

Automated Reasoning and Formal Verification

Module II: Formal Verification

Ch. 06: **Symbolic Model Checking**

Roberto Sebastiani

DISI, Università di Trento, Italy – roberto.sebastiani@unitn.it
URL: <https://disi.unitn.it/rseba/DIDATTICA/arfv2026/>
Teaching assistant: **Gabriele Masina** – gabriele.masina@unitn.it

M.S. in Computer Science, Mathematics, & Artificial Intelligence Systems
Academic year 2025-2026

last update: Wednesday 18th February, 2026, 16:35

Copyright notice: some material (text, figures) displayed in these slides is courtesy of R. Alur, M. Benerecetti, A. Cimatti, M. Di Natale, P. Pandya, M. Pistore, M. Roveri, C. Tinelli, and S. Tonetta, who retain its copyright. Some examples displayed in these slides are taken from [Clarke, Grunberg & Peled, "Model Checking", MIT Press], and their copyright is retained by the authors. All the other material is copyrighted by Roberto Sebastiani. Every commercial use of this material is strictly forbidden by the copyright laws without the authorization of the authors. No copy of these slides can be displayed in public without containing this copyright notice.

Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

- 1 **CTL Symbolic Model Checking**
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

The Main Problem of M.C.: State Space Explosion

- **The bottleneck:**
 - Exhaustive analysis may require to store all the states of the Kripke structure, and to explore them one-by-one
 - The state space may be exponential in the number of components and variables
 - E.g., 300 Boolean vars \implies up to $2^{300} \approx 10^{100}$ states!
 - State Space Explosion:
 - too much memory required
 - too much CPU time required to explore each state
- **A solution: Symbolic Model Checking**

Symbolic Model Checking

Symbolic representation:

- manipulation of **sets of states** (rather than single states);
- sets of states represented by **formulae in propositional logic**;
 - set cardinality not directly correlated to size
- expansion of **sets of transitions** (rather than single transitions);

Symbolic Model Checking [cont.]

- Two main symbolic techniques:
 - Ordered Binary Decision Diagrams (OBDDs)
 - Propositional Satisfiability Checkers (SAT solvers)
- Different model checking algorithms:
 - Fix-point Model Checking (historically, for CTL)
 - Fix-point Model Checking for LTL (conversion to fair CTL MC)
 - Bounded Model Checking (historically, for LTL)
 - Invariant Checking
 - ...

Symbolic Representation of Kripke Models

- **Symbolic representation:**
 - **sets of states** as their **characteristic function** (Boolean formula)
 - provide logical representation and transformations of characteristic functions
- **Example:**
 - three state variables x_1, x_2, x_3 :
 $\{000, 001, 010, 011\}$ represented as “first bit false”: $\neg x_1$
 - with five state variables x_1, x_2, x_3, x_4, x_5 :
 $\{00000, 00001, 00010, 00011, 00100, 00101, 00110, 00111, \dots, 01111\}$ still represented as “first bit false”: $\neg x_1$

Kripke Models in Propositional Logic

- Let $M = (S, I, R, L, AF)$ be a Kripke model
- States $s \in S$ are described by means of **an array V of Boolean state variables**.
- A **state** is a **truth assignment** to each atomic proposition in V .
 - **0100** is represented by the formula $(\neg x_1 \wedge x_2 \wedge \neg x_3 \wedge \neg x_4)$
 - we call $\xi(s)$ the formula representing the state $s \in S$
(Intuition: $\xi(s)$ holds iff the system is in the state s)
- A set of states $Q \subseteq S$ can be represented by any formula which is logically equivalent to the formula $\xi(Q)$:

$$\bigvee_{s \in Q} \xi(s)$$

(Intuition: $\xi(Q)$ holds iff the system is in one of the states $s \in Q$)

- Bijection between models of $\xi(Q)$ and states in Q

Remark

- Every propositional formula is a (typically very compact) representation of the set of assignments satisfying it
- Any formula equivalent to $\xi(Q)$ is a representation of Q
 \implies Typically Q can be encoded by much smaller formulas than $\bigvee_{s \in Q} \xi(s)$!
- Example: $Q = \{ 00000, 00001, 00010, 00011, 00100, 00101, 00110, 00111, \dots, 01111 \}$
represented as “first bit false”: $\neg x_1$

$$\bigvee_{s \in Q} \xi(s) = \left. \begin{array}{l} (\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4 \wedge \neg x_5) \vee \\ (\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4 \wedge x_5) \vee \\ (\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg x_5) \vee \\ \dots \\ (\neg x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5) \end{array} \right\} 2^4 \text{ disjuncts}$$

Symbolic Representation of Set Operators

One-to-one correspondence between sets and Boolean operators

- Set of all the states: $\xi(S) := \top$
- Empty set : $\xi(\emptyset) := \perp$
- Union represented by disjunction:
 $\xi(P \cup Q) := \xi(P) \vee \xi(Q)$
- Intersection represented by conjunction:
 $\xi(P \cap Q) := \xi(P) \wedge \xi(Q)$
- Complement represented by negation:
 $\xi(S/P) := \neg \xi(P)$

Symbolic Representation of Transition Relations

- The transition relation R is a set of pairs of states: $R \subseteq S \times S$
- A transition is a pair of states (s, s')
- A new vector of variables V' (the next state vector) represents the value of variables after the transition has occurred
- $\xi(s, s')$ defined as $\xi(s) \wedge \xi(s')$ (Intuition: $\xi(s, s')$ holds iff the system is in the state s and moves to state s' in next step)
- The transition relation R can be represented by any formula equivalent to:

$$\bigvee_{(s,s') \in R} \xi(s, s') = \bigvee_{(s,s') \in R} (\xi(s) \wedge \xi(s'))$$

Each formula equivalent to $\xi(R)$ is a representation of R

\implies Typically R can be encoded by a much smaller formula than $\bigvee_{(s,s') \in R} \xi(s) \wedge \xi(s')$!

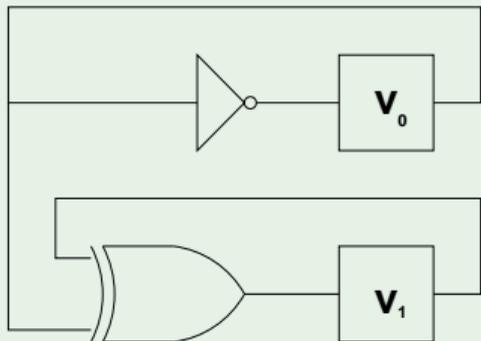
Example: a simple counter

```
MODULE main
VAR
  v0      : boolean;
  v1      : boolean;
  out     : 0..3;

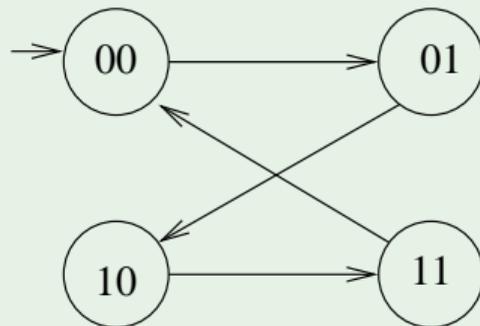
ASSIGN
  init(v0) := 0;
  next(v0) := !v0;

  init(v1) := 0;
  next(v1) := (v0 xor v1);

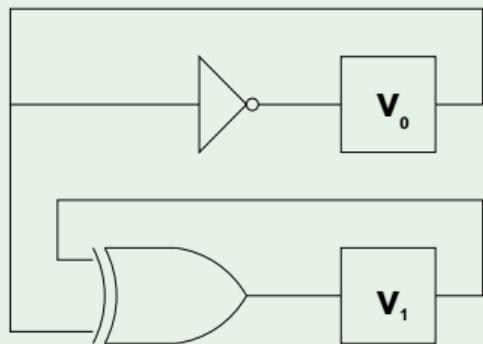
  out := toint(v0) + 2*toint(v1);
```



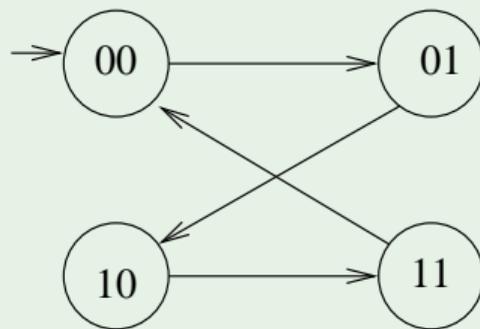
v_1	v_0	v_1'	v_0'
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



Example: a simple counter [cont.]



v_1	v_0	v'_1	v'_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

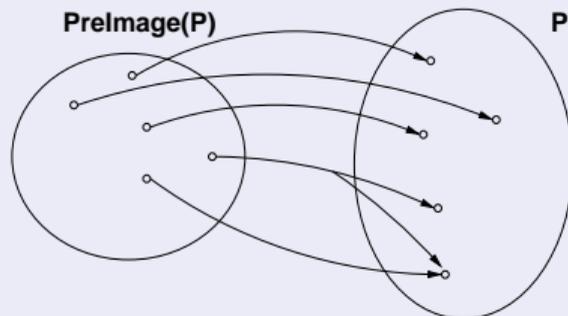


$$\xi(R) = (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1)$$

$$\begin{aligned} \bigvee_{(s,s') \in R} \xi(s) \wedge \xi(s') = & (\neg v_1 \wedge \neg v_0 \wedge \neg v'_1 \wedge v'_0) \vee \\ & (\neg v_1 \wedge v_0 \wedge v'_1 \wedge \neg v'_0) \vee \\ & (v_1 \wedge \neg v_0 \wedge v'_1 \wedge v'_0) \vee \\ & (v_1 \wedge v_0 \wedge \neg v'_1 \wedge \neg v'_0) \end{aligned}$$

Pre-Image

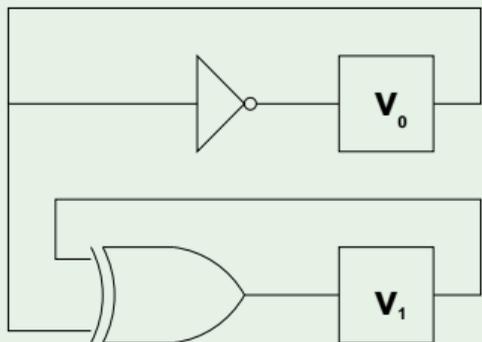
- (Backward) **pre-image** of a set of states:



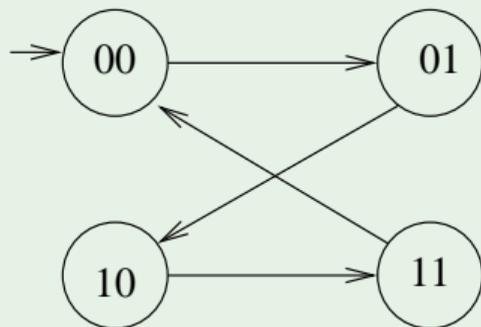
Evaluate one-shot all transitions ending in the states of the set

- Set theoretic view: $PreImage(P, R) := \{s \mid \text{for some } s' \in P, (s, s') \in R\}$
- Logical view: $\xi(PreImage(P, R)) := \exists V'. (\xi(P)[V'] \wedge \xi(R)[V, V'])$
- μ over V is s.t $\mu \models \exists V'. (\xi(P)[V'] \wedge \xi(R)[V, V'])$ iff,
for some μ' over V' , we have: $\mu \cup \mu' \models (\xi(P)[V'] \wedge \xi(R)[V, V'])$,
i.e., $\mu' \models \xi(P)[V']$ and $\mu \cup \mu' \models \xi(R)[V, V']$
 - Intuition: $\mu \iff s, \mu' \iff s', \mu \cup \mu' \iff \langle s, s' \rangle$

Example: simple counter



v_1	v_0	v'_1	v'_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

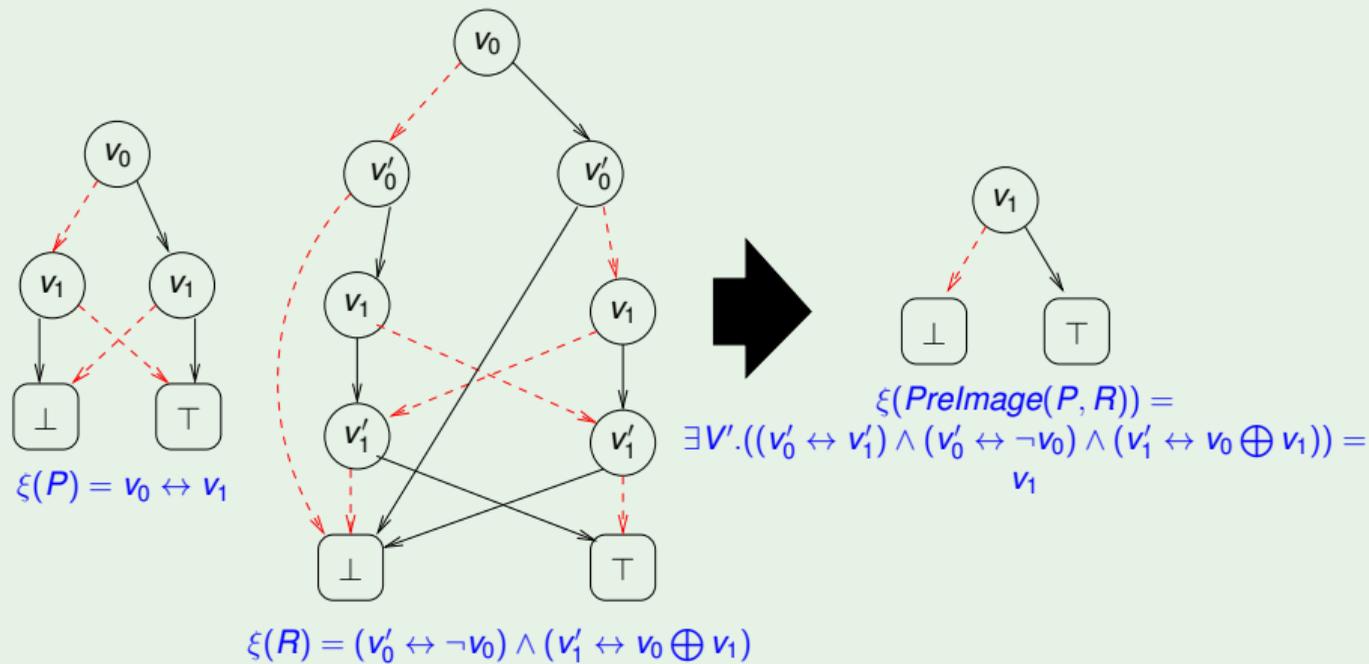


$$\xi(R) = (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1)$$

$$\xi(P) := (v_0 \leftrightarrow v_1) \text{ (i.e., } P = \{00, 11\})$$

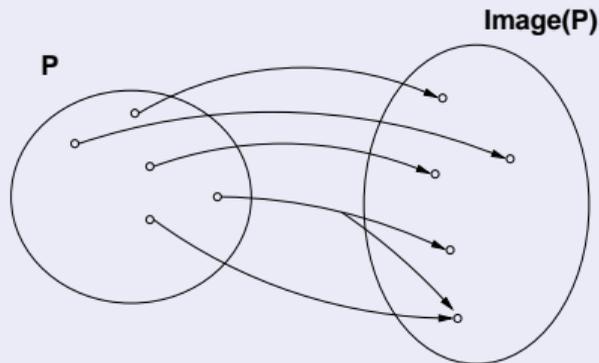
$$\begin{aligned} \xi(\text{PreImage}(P, R)) &= \\ \exists V'. (\xi(P)[V'] \wedge \xi(R)[V, V']) &= \\ \exists v'_0 v'_1. ((v'_0 \leftrightarrow v'_1) \wedge (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1)) &= \\ \underbrace{(\neg v_0 \wedge v_0 \oplus v_1)}_{v'_0=T, v'_1=T} \vee \underbrace{\perp}_{v'_0=T, v'_1=\perp} \vee \underbrace{\perp}_{v'_0=\perp, v'_1=T} \vee \underbrace{(v_0 \wedge \neg(v_0 \oplus v_1))}_{v'_0=\perp, v'_1=\perp} &= \\ v_1 \text{ (i.e., } \{10, 11\}) & \end{aligned}$$

Pre-Image [cont.]



