition relation

$$\begin{aligned}
 &\bar{x}\bar{y}(\bar{z}\bar{x}'y' + z(\bar{x}'\bar{y}'\bar{z}' + x'y'z')) \\
 &+ \bar{x}y(\bar{z}(\bar{x}'\bar{y}'z' + x'y'z') + zx'\bar{z}') \\
 &+ x\bar{y}(\bar{z}(\bar{x}'\bar{y}'z' + x'y'z') + z\bar{x}'y'\bar{z}') \\
 &+ xy(\bar{z}x'z' + z\bar{x}'y'z')
 \end{aligned}$$

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



Definition

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

- ▶ $\llbracket \varphi \rrbracket = \{s \in S \mid \mathcal{M}, s \models \varphi\}$
- ▶ $\text{pre}_\forall(X) = \{s \in S \mid t \in X \text{ for all } t \text{ with } s \rightarrow t\}$
- ▶ $\text{pre}_\exists(X) = \{s \in S \mid s \rightarrow t \text{ for some } t \in X\}$

Lemma

$$\begin{aligned} \llbracket \top \rrbracket &= S & \llbracket p \rrbracket &= \{s \in S \mid p \in L(s)\} \\ \llbracket \perp \rrbracket &= \emptyset & \llbracket AX \varphi \rrbracket &= \text{pre}_\forall(\llbracket \varphi \rrbracket) \\ \llbracket \neg \varphi \rrbracket &= S - \llbracket \varphi \rrbracket & \llbracket EX \varphi \rrbracket &= \text{pre}_\exists(\llbracket \varphi \rrbracket) \\ \llbracket \varphi \wedge \psi \rrbracket &= \llbracket \varphi \rrbracket \cap \llbracket \psi \rrbracket & & \\ \llbracket \varphi \vee \psi \rrbracket &= \llbracket \varphi \rrbracket \cup \llbracket \psi \rrbracket & & \\ \llbracket \varphi \rightarrow \psi \rrbracket &= (S - \llbracket \varphi \rrbracket) \cup \llbracket \psi \rrbracket & \text{pre}_\forall(X) &= S - \text{pre}_\exists(S - X) \end{aligned}$$

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	$X \cap Y$	$\text{apply}(\cdot, B_X, B_Y)$
$\text{pre}_\exists(X)$	$\text{exists}(x'_1, \dots, (\text{exists}(x'_n, \text{apply}(\cdot, B_{\rightarrow}, B_{X'}))) \dots)$	replace x_1, \dots, x_n by x'_1, \dots, x'_n in B_X
$\text{exists}(x', B)$	$\text{apply}(+, \text{restrict}(0, x', B), \text{restrict}(1, x', B))$	

Lemma

$$\begin{aligned} \llbracket AF \varphi \rrbracket &= \llbracket \varphi \rrbracket \cup \text{pre}_\forall(\llbracket AF \varphi \rrbracket) & \llbracket EF \varphi \rrbracket &= \llbracket \varphi \rrbracket \cup \text{pre}_\exists(\llbracket EF \varphi \rrbracket) \\ \llbracket AG \varphi \rrbracket &= \llbracket \varphi \rrbracket \cap \text{pre}_\forall(\llbracket AG \varphi \rrbracket) & \llbracket EG \varphi \rrbracket &= \llbracket \varphi \rrbracket \cap \text{pre}_\exists(\llbracket EG \varphi \rrbracket) \\ \llbracket A[\varphi \cup \psi] \rrbracket &= \llbracket \psi \rrbracket \cup (\llbracket \varphi \rrbracket \cap \text{pre}_\forall(\llbracket A[\varphi \cup \psi] \rrbracket)) & \llbracket E[\varphi \cup \psi] \rrbracket &= \llbracket \psi \rrbracket \cup (\llbracket \varphi \rrbracket \cap \text{pre}_\exists(\llbracket E[\varphi \cup \psi] \rrbracket)) \end{aligned}$$