Forward Image

- Forward image of a set:



Evaluate one-shot all transitions from the states of the set

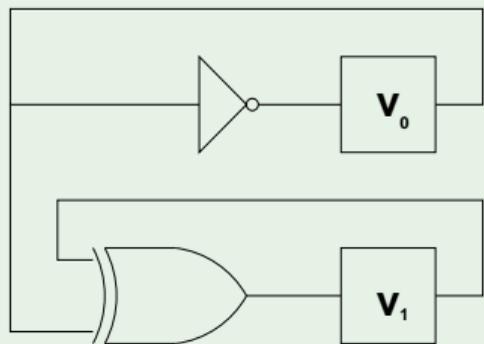
- Set theoretic view:

$$Image(P, R) := \{s' \mid \text{for some } s \in P, (s, s') \in R\}$$

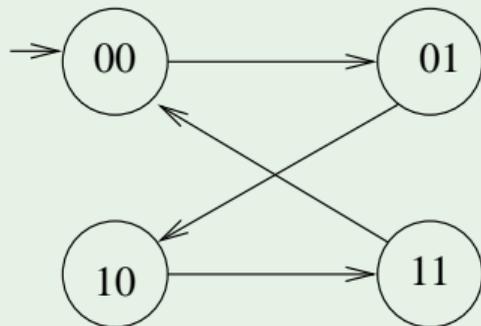
- Logical Characterization:

$$\xi(Image(P, R)) := \exists V. (\xi(P)[V] \wedge \xi(R)[V, V'])$$

Example: simple counter



v_1	v_0	v'_1	v'_0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

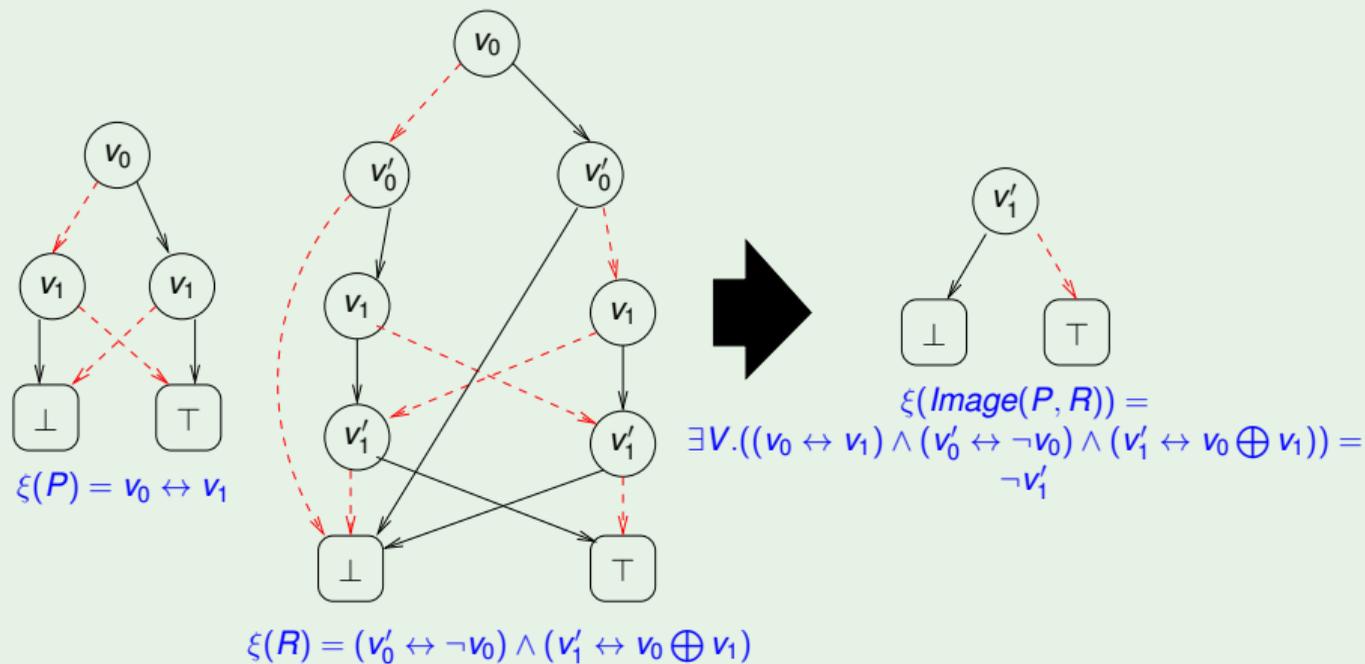


$$\xi(R) = (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1)$$

$$\xi(P) := (v_0 \leftrightarrow v_1) \text{ (i.e., } P = \{00, 11\})$$

$$\begin{aligned}\xi(\text{Image}(P, R)) &= \exists V. (\xi(P)[V] \wedge \xi(R)[V, V']) \\ &= \exists V. ((v_0 \leftrightarrow v_1) \wedge (v'_0 \leftrightarrow \neg v_0) \wedge (v'_1 \leftrightarrow v_0 \oplus v_1)) \\ &= \dots \\ &= \neg v'_1 \text{ (i.e., } \{00, 01\})\end{aligned}$$

Forward Image [cont.]



Application of the Transition Relation

- Image and PreImage of a set of states S computed by means of **quantified Boolean formulae**
- The whole set of transitions can be fired (either forward or backward) in **one logical operation**
- The symbolic computation of PreImage and Image provide the primitives for symbolic search of the state space of FSM's

Notation Remark

Henceforth, for readability sake, we omit the “ $\xi()$ ” notation in symbolic representations of systems.

- Kripke models represented as $\langle I(V), R(V, V') \rangle$
- Fair Kripke models represented as $\langle I(V), R(V, V'), F(V) \rangle$ s.t. $F(V) \stackrel{\text{def}}{=} \{F_1(V), \dots, F_k(V)\}$

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - **Symbolic CTL MC**
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

STATE-SET Check(CTL_formula β) {

case β **of**

\top : **return** S ;

\perp : **return** \emptyset ;

$\neg\beta_1$: **return** $S \setminus \text{Check}(\beta_1)$;

$\beta_1 \wedge \beta_2$: **return** $(\text{Check}(\beta_1) \cap \text{Check}(\beta_2))$;

EX β_1 : **return** $\text{PreImage}(\text{Check}(\beta_1))$;

EG β_1 : **return** $\text{Check_EG}(\text{Check}(\beta_1))$;

E(β_1 **U** β_2): **return** $\text{Check_EU}(\text{Check}(\beta_1), \text{Check}(\beta_2))$;

}

General Symbolic CTL MC Procedure

```
OBDD    Check(CTL_formula  $\beta$ ) {  
  if (In_OBDD_Hash( $\beta$ )) return OBDD_Get_From_Hash( $\beta$ );  
  case  $\beta$  of  
     $\top$ :          return obdd_true;  
     $\perp$ :          return obdd_false;  
     $\neg\beta_1$ :      return  $\neg$  Check( $\beta_1$ );  
     $\beta_1 \wedge \beta_2$ : return (Check( $\beta_1$ )  $\wedge$  Check( $\beta_2$ ));  
    EX $\beta_1$ :      return PreImage(Check( $\beta_1$ ));  
    EG $\beta_1$ :      return Check_EG(Check( $\beta_1$ ));  
    E( $\beta_1$  U  $\beta_2$ ): return Check_EU(Check( $\beta_1$ ), Check( $\beta_2$ ));  
  }
```

Some primitive functions from CTL Model Checking:

- **Symbolic Check_EX(ϕ):**
returns **an OBDD representing** the set of states from which a path verifying **X ϕ** begins (i.e., the **symbolic** preimage of the set of states where ϕ holds)
- **Symbolic Check_EG(ϕ):**
returns **an OBDD representing** the set of states from which a path verifying **G ϕ** begins
- **Symbolic Check_EU(ϕ_1, ϕ_2):**
returns **an OBDD representing** the set of states from which a path verifying **ϕ_1 U ϕ_2** begins

Check_EX

Explicit-state

```
State Set Check_EX(State Set X)  
  return {s | for some  $s' \in X, (s, s') \in R$ };
```

Symbolic

```
OBDD Check_EX(OBDD X)  
  return  $\exists V'. (X[V'] \wedge R[V, V'])$ ;
```

Same as Pre-Image computation.

Explicit-State

```
State Set Check_EG(State Set X)
  Y' := X;
  repeat
    Y := Y';
    Y' := Y  $\cap$  Check_EX(Y);
  until (Y' = Y);
  return Y;
```

Symbolic

```
OBDD Check_EG(OBDD X)
  Y' := X;
  repeat
    Y := Y';
    Y' := Y  $\wedge$  Check_EX(Y);
  until (Y'  $\leftrightarrow$  Y);
  return Y;
```

Hint (tableaux rule): $s \models \mathbf{EG}\phi$ only if $s \models \phi \wedge \mathbf{EXEG}\phi$

Explicit-State

```
State Set Check_EU(State Set  $X_1, X_2$ )  
   $Y' := X_2$ ;  
  repeat  
     $Y := Y'$ ;  
     $Y' := Y \cup (X_1 \cap \text{Check\_EX}(Y))$ ;  
  until ( $Y' = Y$ );  
return  $Y$ ;
```

Symbolic

```
OBDD Check_EU(OBDD  $X_1, X_2$ )  
   $Y' := X_2$ ;  
  repeat  
     $Y := Y'$ ;  
     $Y' := Y \vee (X_1 \wedge \text{Check\_EX}(Y))$ ;  
  until ( $Y' \leftrightarrow Y$ );  
return  $Y$ ;
```

Hint (tableaux rule): $s \models \mathbf{E}(\phi_1 \mathbf{U} \phi_2)$ if $s \models \phi_2 \vee (\phi_1 \wedge \mathbf{EXE}(\phi_1 \mathbf{U} \phi_2))$

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - **Symbolic Fair CTL MC**
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

Language-Emptiness Checking for Fair Kripke Models

Fair_CheckEG

Given: a fair Kripke model $M_F := \langle S, R, I, AP, L, F \rangle$ and a CTL formula φ s.t. $[\varphi] \subseteq S$,
 $\text{Fair_CheckEG}(\varphi)$ returns the subset of the states s in $[\varphi]$ from which at least one fair path π entirely included in $[\varphi]$ passes through

Symbolic Fair_CheckEG

Given: the symbolic representation of a fair Kripke model $M_F := \langle I, R, F \rangle$
and a Boolean formula (OBDD) Ψ ,
 $\text{Fair_CheckEG}(\Psi)$ returns a Boolean formula (OBDD) representing the subset of the states s in Ψ from which at least one fair path π entirely included in Ψ passes through

$\text{Fair_CheckEG}(\text{true})$ computes (the symbolic representation of) the set of fair states of M_f
 $\implies I \subseteq \text{Fair_CheckEG}(\text{true})$ iff $\mathcal{L}(M_f) \neq \emptyset$

Ingredients (from Symbolic CTL Model Checking)

Some primitive functions from CTL Model Checking:

- **Symbolic Check_EX(ϕ)**: returns an OBDD representing the set of states from which a path verifying **X** ϕ begins
(i.e., the **symbolic** preimage of the set of states where ϕ holds)
- **Symbolic Check_EG(ϕ)**: returns an OBDD representing the set of states from which a path verifying **G** ϕ begins
- **Symbolic Check_EU(ϕ_1, ϕ_2)**: returns an OBDD representing the set of states from which a path verifying ϕ_1 **U** ϕ_2 begins

Emerson-Lei Algorithm

Recall: $[E_f G \phi] = \nu Z.([\phi] \cap \bigcap_{F_i \in FT} [EX E(Z \cup (Z \cap F_i))])$

```
state_set Check_FairEG(state_set [ $\phi$ ]) {  
   $Z' := [\phi]$ ;  
  repeat  
     $Z := Z'$ ;  
    for each  $F_i$  in FT  
       $Y := \text{Check\_EU}(Z', F_i \cap Z')$ ;  
       $Z' := Z' \cap \text{PreImage}(Y)$ ;  
    end for;  
  until ( $Z' = Z$ );  
  return  $Z$ ;  
}
```

Slight improvement: do not consider states in $Z \setminus Z'$

Emerson-Lei Algorithm (symbolic version)

Recall: $[E_f G \phi] = \nu Z.([\phi] \cap \bigcap_{F_i \in FT} [EX E(ZU(Z \wedge F_i))])$

```
Obdd Check_FairEG( Obdd  $\phi$  ) {  
   $Z' := \phi$ ;  
  repeat  
     $Z := Z'$ ;  
    for each  $F_i$  in FT  
       $Y := \text{Check\_EU}(Z', F_i \wedge Z')$ ;  
       $Z' := Z' \wedge \text{PreImage}(Y)$ ;  
    end for;  
  until ( $Z' \leftrightarrow Z$ );  
  return  $Z$ ;  
}
```

Symbolic version.