Remark

- ▶ $\llbracket AF \varphi \rrbracket$ is **least fixed point** of function $F_{AF}(X) = \llbracket \varphi \rrbracket \cup \text{pre}_\forall(X)$
- ▶ $\llbracket EG \varphi \rrbracket$ is **greatest fixed point** of function $F_{EG}(X) = \llbracket \varphi \rrbracket \cap \text{pre}_\exists(X)$

Theorem (Knaster-Tarski)

every **monotone** function $F: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ with $|S| = n$ admits

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

Theorem (Knaster-Tarski)

every **monotone** function $F: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ with $|S| = n$ admits

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

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

Proof

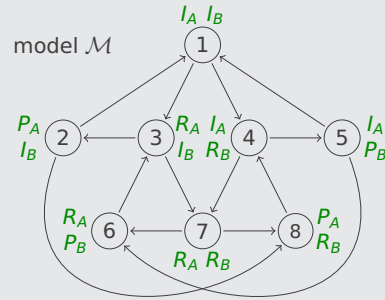
see overlay version of slides

Definition

function $F_{AF}: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$: $F_{AF}(X) = \llbracket \varphi \rrbracket \cup \text{pre}_\forall(X)$

Example

$\varphi = I_B$ $X = \emptyset$
 $F_{AF}(X) = \llbracket \varphi \rrbracket = \{1, 2, 3\}$
 $F_{AF}^2(X) = F_{AF}(F_{AF}(X)) = \{1, 2, 3\} \cup \{6\}$
 $F_{AF}^3(X) = \{1, 2, 3\} \cup \{5, 6\}$
 $F_{AF}^4(X) = \{1, 2, 3\} \cup \{5, 6\}$
 $\llbracket AF I_B \rrbracket = \{1, 2, 3, 5, 6\}$

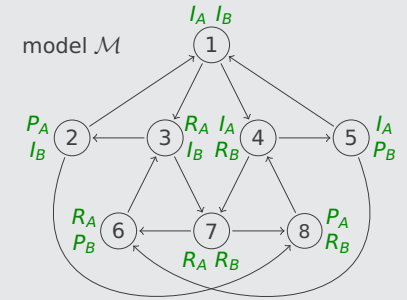


Definition

function $F_{EG}: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$: $F_{EG}(X) = \llbracket \varphi \rrbracket \cap \text{pre}_\exists(X)$

Example

$\varphi = P_A \vee I_B$
 $X = \{1, 2, 3, 4, 5, 6, 7, 8\} = \text{pre}_\exists(X)$
 $F_{EG}(X) = \llbracket \varphi \rrbracket = \{1, 2, 3, 8\}$
 $F_{EG}^2(X) = \{1, 2, 3, 8\} \cap \{1, 2, 3, 5, 6, 7\}$
 $F_{EG}^3(X) = \{1, 2, 3, 8\} \cap \{1, 2, 3, 5, 6\}$
 $\llbracket EG(P_A \vee I_B) \rrbracket = \{1, 2, 3\}$



Definition

function $F_{EU}: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$: $F_{EU}(X) = \llbracket \psi \rrbracket \cup (\llbracket \varphi \rrbracket \cap \text{pre}_\exists(X))$

Lemma

$\llbracket E[\varphi \cup \psi] \rrbracket$ is least fixed point of **monotone** function F_{EU}

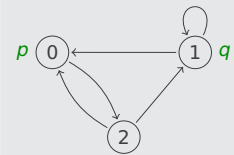
Algorithm

```

W :=  $\llbracket \varphi \rrbracket$ ;
X :=  $\emptyset$ ;
Y :=  $\llbracket \psi \rrbracket$ ;
repeat until X = Y
    X := Y;
    Y :=  $Y \cup (W \cap \text{pre}_\exists(Y))$ 
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 : $\bar{x}\bar{x}'y' + \bar{x}x'\bar{y}' + x\bar{x}'\bar{y}\bar{y}'$
 p : $x\bar{y}$ S, T : $x\bar{y} + \bar{x}y + \bar{x}\bar{y} = \bar{x} + \bar{y}$
 q : $\bar{x}y$ $\neg p \wedge q$: $((\bar{x} + \bar{y}) \oplus x\bar{y}) \cdot \bar{x}y = \bar{x}y$
 W : $\bar{x} + \bar{y}$

$AG(p \vee \neg q) \equiv \neg E[TU \neg p \wedge q]$