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - **A simple example**
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

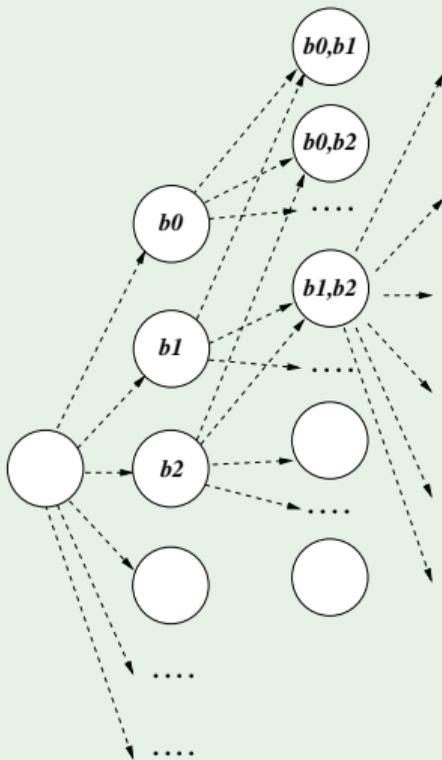
A simple example

```
MODULE main
VAR
  b0 : boolean;
  b1 : boolean;
  ...
ASSIGN
  init(b0) := 0;
  next(b0) := case
    b0 : 1;
    !b0 : {0,1};
  esac;
  init(b1) := 0;
  next(b1) := case
    b1 : 1;
    !b1 : {0,1};
  esac;
  ...
```

A simple example [cont.]

- N Boolean variables b_0, b_1, \dots
- Initially, all variables set to 0
- Each variable can pass from 0 to 1, but not vice-versa
- 2^N states, all reachable
- (Simplified) model of a student career behaviour.

A simple example: FSM

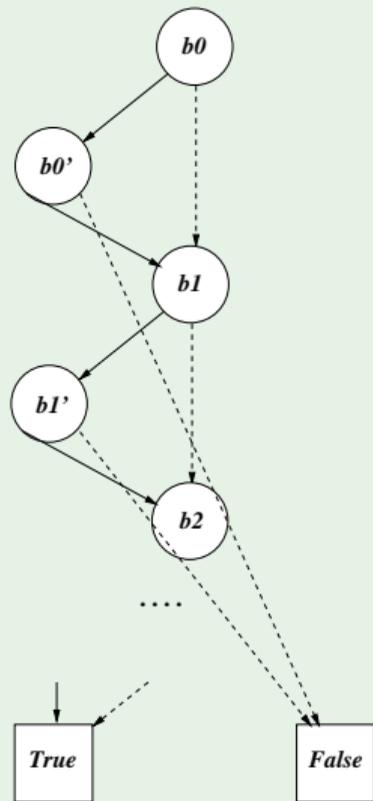


(transitive transitions omitted)

2^N STATES

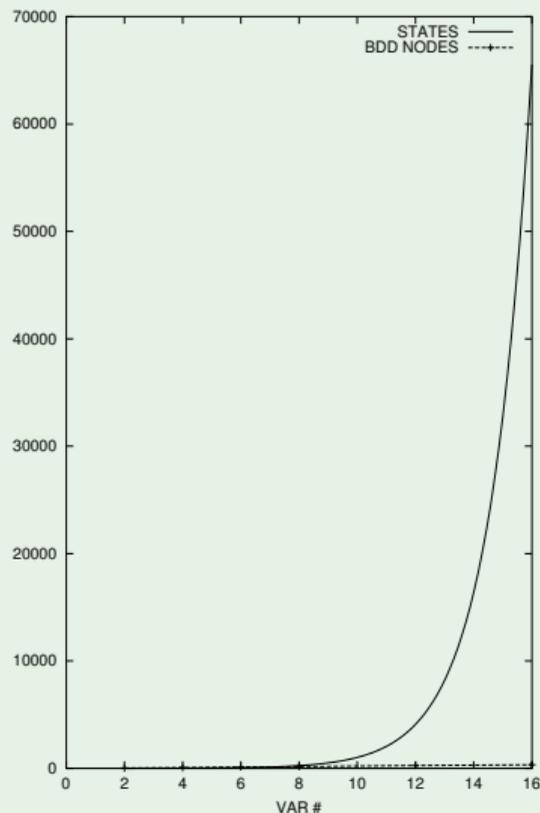
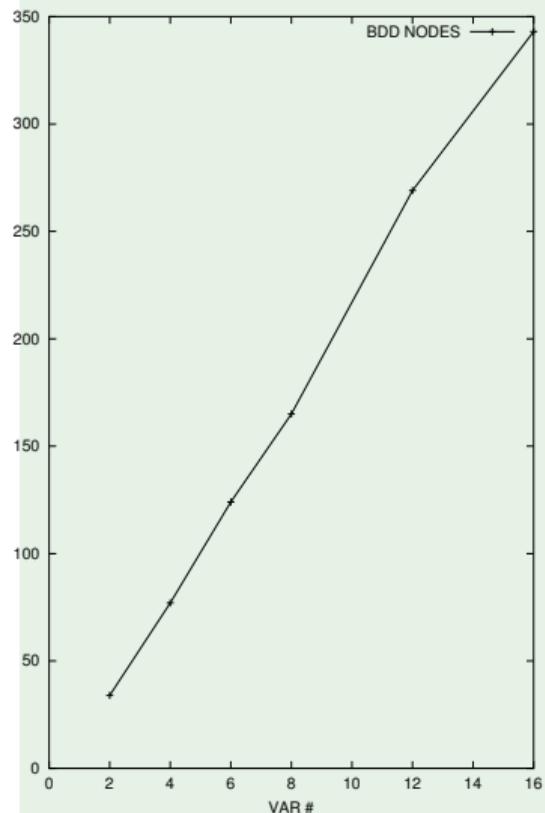
$O(2^N)$ TRANSITIONS

A simple example: $OBDD(\xi(R))$



$2N + 2$ NODES

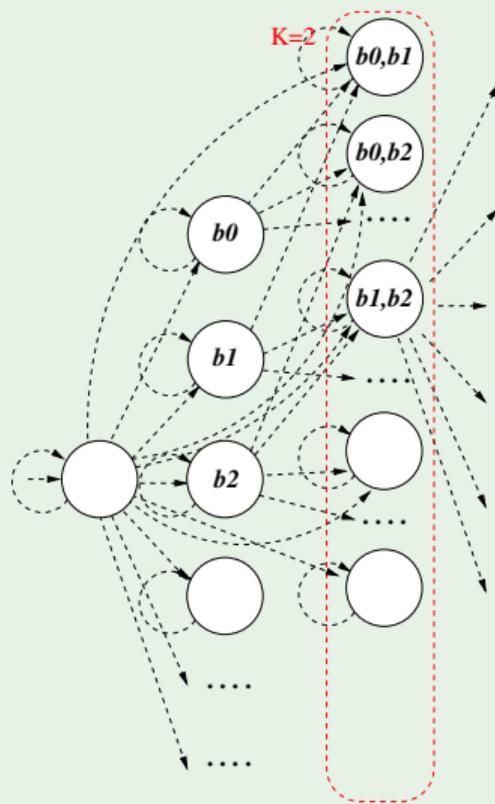
A simple example: states vs. OBDD nodes [NuSMV.2]



A simple example: reaching K bits true

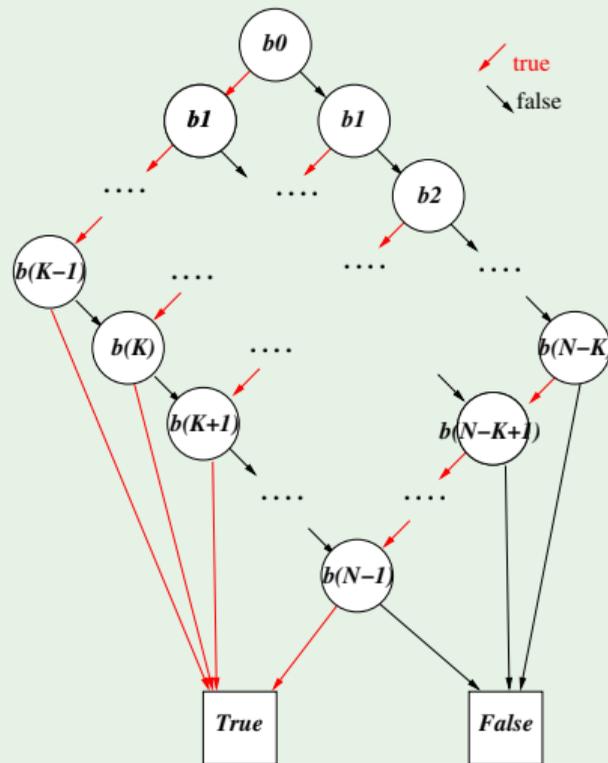
- Property $\mathbf{EF}(b_0 + b_1 + \dots + b_{(N-1)} \geq K)$ ($K \leq N$)
(it may be reached a state in which K bits are true)
- E.g.: “it is reachable a state where K exams are passed”

A simple example: FSM



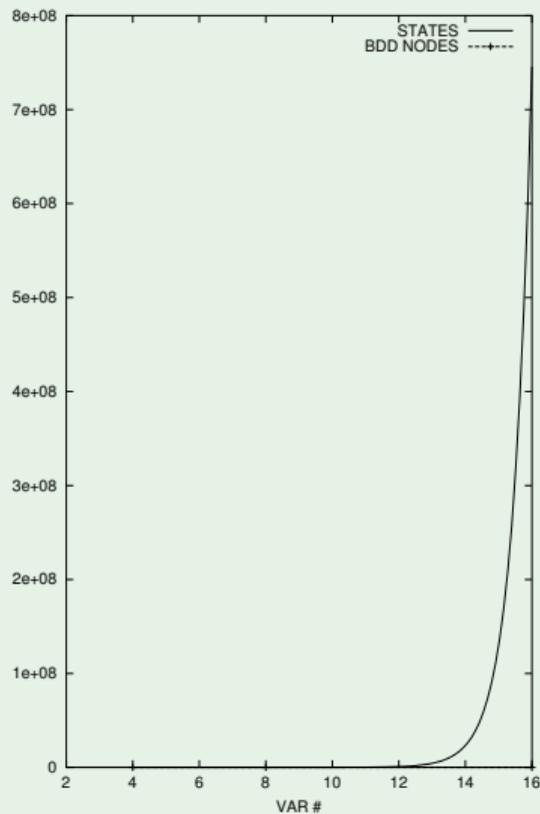
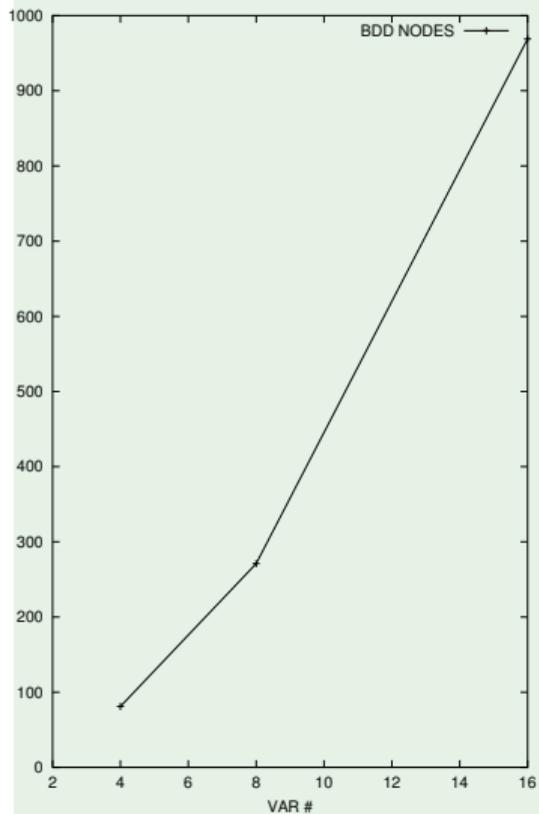
$$\binom{N}{K} + \binom{N}{K+1} + \dots + \binom{N}{N}$$

A simple example: $OBDD(\xi(\varphi))$



$(N - K + 1) \cdot K + 2$ NODES

A simple example: states vs. OBDD nodes [NuSMV.2]



Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 **The Symbolic Approach to LTL Model Checking**
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

Symbolic LTL Satisfiability and Entailment

LTL Validity/Satisfiability

- Let ψ be an LTL formula

$$\models \psi \quad (\text{LTL})$$

$$\iff \neg\psi \text{ unsat}$$

$$\iff \mathcal{L}(T_{\neg\psi}) = \emptyset$$

- $T_{\neg\psi}$ is a **fair Kripke model** (aka **tableaux**) which represents all and only the paths that satisfy $\neg\psi$ (do not satisfy ψ)

LTL Entailment

- Let φ, ψ be an LTL formula

$$\varphi \models \psi \quad (\text{LTL})$$

$$\models \varphi \rightarrow \psi \quad (\text{LTL})$$

$$\iff \varphi \wedge \neg\psi \text{ unsat}$$

$$\iff \mathcal{L}(T_{\varphi \wedge \neg\psi}) = \emptyset$$

- $T_{\varphi \wedge \neg\psi}$ is a **fair Kripke model** (aka **tableaux**) which represents all and only the paths that satisfy $\varphi \wedge \neg\psi$ (satisfy φ and do not satisfy ψ)

Symbolic LTL Model Checking

LTL Model Checking

- Let M be a Kripke model and ψ be an LTL formula

$$M \models \psi \quad (\text{LTL})$$

$$\iff \mathcal{L}(M) \subseteq \mathcal{L}(\psi)$$

$$\iff \mathcal{L}(M) \cap \overline{\mathcal{L}(\psi)} = \emptyset$$

$$\iff \mathcal{L}(M) \cap \mathcal{L}(\neg\psi) = \emptyset$$

$$\iff \mathcal{L}(M) \cap \mathcal{L}(T_{\neg\psi}) = \emptyset$$

$$\iff \mathcal{L}(M \times T_{\neg\psi}) = \emptyset$$

- $T_{\neg\psi}$ is a **fair Kripke model** (aka **tableaux**) which represents all and only the paths that satisfy $\neg\psi$ (do not satisfy ψ)

$\implies M \times T_{\neg\psi}$ represents all and only the paths appearing in M and not in ψ .

Symbolic LTL Model Checking

Three steps

Let $\varphi \stackrel{\text{def}}{=} \neg\psi$:

- (i) Compute T_φ
- (ii) Compute the product $M \times T_\varphi$
- (iii) Check the emptiness of $\mathcal{L}(M \times T_\varphi)$

Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - **Compute the Tableau T_ψ**
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

The Set of States

- Elementary subformulas of ψ : $el(\psi)$
 - $el(p) := \{p\}$
 - $el(\neg\varphi_1) := el(\varphi_1)$
 - $el(\varphi_1 \wedge \varphi_2) := el(\varphi_1) \cup el(\varphi_2)$
 - $el(\mathbf{X}\varphi_1) = \{\mathbf{X}\varphi_1\} \cup el(\varphi_1)$
 - $el(\varphi_1 \mathbf{U}\varphi_2) := \{\mathbf{X}(\varphi_1 \mathbf{U}\varphi_2)\} \cup el(\varphi_1) \cup el(\varphi_2)$
- Intuition: $el(\psi)$ is the set of propositions and \mathbf{X} -formulas occurring in ψ , ψ' being the result of applying recursively the tableau expansion rules to ψ
- The set of states S_{T_ψ} of T_ψ is given by $2^{el(\psi)}$
- The labeling function L_{T_ψ} of T_ψ comes straightforwardly (the label is the Boolean component of each state)

Example: $\psi := p\mathbf{U}q$

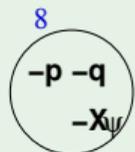
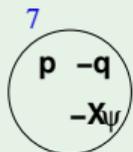
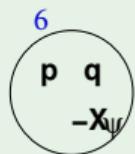
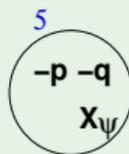
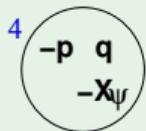
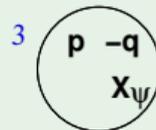
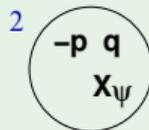
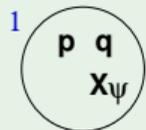
- $el(p\mathbf{U}q) = el((q \vee (p \wedge \mathbf{X}(p\mathbf{U}q))) = \{p, q, \mathbf{X}(p\mathbf{U}q)\}$

$$\implies S_{T_\psi} = \{$$

- | | | |
|-----|--|-----------------------|
| 1 : | $\{p, q, \mathbf{X}(p\mathbf{U}q)\},$ | $[p\mathbf{U}q]$ |
| 2 : | $\{\neg p, q, \mathbf{X}(p\mathbf{U}q)\},$ | $[p\mathbf{U}q]$ |
| 3 : | $\{p, \neg q, \mathbf{X}(p\mathbf{U}q)\},$ | $[p\mathbf{U}q]$ |
| 4 : | $\{\neg p, q, \neg\mathbf{X}(p\mathbf{U}q)\},$ | $[p\mathbf{U}q]$ |
| 5 : | $\{\neg p, \neg q, \mathbf{X}(p\mathbf{U}q)\},$ | $[\neg p\mathbf{U}q]$ |
| 6 : | $\{p, q, \neg\mathbf{X}(p\mathbf{U}q)\},$ | $[p\mathbf{U}q]$ |
| 7 : | $\{p, \neg q, \neg\mathbf{X}(p\mathbf{U}q)\},$ | $[\neg p\mathbf{U}q]$ |
| 8 : | $\{\neg p, \neg q, \neg\mathbf{X}(p\mathbf{U}q)\}$ | $[\neg p\mathbf{U}q]$ |

}

Example: $\psi := p \mathbf{U} q$ [cont.]



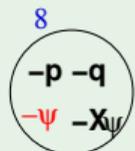
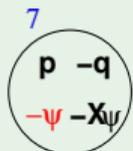
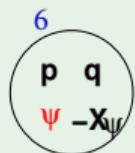
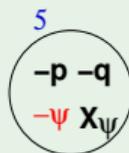
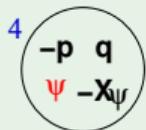
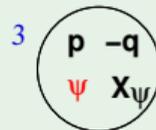
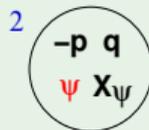
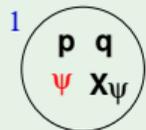
sat()

- Set of states in S_{T_ψ} satisfying φ_i : $sat(\varphi_i)$
 - $sat(\varphi_1) := \{s \mid \varphi_1 \in s\}$, $\varphi_1 \in el(\psi)$
 - $sat(\neg\varphi_1) := S_{T_\psi} / sat(\varphi_1)$
 - $sat(\varphi_1 \wedge \varphi_2) := sat(\varphi_1) \cap sat(\varphi_2)$
 - $sat(\varphi_1 \mathbf{U} \varphi_2) := sat(\varphi_2) \cup (sat(\varphi_1) \cap sat(\mathbf{X}(\varphi_1 \mathbf{U} \varphi_2)))$
- intuition: $sat()$ establishes in which states subformulas are true

Remark

- Semantics of “ $\varphi_1 \mathbf{U} \varphi_2$ ” here induced by tableaux rule: $\varphi_1 \mathbf{U} \varphi_2 \stackrel{\text{def}}{=} \varphi_2 \vee (\varphi_1 \wedge \mathbf{X}(\varphi_1 \mathbf{U} \varphi_2))$
- \implies weaker than standard semantics (aka “**weak until**”, “ $\varphi_1 \mathbf{W} \varphi_2$ ”):
a path where φ_1 is always true and φ_2 is always false satisfies it

Example: $\psi := p \mathbf{U} q$ [cont.]



Initial States and Transition Relation

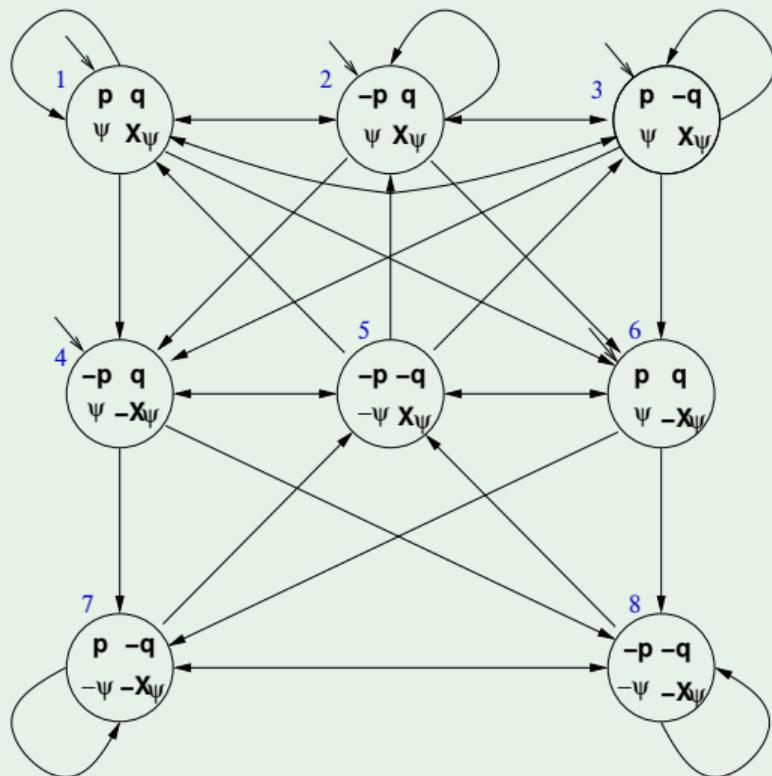
- Set of states in S_{T_ψ} satisfying φ_i : $sat(\varphi_i)$
 - $sat(\varphi_1) := \{s \mid \varphi_1 \in s\}, \varphi_1 \in el(\psi)$
 - $sat(\neg\varphi_1) := S_{T_\psi} / sat(\varphi_1)$
 - $sat(\varphi_1 \wedge \varphi_2) := sat(\varphi_1) \cap sat(\varphi_2)$
 - $sat(\varphi_1 \mathbf{U} \varphi_2) := sat(\varphi_2) \cup (sat(\varphi_1) \cap sat(\mathbf{X}(\varphi_1 \mathbf{U} \varphi_2)))$
- Intuition: $sat()$ establishes in which states subformulas are true
- The set of initial states I_{T_ψ} is defined as

$$I_{T_\psi} = sat(\psi)$$

- The transition relation R_{T_ψ} is defined as

$$R_{T_\psi}(s, s') = \bigcap_{\mathbf{X}\varphi_i \in el(\psi)} \{(s, s') \mid s \in sat(\mathbf{X}\varphi_i) \Leftrightarrow s' \in sat(\varphi_i)\}$$

Example: $\psi := p \mathbf{U} q$ [cont.]



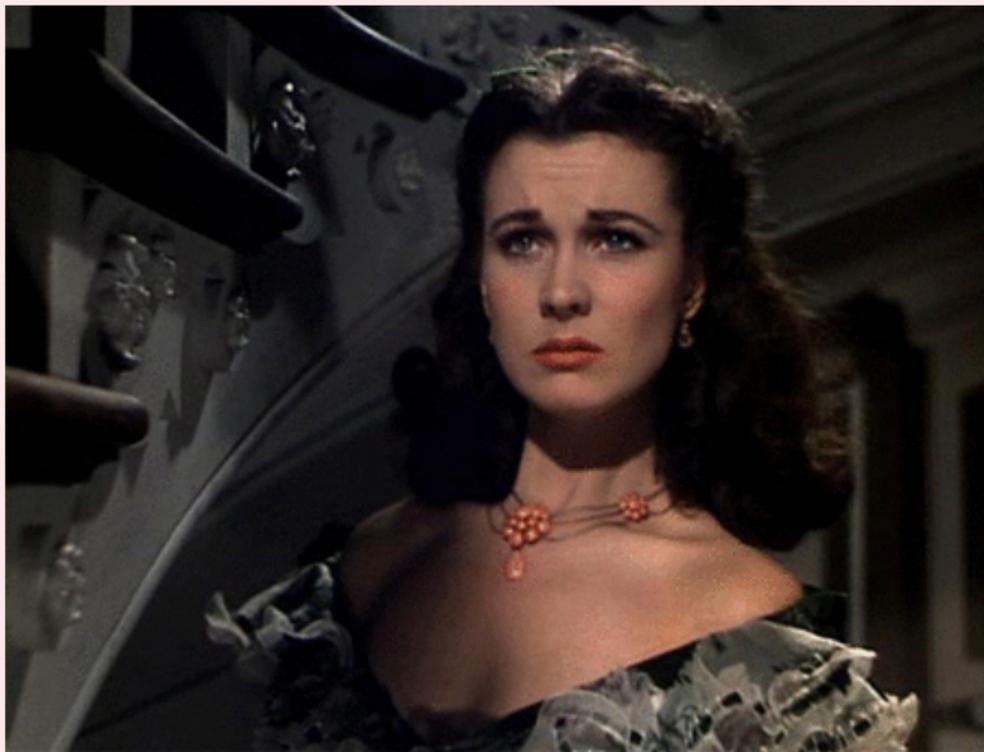
Problems with **U**-subformulas

- R_{T_ψ} does not guarantee that the **U**-subformulas are fulfilled
- Example: state 3 $\{p, \neg q, \mathbf{X}(p\mathbf{U}q)\}$:
although state 3 belongs to

$$sat(p\mathbf{U}q) := sat(q) \cup (sat(p) \cap sat(\mathbf{X}(p\mathbf{U}q))),$$

the path which loops forever in state 3 does not satisfy $p\mathbf{U}q$, as q never holds in that path.

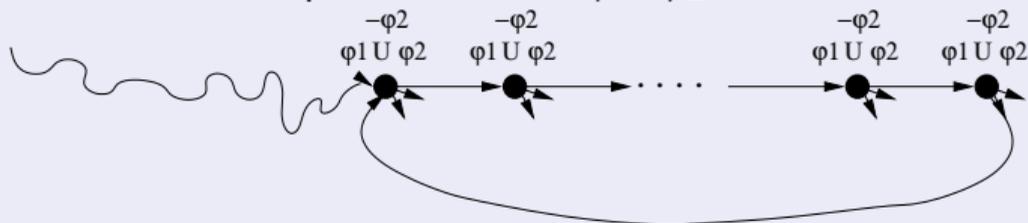
Tableaux Rules: a Quote



*"After all... tomorrow is another day."
[Scarlett O'Hara, "Gone with the Wind"]*

Fairness conditions for every **U**-subformula

- It must never happen that we get into a state s' from which we can enter a path π' in which $\varphi_1 \mathbf{U} \varphi_2$ holds forever and φ_2 never holds.



⇒ For every [positive] **U**-subformula $\varphi_1 \mathbf{U} \varphi_2$ of ψ , we must add a **fairness LTL condition** $\mathbf{GF}(\neg(\varphi_1 \mathbf{U} \varphi_2) \vee \varphi_2)$

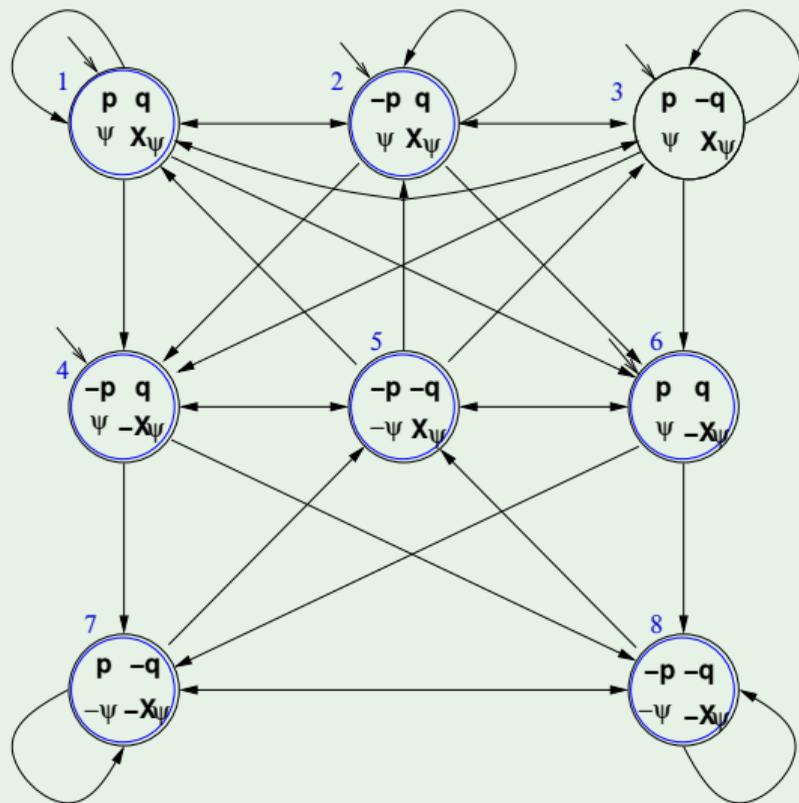
If no [positive] **U**-subformulas, then add one fairness condition $\mathbf{GF}\top$.