```

W :=  $\llbracket T \rrbracket$ ;
X :=  $\emptyset$ ;
Y :=  $\llbracket \neg p \wedge q \rrbracket$ ;
repeat until X = Y
    X := Y;
    Y :=  $Y \cup (W \cap \text{pre}_\exists(Y))$ 
return Y
    
```

X_0 0 X_1 $\bar{x}y$ X_2 \bar{x} X_3 $\bar{x} + \bar{y}$
 Y_0 $\bar{x}y$ Y_1 \bar{x} Y_2 $\bar{x} + \bar{y}$ Y_3 $\bar{x} + \bar{y}$ $X_3 = Y_3$
 $E[TU \neg p \wedge q]$: $\bar{x} + \bar{y}$
 $AG(p \vee \neg q)$: $(\bar{x} + \bar{y}) \oplus (\bar{x} + \bar{y}) = 0$

Outline

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

Outline

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

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 $\llbracket p \vee \neg p \rrbracket$ corresponds to the reduced BDD $\llbracket 0 \rrbracket$.
- C Every monotone function $F: \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ with $|S| = n$ admits a least fixed point $\mu F = F^n(S)$.
- D $\llbracket \varphi \rightarrow \perp \rrbracket = (S - \llbracket \varphi \rrbracket)$



Definitions

► LTL (linear-time temporal logic) formulas are built from

- atoms p, q, r, p_1, p_2, \dots
- logical connectives $\perp, \top, \neg, \wedge, \vee, \rightarrow$
- temporal connectives X, F, G, U, W, R

according to following BNF grammar:

$$\varphi ::= \perp \mid \top \mid p \mid (\neg \varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid (X \varphi) \mid (F \varphi) \mid (G \varphi) \mid (\varphi U \varphi) \mid (\varphi W \varphi) \mid (\varphi R \varphi)$$

► notational conventions:

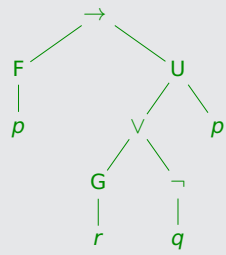
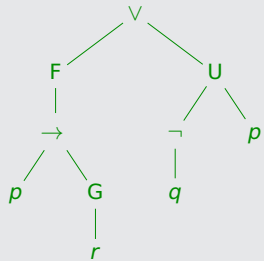
- binding precedence $\neg, X, F, G > U, W, R > \wedge, \vee > \rightarrow$
- omit outer parentheses
- $\rightarrow, \wedge, \vee$ are right-associative

Example

formula $F(p \rightarrow Gr) \vee \neg qUp$

formula $Fp \rightarrow (Gr \vee \neg q)Up$

parse tree



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

Outline

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

Definition

- ▶ **path** in model $\mathcal{M} = (S, \rightarrow, L)$ is infinite sequence $s_1 \rightarrow s_2 \rightarrow \dots$
- ▶ \forall paths $\pi = s_1 \rightarrow s_2 \rightarrow \dots \quad \forall i \geq 1 \quad \pi^i = s_i \rightarrow s_{i+1} \rightarrow \dots$

Definition

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

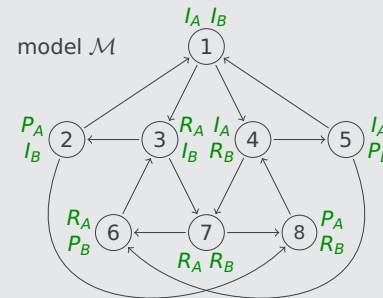
$$\pi \models \varphi$$

is defined by induction on φ :

$\pi \models \top$	$\pi \not\models \perp$	$\pi \models \varphi \wedge \psi \iff \pi \models \varphi \text{ and } \pi \models \psi$