⇒ We restrict the admissible paths of T_ψ to those which verify the fairness condition:

$$T_\psi := \langle S_{T_\psi}, I_{T_\psi}, R_{T_\psi}, L_{T_\psi}, F_{T_\psi} \rangle$$

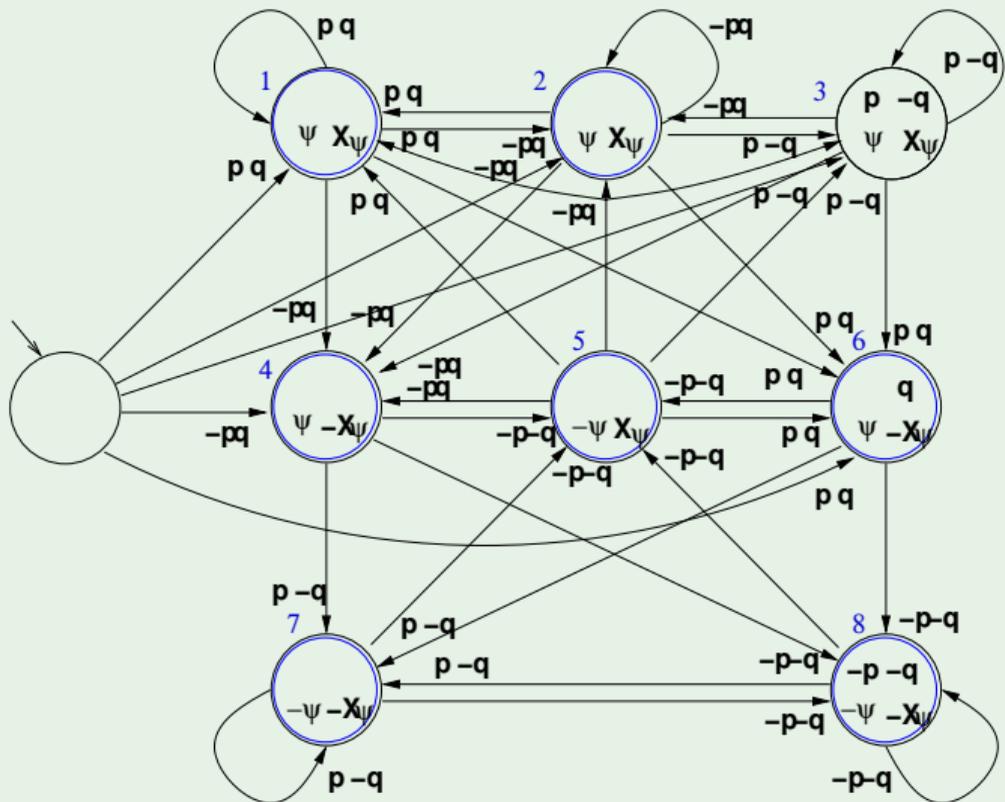
$$F_{T_\psi} := \{ \text{sat}(\neg(\varphi_1 \mathbf{U} \varphi_2) \vee \varphi_2) \text{ s.t. } (\varphi_1 \mathbf{U} \varphi_2) \text{ occurs [positively] in } \psi \}$$

Example: $\psi := p \mathbf{U} q$ [cont.]



Note: easily transformed into a generalized Büchi automaton

Example: $\psi := p \mathbf{U} q$ [cont.]



Note: easily transformed into a generalized Büchi automaton

Symbolic Representation of T_ψ

- State variables: one Boolean variable for each formula in $eI(\psi)$
 - EX: p, q and x and primed versions p', q' and x'
[x is a Boolean label for $\mathbf{X}(p\mathbf{U}q)$]
- $sat(\varphi_i)$:
 - $sat(p) := p$, s.t. p Boolean state variable
 - $sat(\neg\varphi_1) := \neg sat(\varphi_1)$
 - $sat(\varphi_1 \wedge \varphi_2) := sat(\varphi_1) \wedge sat(\varphi_2)$
 - $sat(\mathbf{X}\varphi_i) := x_{[x\varphi_i]}$, s.t. $x_{[x\varphi_i]}$ Boolean state variable
 - $sat(\varphi_1 \mathbf{U}\varphi_2) := sat(\varphi_2) \vee (sat(\varphi_1) \wedge sat(\mathbf{X}(\varphi_1 \mathbf{U}\varphi_2)))$
 - $\Rightarrow sat(\varphi_1 \mathbf{U}\varphi_2) := sat(\varphi_2) \vee (sat(\varphi_1) \wedge x_{[x\varphi_1 \mathbf{U}\varphi_2]})$
- ...

Symbolic Representation of T_ψ [cont.]

- ...
- Initial states: $I_{T_\psi} = \text{sat}(\psi)$
 - EX: $I(p, q, x) = q \vee (p \wedge x)$
- Transition Relation: $R_{T_\psi}(s, s') = \bigcap_{\mathbf{x}\varphi_i \in \text{el}(\psi)} \{(s, s') \mid s \in \text{sat}(\mathbf{X}\varphi_i) \leftrightarrow s' \in \text{sat}(\varphi_i)\}$
 - $R_{T_\psi} = \bigwedge_{\mathbf{x}\varphi_i \in \text{el}(\psi)} (\text{sat}(\mathbf{X}\varphi_i) \leftrightarrow \text{sat}'(\varphi_i))$
where $\text{sat}'(\varphi_i)$ is $\text{sat}(\varphi_i)$ on primed variables
 - EX: $R_{T_\psi}(p, q, x, p', q', x') = x \leftrightarrow (q' \vee (p' \wedge x'))$
- **Fairness Conditions:** $F_{T_\psi} := \{\text{sat}(\neg(\varphi_1 \mathbf{U} \varphi_2) \vee \varphi_2)\}$ s.t. $(\varphi_1 \mathbf{U} \varphi_2)$ occurs [positively] in ψ
 - EX: $F_{T_\psi}(p, q, x) = \neg(q \vee (p \wedge x)) \vee q = \dots = \neg p \vee \neg x \vee q$

Symbolic Representation of T_ψ : Examples

- $I_{T_\psi}(p, q, x) = q \vee (p \wedge x)$

- 1 : $\{p, q, x\} \models I_{T_\psi}$

- 3 : $\{p, \neg q, x\} \models I_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \not\models I_{T_\psi}$

- $R_{T_\psi}(p, q, x, p', q', x') = x \leftrightarrow (q' \vee (p' \wedge x'))$

- 1 \Rightarrow 1 : $\{p, q, x, p', q', x'\} \models R_{T_\psi}$

- 6 \Rightarrow 7 : $\{p, q, \neg x, p', \neg q', \neg x'\} \models R_{T_\psi}$

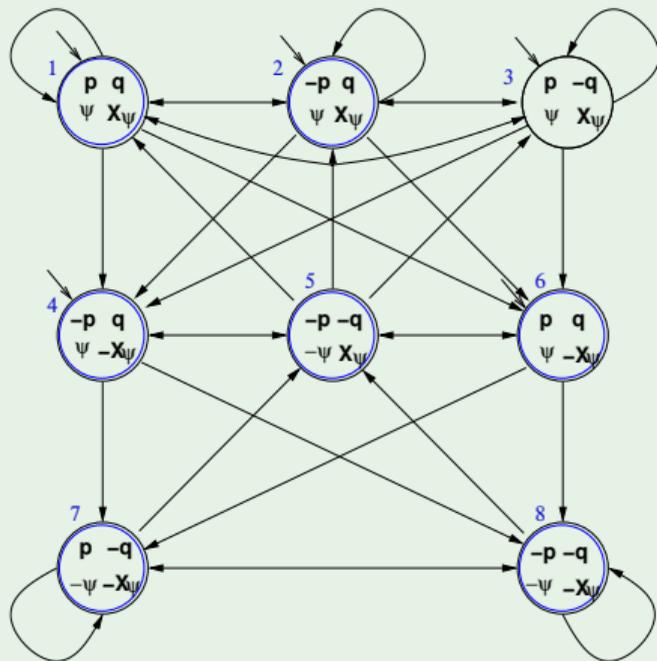
- 6 $\not\Rightarrow$ 1 : $\{p, q, \neg x, p', q', x'\} \not\models R_{T_\psi}$

- $F_{T_\psi}(p, q, x) = \neg p \vee \neg x \vee q$

- 1 : $\{p, q, x\} \models F_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \models F_{T_\psi}$

- 3 : $\{p, \neg q, x\} \not\models F_{T_\psi}$



Symbolic Representation of T_ψ : Examples

- $I_{T_\psi}(p, q, x) = q \vee (p \wedge x)$

- 1 : $\{p, q, x\} \models I_{T_\psi}$

- 3 : $\{p, \neg q, x\} \models I_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \not\models I_{T_\psi}$

- $R_{T_\psi}(p, q, x, p', q', x') = x \leftrightarrow (q' \vee (p' \wedge x'))$

- 1 \Rightarrow 1 : $\{p, q, x, p', q', x'\} \models R_{T_\psi}$

- 6 \Rightarrow 7 : $\{p, q, \neg x, p', \neg q', \neg x'\} \models R_{T_\psi}$

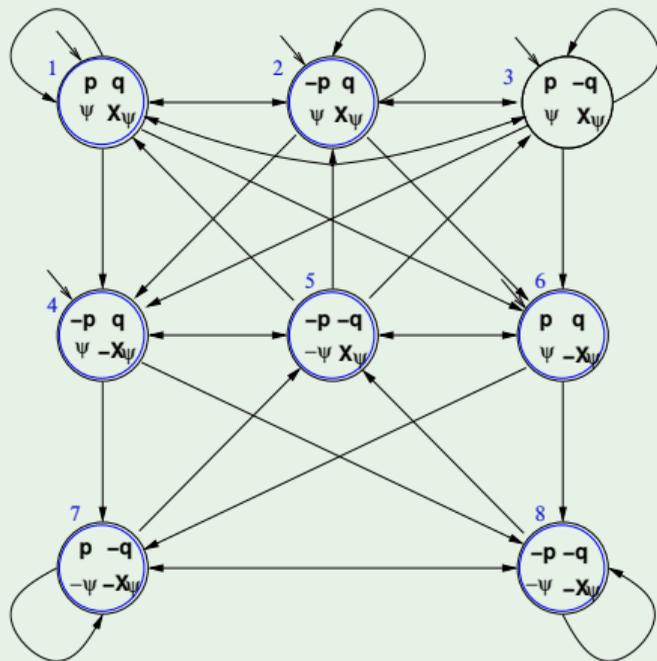
- 6 $\not\Rightarrow$ 1 : $\{p, q, \neg x, p', q', x'\} \not\models R_{T_\psi}$

- $F_{T_\psi}(p, q, x) = \neg p \vee \neg x \vee q$

- 1 : $\{p, q, x\} \models F_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \models F_{T_\psi}$

- 3 : $\{p, \neg q, x\} \not\models F_{T_\psi}$



Symbolic Representation of T_ψ : Examples

- $I_{T_\psi}(p, q, x) = q \vee (p \wedge x)$

- 1 : $\{p, q, x\} \models I_{T_\psi}$

- 3 : $\{p, \neg q, x\} \models I_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \not\models I_{T_\psi}$

- $R_{T_\psi}(p, q, x, p', q', x') = x \leftrightarrow (q' \vee (p' \wedge x'))$

- 1 \Rightarrow 1 : $\{p, q, x, p', q', x'\} \models R_{T_\psi}$

- 6 \Rightarrow 7 : $\{p, q, \neg x, p', \neg q', \neg x'\} \models R_{T_\psi}$

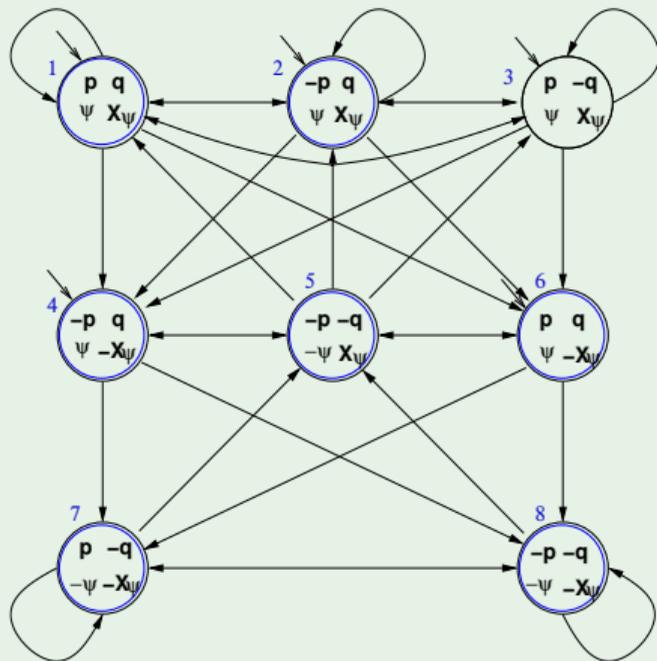
- 6 $\not\Rightarrow$ 1 : $\{p, q, \neg x, p', q', x'\} \not\models R_{T_\psi}$

- $F_{T_\psi}(p, q, x) = \neg p \vee \neg x \vee q$

- 1 : $\{p, q, x\} \models F_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \models F_{T_\psi}$

- 3 : $\{p, \neg q, x\} \not\models F_{T_\psi}$



Symbolic Representation of T_ψ : Examples

- $I_{T_\psi}(p, q, x) = q \vee (p \wedge x)$

- 1 : $\{p, q, x\} \models I_{T_\psi}$

- 3 : $\{p, \neg q, x\} \models I_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \not\models I_{T_\psi}$

- $R_{T_\psi}(p, q, x, p', q', x') = x \leftrightarrow (q' \vee (p' \wedge x'))$

- 1 \Rightarrow 1 : $\{p, q, x, p', q', x'\} \models R_{T_\psi}$

- 6 \Rightarrow 7 : $\{p, q, \neg x, p', \neg q', \neg x'\} \models R_{T_\psi}$

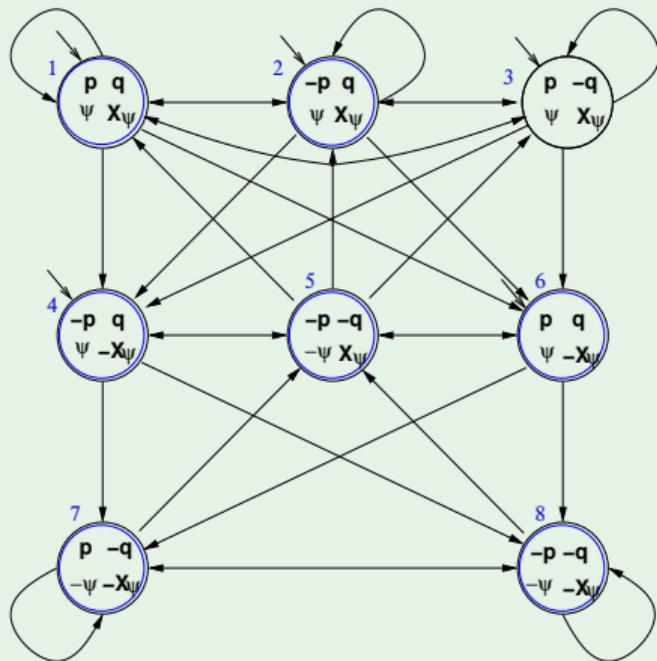
- 6 $\not\Rightarrow$ 1 : $\{p, q, \neg x, p', q', x'\} \not\models R_{T_\psi}$

- $F_{T_\psi}(p, q, x) = \neg p \vee \neg x \vee q$

- 1 : $\{p, q, x\} \models F_{T_\psi}$

- 5 : $\{\neg p, \neg q, x\} \models F_{T_\psi}$

- 3 : $\{p, \neg q, x\} \not\models F_{T_\psi}$



Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - **Compute the Product $M \times T_\psi$**
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

Computing the product $P := T_\psi \times M$

- Given $M := \langle S_M, I_M, R_M, L_M \rangle$ and $T_\psi := \langle S_{T_\psi}, I_{T_\psi}, R_{T_\psi}, L_{T_\psi}, F_{T_\psi} \rangle$, we compute the product $P := T_\psi \times M = \langle S, I, R, L, F \rangle$ as follows:
 - $S := \{(s, s') \mid s \in S_{T_\psi}, s' \in S_M \text{ and } L_M(s')|_\psi = L_{T_\psi}(s)\}$
 - $I := \{(s, s') \mid s \in I_{T_\psi}, s' \in I_M \text{ and } L_M(s')|_\psi = L_{T_\psi}(s)\}$
 - Given $(s, s'), (t, t') \in S$, $((s, s'), (t, t')) \in R$ iff $(s, t) \in R_{T_\psi}$ and $(s', t') \in R_M$
 - $L((s, s')) = L_{T_\psi}(s) \cup L_M(s')$
- Extension of $\text{sat}()$ and F_{T_ψ} to P :
 - $(s, s') \in \text{sat}(\psi) \iff s \in \text{sat}(\psi)$
 - $F := \{\text{sat}(\neg(\varphi_1 \mathbf{U} \varphi_2) \vee \varphi_2) \text{ s.t. } (\varphi_1 \mathbf{U} \varphi_2) \text{ occurs [positively] in } \psi\}$

Computing the product $P := T_\psi \times M$ symbolically

Let V, W be the array of Boolean state variables of T_ψ and M respectively:

- Initial states: $I(V \cup W) = I_{T_\psi}(V) \wedge I_M(W)$
- Transition Relation: $R(V \cup W, V' \cup W') = R_{T_\psi}(V, V') \wedge R_M(W, W')$
- Fairness conditions: $\{F_1(V \cup W), \dots, F_k(V \cup W)\} = \{F_{T_\psi 1}(V), \dots, F_{T_\psi k}(V)\}$

Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

Main theorem [Clarke, Grumberg & Hamaguchi; 94]

Theorem

THEOREM: $M.s' \models \mathbf{E}\psi$ iff there is a state s in T_ψ s.t. $(s, s') \in \text{sat}(\psi)$ and $T_\psi \times M, (s, s') \models \mathbf{EG}true$ under the fairness conditions:

$\{\text{sat}(\neg(\varphi_1 \mathbf{U} \varphi_2) \vee \varphi_2)\}$ s.t. $(\varphi_1 \mathbf{U} \varphi_2)$ occurs in ψ .

$\Rightarrow M \models \mathbf{E}\psi$ iff $T_\psi \times M \models \mathbf{E}_f \mathbf{G}true$

$\Rightarrow M \models \neg\psi$ iff $T_\psi \times M \not\models \mathbf{E}_f \mathbf{G}true$

- LTL M.C. reduced to Fair CTL M.C.!!!
- Symbolic OBDD-based techniques apply.

Note

The transition relation R of $T_\psi \times M$ may not be total.

$\Rightarrow \text{Check_FairEG}$ does not consider states without successors, restricting R to the remaining states.

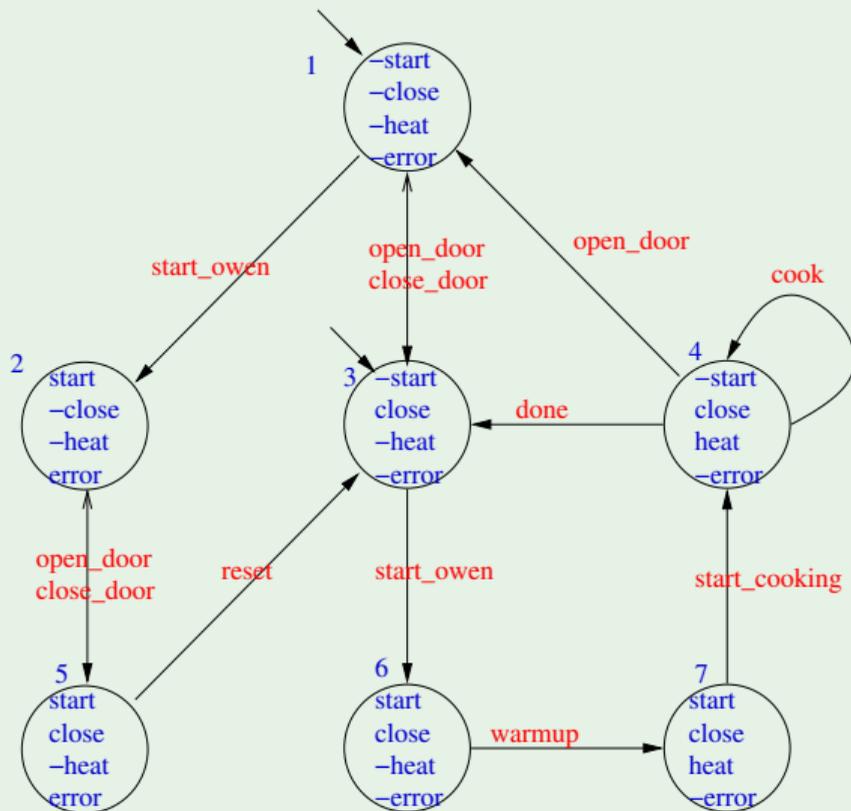
Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

A microwave oven

- 4 state variables: **start, close, heat, error**
- Actions (implicit): start_oven, open_door, close_door, reset, warmup, start_cooking, cook, done
- Error situation: if oven is started while the door is open
- Represented as a Kripke structure (and hence as a OBDD's)

A microwave oven [cont.]



A microwave oven: symbolic representation

- Initial states: $I_M(s, c, h, e) = \neg s \wedge \neg h \wedge \neg e$
- Transition relation: $R_M(s, c, h, e, s', c', h', e') =$ [a simplification of]
 - ($\neg s \wedge \neg c \wedge \neg h \wedge \neg e \wedge \neg s' \wedge c' \wedge \neg h' \wedge \neg e'$) \vee (close_door, no error)
 - ($s \wedge \neg c \wedge \neg h \wedge e \wedge s' \wedge c' \wedge \neg h' \wedge e'$) \vee (close_door, error)
 - ($\neg s \wedge c \wedge \neg h \wedge \neg e \wedge \neg s' \wedge \neg c' \wedge \neg h' \wedge \neg e'$) \vee (open_door, no error)
 - ($s \wedge c \wedge \neg h \wedge e \wedge s' \wedge \neg c' \wedge \neg h' \wedge e'$) \vee (open_door, error)
 - ($\neg s \wedge c \wedge \neg h \wedge \neg e \wedge s' \wedge c' \wedge \neg h' \wedge \neg e'$) \vee (start_oven, no error)
 - ($\neg s \wedge \neg c \wedge \neg h \wedge \neg e \wedge s' \wedge \neg c' \wedge \neg h' \wedge e'$) \vee (start_oven, error)
 - ($s \wedge c \wedge \neg h \wedge e \wedge \neg s' \wedge c' \wedge \neg h' \wedge \neg e'$) \vee (reset)
 - ($s \wedge c \wedge \neg h \wedge \neg e \wedge s' \wedge c' \wedge h' \wedge \neg e'$) \vee (warmup)
 - ($s \wedge c \wedge h \wedge \neg e \wedge \neg s' \wedge c' \wedge h' \wedge \neg e'$) \vee (start_cooking)
 - ($\neg s \wedge c \wedge h \wedge \neg e \wedge \neg s' \wedge c' \wedge h' \wedge \neg e'$) \vee (cook)
 - ($\neg s \wedge c \wedge h \wedge e \wedge \neg s' \wedge c' \wedge \neg h' \wedge \neg e'$) \vee (done)

Note: the third row represents two transitions: $3 \rightarrow 1$ and $4 \rightarrow 1$.

- “necessarily, the oven’s door eventually closes and, till there, the oven does not heat”:

$$M \models \neg \text{heat } \mathbf{U} \text{ close},$$

i.e.,

$$M \models \neg \mathbf{E} \neg (\neg \text{heat } \mathbf{U} \text{ close})$$

Tableau construction for $\psi = \neg(\neg\text{heat } \mathbf{U} \text{ close})$

- $\varphi := \neg\psi = (\neg\text{heat } \mathbf{U} \text{ close})$
- Tableaux expansion: $\psi = \neg(\neg\text{heat } \mathbf{U} \text{ close}) = \neg(\text{close} \vee (\neg\text{heat} \wedge \mathbf{X}(\neg\text{heat } \mathbf{U} \text{ close})))$
- $el(\psi) = el(\varphi) = \{\text{heat}, \text{close}, \mathbf{X}\varphi\}$ ($\{h, c, \mathbf{X}\varphi\}$)
- States:
 - 1 := $\{\neg h, c, \mathbf{X}\varphi\}$, 2 := $\{h, c, \mathbf{X}\varphi\}$, 3 := $\{\neg h, \neg c, \mathbf{X}\varphi\}$,
 - 4 := $\{h, c, \neg\mathbf{X}\varphi\}$, 5 := $\{h, \neg c, \mathbf{X}\varphi\}$, 6 := $\{\neg h, c, \neg\mathbf{X}\varphi\}$,
 - 7 := $\{\neg h, \neg c, \neg\mathbf{X}\varphi\}$, 8 := $\{h, \neg c, \neg\mathbf{X}\varphi\}$

Tableau construction for $\psi = \neg(\neg heat \text{ U } close)$ [cont.]

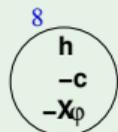
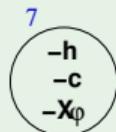
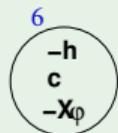
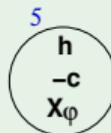
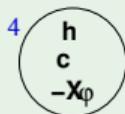
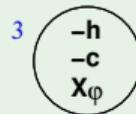
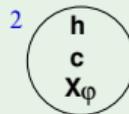
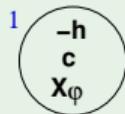


Tableau construction for $\psi = \neg(\neg\text{heat } \mathbf{U} \text{ close})$

- ...

- States:

$$\begin{aligned} 1 &:= \{\neg h, c, \mathbf{X}\varphi\}, & 2 &:= \{h, c, \mathbf{X}\varphi\}, & 3 &:= \{\neg h, \neg c, \mathbf{X}\varphi\}, \\ 4 &:= \{h, c, \neg\mathbf{X}\varphi\}, & 5 &:= \{h, \neg c, \mathbf{X}\varphi\}, & 6 &:= \{\neg h, c, \neg\mathbf{X}\varphi\}, \\ 7 &:= \{\neg h, \neg c, \neg\mathbf{X}\varphi\}, & 8 &:= \{h, \neg c, \neg\mathbf{X}\varphi\} \end{aligned}$$

- $\text{sat}()$:

$$\begin{aligned} \text{sat}(h) &= \{2, 4, 5, 8\} \implies \text{sat}(\neg h) = \{1, 3, 6, 7\}, \\ \text{sat}(c) &= \{1, 2, 4, 6\} \implies \text{sat}(\neg c) = \{3, 5, 7, 8\}, \\ \text{sat}(\mathbf{X}\varphi) &= \{1, 2, 3, 5\} \implies \text{sat}(\neg\mathbf{X}\varphi) = \{4, 6, 7, 8\}, \\ \text{sat}(\varphi) &= \text{sat}(c) \cup (\text{sat}(\neg h) \cap \text{sat}(\mathbf{X}(\neg h \mathbf{U} c))) = \{1, 2, 3, 4, 6\} \\ \implies \text{sat}(\psi) &= \text{sat}(\neg\varphi) = \{5, 7, 8\} \end{aligned}$$

Tableau construction for $\psi = \neg(\neg heat \mathbf{U} close)$ [cont.]