$\pi \models p \iff p \in L(s_1)$	$\pi \models \varphi \vee \psi \iff \pi \models \varphi \text{ or } \pi \models \psi$	
$\pi \models \neg \varphi \iff \pi \not\models \varphi$	$\pi \models \varphi \rightarrow \psi \iff \pi \not\models \varphi \text{ or } \pi \models \psi$	

Example

model \mathcal{M}



$$\pi_1 = 1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow \dots$$

$$\pi_2 = 7 \rightarrow 6 \rightarrow 3 \rightarrow 7 \rightarrow 6 \rightarrow 3 \rightarrow \dots$$

special notation for infinite paths:

$$\pi_1 = (132)^\omega \quad \pi_2 = (763)^\omega$$

$$\pi_1 \models I_A$$

$$\pi_1 \not\models R_A \wedge I_B$$

$$\pi_1 \not\models I_B \rightarrow P_A \vee R_B$$

$$\pi_2 \not\models I_A$$

$$\pi_2^6 \models R_A \wedge I_B$$

$$\pi_2 \models I_B \rightarrow P_A \vee R_B$$

Definition

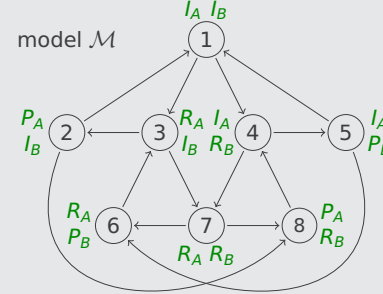
satisfaction of LTL formula φ with respect to path $\pi = s_1 \rightarrow s_2 \rightarrow \dots$ in model $\mathcal{M} = (S, \rightarrow, L)$

$$\pi \models \varphi$$

is defined by induction on φ :

$$\begin{aligned} \pi \models X\varphi &\iff \pi^2 \models \varphi \\ \pi \models F\varphi &\iff \exists i \geq 1 \pi^i \models \varphi \\ \pi \models G\varphi &\iff \forall i \geq 1 \pi^i \models \varphi \\ \pi \models \varphi U \psi &\iff \exists i \geq 1 \pi^i \models \psi \text{ and } \forall j < i \pi^j \models \varphi \\ \pi \models \varphi W \psi &\iff (\exists i \geq 1 \pi^i \models \psi \text{ and } \forall j < i \pi^j \models \varphi) \text{ or } \forall i \geq 1 \pi^i \models \varphi \\ \pi \models \varphi R \psi &\iff (\exists i \geq 1 \pi^i \models \varphi \text{ and } \forall j \leq i \pi^j \models \psi) \text{ or } \forall i \geq 1 \pi^i \models \psi \end{aligned}$$

Example



$$\pi_1 = (132)^\omega \quad \pi_2 = (763)^\omega$$

$$\pi_1 \models X(R_A \vee R_B)$$

$$\pi_1 \models F P_A$$

$$\pi_1 \not\models X X P_B$$

$$\pi_2 \not\models F P_A$$

$$\pi_2 \models G \neg I_A$$

$$\pi_2 \models G F P_B$$

$$\pi_1 \not\models I_A U P_A$$

$$\pi_2 \models \neg I_A W P_A$$

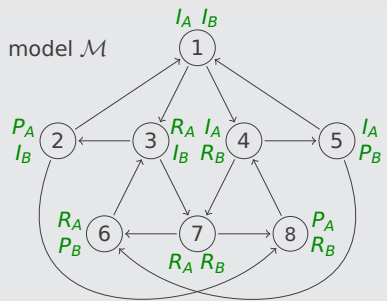
$$\pi_2 \not\models P_B R R_B$$

Definition

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

$$\mathcal{M}, s \models \varphi \iff \forall \text{ paths } \pi = s \rightarrow \dots \quad \pi \models \varphi \quad \text{"formula } \varphi \text{ holds in state } s \text{ of model } \mathcal{M}\text{"}$$

Example