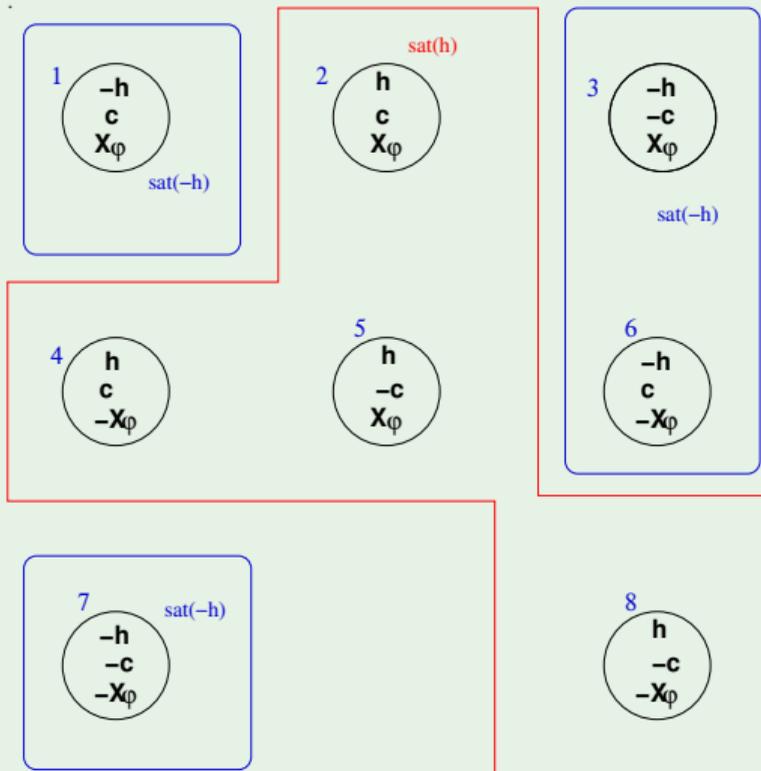


Tableau construction for $\psi = \neg(\neg heat \text{ U } close)$ [cont.]

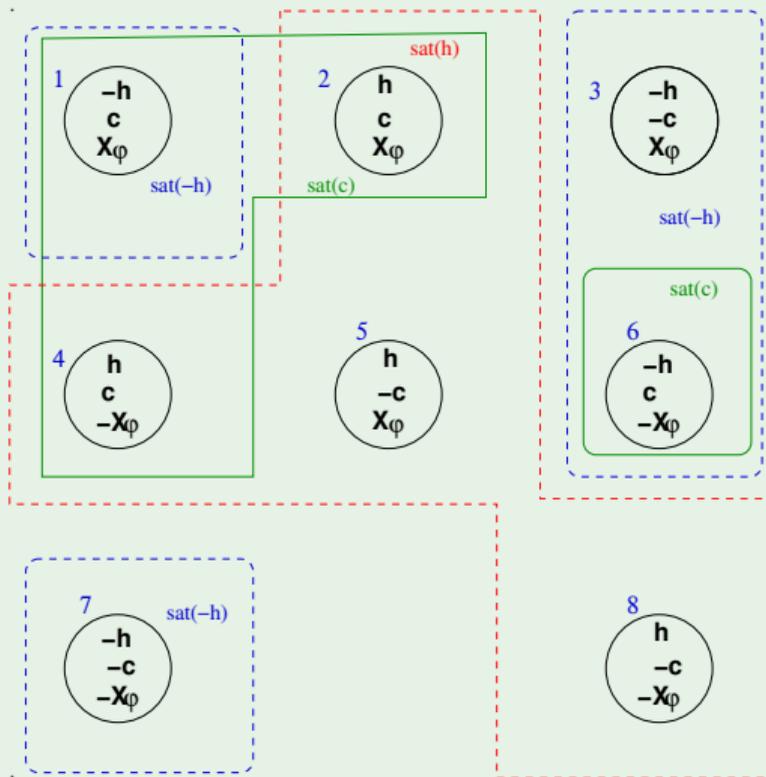


Tableau construction for $\psi = \neg(\neg heat \text{ U } close)$ [cont.]

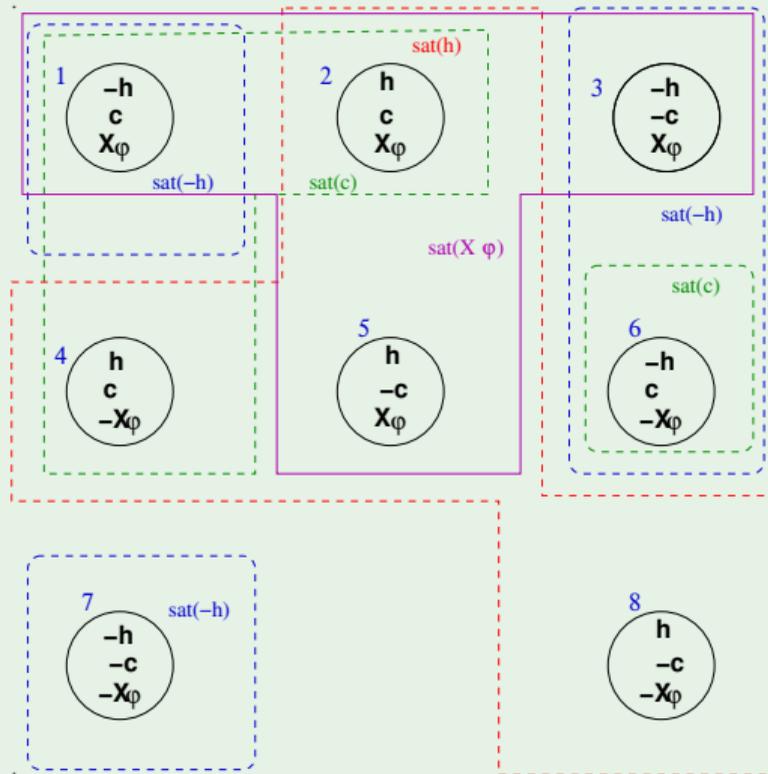


Tableau construction for $\psi = \neg(\neg heat \text{ U } close)$ [cont.]

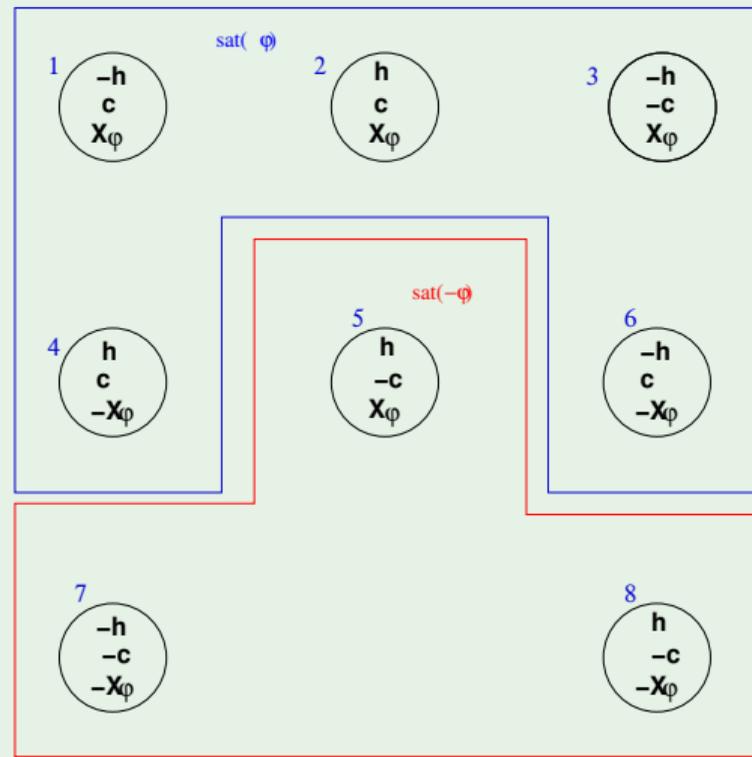


Tableau construction for $\psi = \neg(\neg heat \mathbf{U} close)$ [cont.]

- ...

- $sat()$:

$$sat(h) = \{2, 4, 5, 8\} \implies sat(\neg h) = \{1, 3, 6, 7\},$$

$$sat(c) = \{1, 2, 4, 6\} \implies sat(\neg c) = \{3, 5, 7, 8\},$$

$$sat(\mathbf{X}\varphi) = \{1, 2, 3, 5\} \implies sat(\neg\mathbf{X}\varphi) = \{4, 6, 7, 8\},$$

$$sat(\varphi) = sat(c) \cup (sat(\neg h) \cap sat(\mathbf{X}(\neg h \mathbf{U} c))) = \{1, 2, 3, 4, 6\}$$

- Initial states I : $sat(\psi) = sat(\neg\varphi) = \{5, 7, 8\}$

- **Transition Relation R :**

- add an edge from every state in $sat(\mathbf{X}\varphi)$ to every state in $sat(\varphi)$
- add an edge from every state in $sat(\neg\mathbf{X}\varphi)$ to every state in $sat(\neg\varphi)$

Tableau construction for $\psi = \neg(\neg heat \mathbf{U} close)$ [cont.]

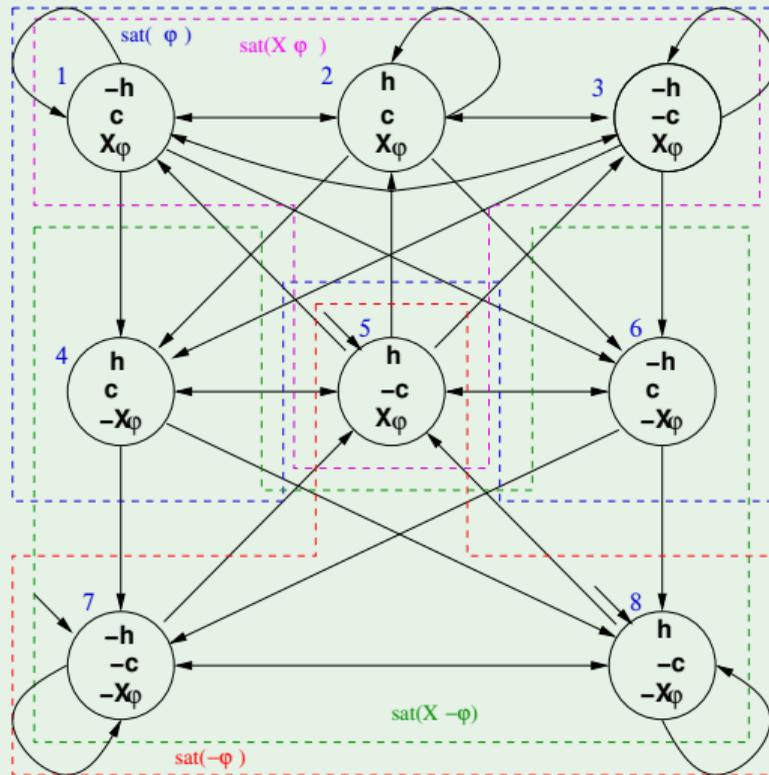
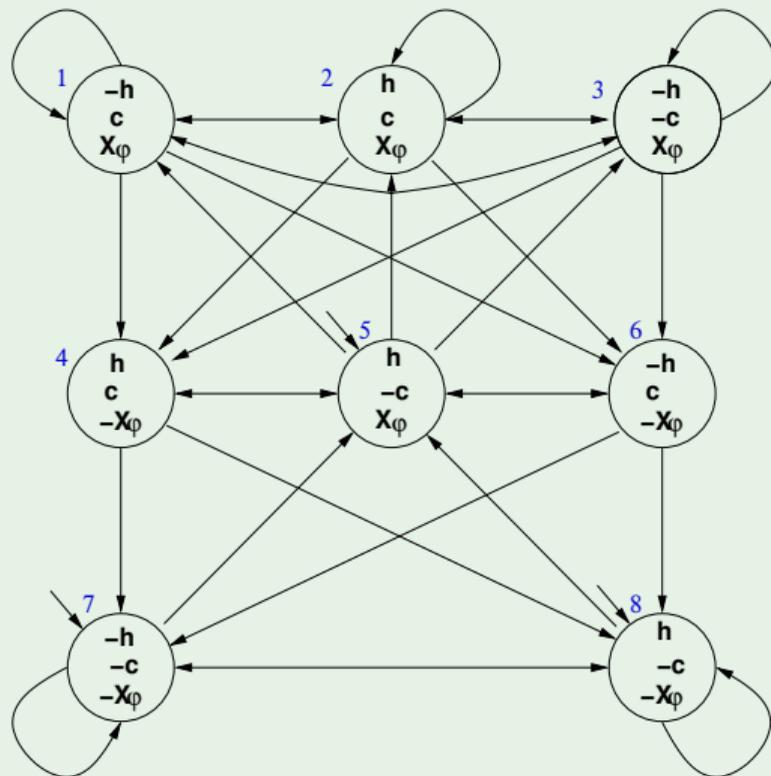


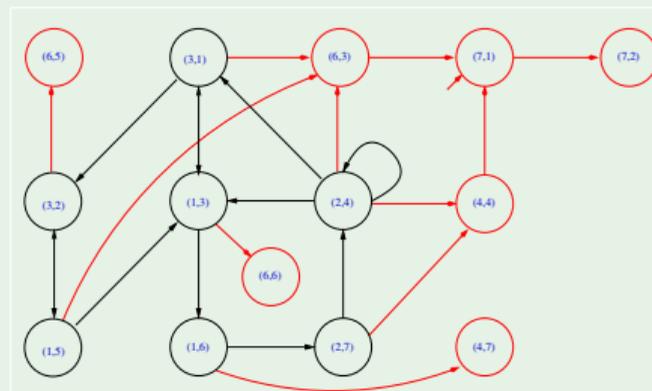
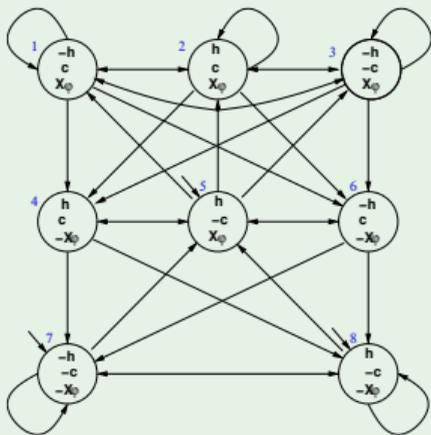
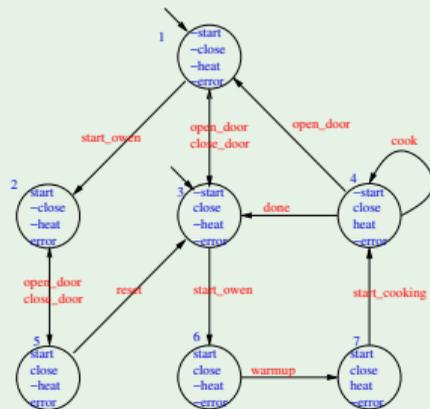
Tableau construction for $\psi = \neg(\neg heat \mathbf{U} close)$ [cont.]