$$\mathcal{M}, 1 \not\models G(R_A \rightarrow F P_A)$$

$$\mathcal{M}, 4 \not\models \neg(R_B U P_B)$$

$$\mathcal{M}, 4 \not\models R_B U P_B$$

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

Definition

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

$$\begin{aligned} \forall \text{ models } \mathcal{M} = (S, \rightarrow, L) \quad \pi \models \varphi &\iff \pi \models \psi \\ \forall \text{ paths } \pi \text{ in } \mathcal{M} \end{aligned}$$

Theorem

$$\neg X\varphi \equiv X\neg\varphi$$

$$\neg F\varphi \equiv G\neg\varphi$$

$$\neg G\varphi \equiv F\neg\varphi$$

$$\neg(\varphi U \psi) \equiv \neg\varphi R \neg\psi$$

$$\neg(\varphi R \psi) \equiv \neg\varphi U \neg\psi$$

$$\varphi U \psi \equiv \varphi W \psi \wedge F\psi$$

$$\varphi W \psi \equiv \varphi U \psi \vee G\varphi$$

$$\varphi U \psi \equiv \neg(\neg\psi U (\neg\varphi \wedge \neg\psi)) \wedge F\psi$$

$$F(\varphi \vee \psi) \equiv F\varphi \vee F\psi$$

$$G(\varphi \wedge \psi) \equiv G\varphi \wedge G\psi$$

$$F\varphi \equiv T U \varphi$$

$$G\varphi \equiv \perp R \varphi$$

$$\varphi W \psi \equiv \psi R (\varphi \vee \psi)$$

$$\varphi R \psi \equiv \psi W (\varphi \wedge \psi)$$

Theorem

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

see overlay version for proof

Outline

1. Summary of Previous Lecture

2. Symbolic Model Checking

3. Intermezzo

4. Linear-Time Temporal Logic (LTL)

Syntax Semantics Example

5. Further Reading

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 $n_i \rightarrow t_i \rightarrow c_i \rightarrow n_i \rightarrow \dots$ $(n_i t_i c_i)^\omega$

▶ asynchronous interleaving

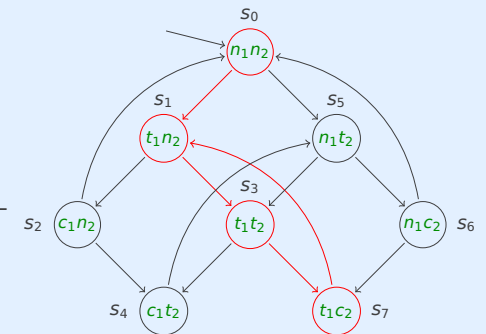
▶ model (protocol):

▶ safety: $G \neg(c_1 \wedge c_2)$ ✓

▶ liveness: $G(t_1 \rightarrow F c_1)$ ✗

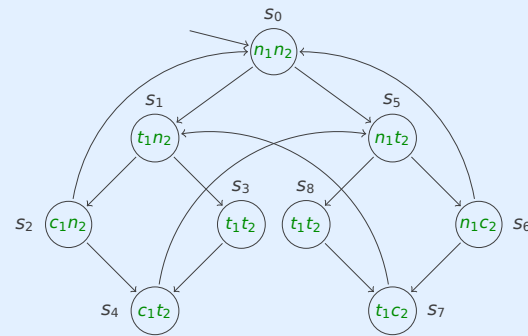
▶ non-blocking holds but is not expressible in LTL

$AG(n_1 \rightarrow EX t_1)$ in CTL



Mutual Exclusion (second modeling attempt)

- ▶ safety: $G \neg (c_1 \wedge c_2)$ ✓
- ▶ liveness: $G (t_1 \rightarrow F c_1)$ ✓



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 )
    
```

NuSMV (New Symbolic Model Verifier)

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

Outline

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

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
- ▶ pre_{\forall}
- ▶ pre_{\exists}
- ▶ R
- ▶ safety
- ▶ symbolic model checking
- ▶ U
- ▶ W
- ▶ X

homework for June 6

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