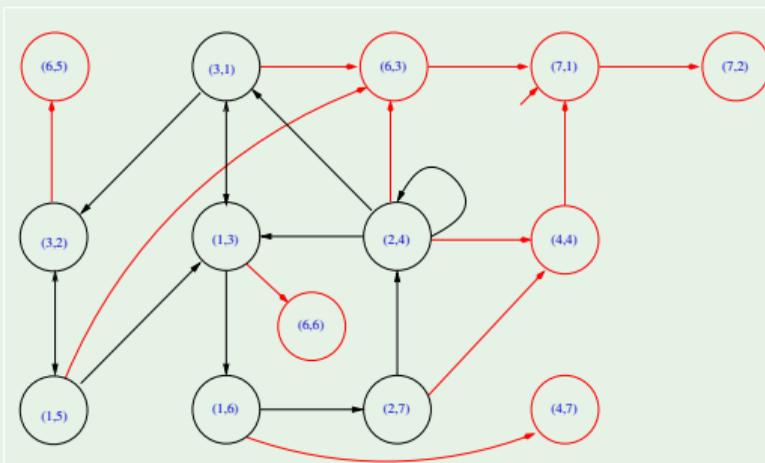
Symbolic representation of T_ψ , s.t. $\psi := \neg(\neg h \mathbf{U} c)$

- State variables: h , c and x and primed versions h' , c' and x'
[x is a Boolean label for $\mathbf{X}(\neg h \mathbf{U} c)$]
- Initial states: $I_{T_\psi} = \text{sat}(\psi)$
 $\implies I(h, c, x) = \neg(c \vee (\neg h \wedge x))$
- Transition Relation: $R_{T_\psi} = \bigwedge_{\mathbf{x}\varphi_i \in \text{el}(\psi)} (\text{sat}(\mathbf{X}\varphi_i) \leftrightarrow \text{sat}'(\varphi_i))$
 $\implies R_{T_\psi}(h, c, x, h', c', x') = x \leftrightarrow (c' \vee (\neg h' \wedge x'))$
- Fairness Property: (due to negative polarity of $(\neg h \mathbf{U} c)$ in ψ):
 $F_{T_\psi}(h, c, x) = \top$

$$\text{Product } P = T_\psi \times M$$



Product $P = T_\psi \times M$ [cont.]



- $P = T_\psi \times M$ (reachable states only)
- compute **[EGtrue]** (e.g. by Emerson-Lei):
 - ⇒ states (4, 4), (4, 7), (6, 3), (6, 5), (6, 6), (7, 1), (7, 2) are not part of a (fair) infinite path
 - ⇒ no initial states in **[EGtrue]** ((7.1) has been removed).
 - ⇒ $T_\psi \times M \not\models \mathbf{EGtrue}$
 - ⇒ **Property verified!**
- N.B.: fairness condition \top irrelevant here

Product $P = T_\psi \times M$: symbolic representation

- Initial states: $I(s, c, h, e, x) = (\neg s \wedge \neg h \wedge \neg e) \wedge \neg(c \vee (\neg h \wedge x)) = \neg s \wedge \neg h \wedge \neg e \wedge \neg c \wedge \neg x$
- Transition relation: $R(s, c, h, e, x, s', c', h', e', x') =$ (an OBDD for $(x \leftrightarrow (c' \vee (\neg h' \wedge x')))) \wedge$
 - $(\neg s \wedge \neg c \wedge \neg h \wedge \neg e \wedge \neg s' \wedge c' \wedge \neg h' \wedge \neg e') \vee$ (close_door, no error)
 - $(s \wedge \neg c \wedge \neg h \wedge e \wedge s' \wedge c' \wedge \neg h' \wedge e') \vee$ (close_door, error)
 - $(\neg s \wedge c \wedge \neg h \wedge \neg e \wedge \neg s' \wedge \neg c' \wedge \neg h' \wedge \neg e') \vee$ (open_door, no error)
 - $(s \wedge c \wedge \neg h \wedge e \wedge s' \wedge \neg c' \wedge \neg h' \wedge e') \vee$ (open_door, error)
 - $(\neg s \wedge c \wedge \neg h \wedge \neg e \wedge s' \wedge c' \wedge \neg h' \wedge \neg e') \vee$ (start_oven, no error)
 - $(\neg s \wedge \neg c \wedge \neg h \wedge \neg e \wedge s' \wedge \neg c' \wedge \neg h' \wedge e') \vee$ (start_oven, error)
 - $(s \wedge c \wedge \neg h \wedge e \wedge \neg s' \wedge c' \wedge \neg h' \wedge \neg e') \vee$ (reset)
 - $(s \wedge c \wedge \neg h \wedge \neg e \wedge s' \wedge c' \wedge h' \wedge \neg e') \vee$ (warmup)
 - $(s \wedge c \wedge h \wedge \neg e \wedge \neg s' \wedge c' \wedge h' \wedge \neg e') \vee$ (start_cooking)
 - $(\neg s \wedge c \wedge h \wedge \neg e \wedge \neg s' \wedge c' \wedge h' \wedge \neg e') \vee$ (cook)
 - $(\neg s \wedge c \wedge h \wedge \neg e \wedge \neg s' \wedge c' \wedge \neg h' \wedge \neg e')$ (done)

[EGtrue]: symbolic representation

- Emerson-Lei returns (an OBDD equivalent to):

EGtrue =

$$(\neg s \wedge \neg c \wedge \neg h \wedge \neg e \wedge x) \vee \quad (3, 1)$$

$$(s \wedge \neg c \wedge \neg h \wedge e \wedge x) \vee \quad (3, 2)$$

$$(\neg s \wedge c \wedge \neg h \wedge \neg e \wedge x) \vee \quad (1, 3)$$

$$(\neg s \wedge c \wedge h \wedge \neg e \wedge x) \vee \quad (2, 4)$$

$$(s \wedge c \wedge \neg h \wedge e \wedge x) \vee \quad (1, 5)$$

$$(s \wedge c \wedge \neg h \wedge \neg e \wedge x) \vee \quad (1, 5)$$

$$(s \wedge c \wedge h \wedge \neg e \wedge x) \vee \quad (2, 7)$$

...

(other unreachable states)

- Initial states: $I(s, c, h, e, x) = \neg s \wedge \neg h \wedge \neg e \wedge \neg c \wedge \neg x$

$$\Rightarrow I(s, c, h, e, x) \not\models \mathbf{EGtrue}$$

$$\Rightarrow I \not\subseteq [\mathbf{EGtrue}]$$

$$\Rightarrow T_\psi \times M \not\models \mathbf{EGtrue}$$

$$\Rightarrow \text{Property verified!}$$



The property verified is...

Outline

- 1 CTL Symbolic Model Checking
 - Symbolic Representation of Systems
 - Symbolic CTL MC
 - Symbolic Fair CTL MC
 - A simple example
- 2 The Symbolic Approach to LTL Model Checking
 - General Ideas
 - Compute the Tableau T_ψ
 - Compute the Product $M \times T_\psi$
 - Check the Emptiness of $\mathcal{L}(M \times T_\psi)$
- 3 A Complete Example
- 4 Exercises

Ex: Symbolic CTL Model Checking

Given the following finite state machine expressed in NuSMV input language:

```
MODULE main
VAR v1 : boolean; v2 : boolean;
INIT (!v1 & !v2)
TRANS (next(v1) <-> !v1) & (next(v2) <-> (v1<->v2))
```

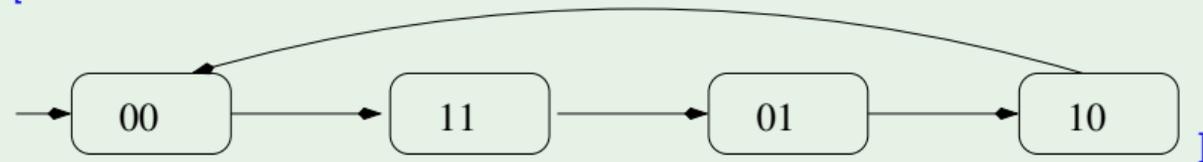
and consider the property $P \stackrel{\text{def}}{=} (v_1 \wedge v_2)$. Write:

- the Boolean formulas $I(v_1, v_2)$ and $T(v_1, v_2, v'_1, v'_2)$ representing respectively the initial states and the transition relation of M .

[Solution: $I(v_1, v_2)$ is $(\neg v_1 \wedge \neg v_2)$, $T(v_1, v_2, v'_1, v'_2)$ is $(v'_1 \leftrightarrow \neg v_1) \wedge (v'_2 \leftrightarrow (v_1 \leftrightarrow v_2))$]

- the graph representing the FSM. (Assume the notation " $v_1 v_2$ " for labeling the states: e.g. "10" means " $v_1 = 1, v_2 = 0$ ".)

[Solution:



Ex: Symbolic CTL Model Checking (cont.)

- the Boolean formula representing symbolically **EXP**. [The formula must be computed symbolically, not simply inferred from the graph of the previous question!]

[Solution:

$$\begin{aligned}\mathbf{EX}(P) &= \exists v'_1, v'_2. (T(v_1, v_2, v'_1, v'_2) \wedge P(v'_1, v'_2)) \\ &= \exists v'_1, v'_2. ((v'_1 \leftrightarrow \neg v_1) \wedge (v'_2 \leftrightarrow (v_1 \leftrightarrow v_2))) \wedge \underbrace{(v'_1 \wedge v'_2)}_{\implies v'_1=T, v'_2=T} \\ &= \underbrace{v'_1=T, v'_2=T}_{(\neg v_1 \wedge \neg v_2)} \vee \perp \vee \perp \vee \perp \\ &= (\neg v_1 \wedge \neg v_2)\end{aligned}$$

.]

Ex: Symbolic CTL Model Checking

Given the following finite state machine expressed in NuSMV input language:

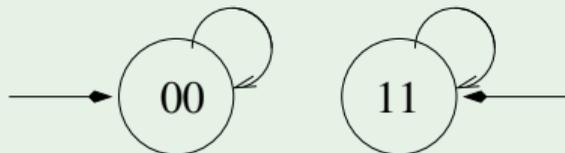
```
VAR    v1 : boolean;  v2 : boolean;
INIT   init(v1) <-> init(v2)
TRANS  (v1 <-> next(v2)) & (v2 <-> next(v1));
```

write:

- the Boolean formulas $I(v_1, v_2)$ and $T(v_1, v_2, v'_1, v'_2)$ representing the initial states and the transition relation of M respectively.

[Solution: $I(v_1, v_2)$ is $(v_1 \leftrightarrow v_2)$, $T(v_1, v_2, v'_1, v'_2)$ is $(v_1 \leftrightarrow v'_1) \wedge (v_2 \leftrightarrow v'_2)$]

- the graph representing the FSM. (Assume the notation " $v_1 v_2$ " for labeling the states. E.g., "10" means " $v_1 = 1, v_2 = 0$ ".)



[Solution:

]

Ex: Symbolic CTL Model Checking (cont.)

- the Boolean formula $R^1(v'_1, v'_2)$ representing the set of states which can be reached after exactly 1 step.
NOTE: this must be computed symbolically, not simply deduced from the graph of question b).

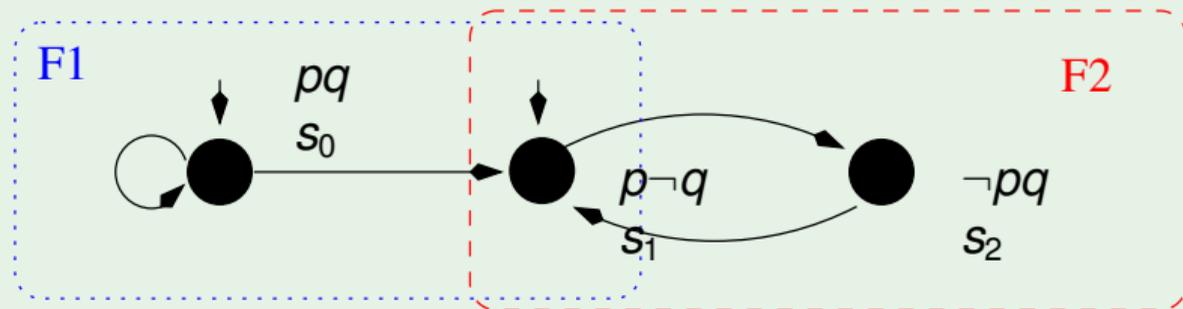
[Solution:

$$\begin{aligned}R^1(v'_1, v'_2) &= \exists v_1, v_2. (I(v_1, v_2) \wedge T(v_1, v_2, v'_1, v'_2)) \\ &= \exists v_1, v_2. ((v_1 \leftrightarrow v_2) \wedge (v_1 \leftrightarrow v'_2) \wedge (v_2 \leftrightarrow v'_1)) \\ &= ((v_1 \leftrightarrow v_2) \wedge (v_1 \leftrightarrow v'_2) \wedge (v_2 \leftrightarrow v'_1))[v_1 = \perp, v_2 = \perp] \vee \\ &\quad ((v_1 \leftrightarrow v_2) \wedge (v_1 \leftrightarrow v'_2) \wedge (v_2 \leftrightarrow v'_1))[v_1 = \perp, v_2 = \top] \vee \\ &\quad ((v_1 \leftrightarrow v_2) \wedge (v_1 \leftrightarrow v'_2) \wedge (v_2 \leftrightarrow v'_1))[v_1 = \top, v_2 = \perp] \vee \\ &\quad ((v_1 \leftrightarrow v_2) \wedge (v_1 \leftrightarrow v'_2) \wedge (v_2 \leftrightarrow v'_1))[v_1 = \top, v_2 = \top] \\ &= (\neg v'_1 \wedge \neg v'_2) \vee \perp \vee \perp \vee (v'_1 \wedge v'_2) \\ &= (\neg v'_1 \wedge \neg v'_2) \vee (v'_1 \wedge v'_2) \\ &= (v'_1 \leftrightarrow v'_2)\end{aligned}$$

.]

Ex: Fair CTL Model Checking

Consider the following *fair* Kripke Model M :

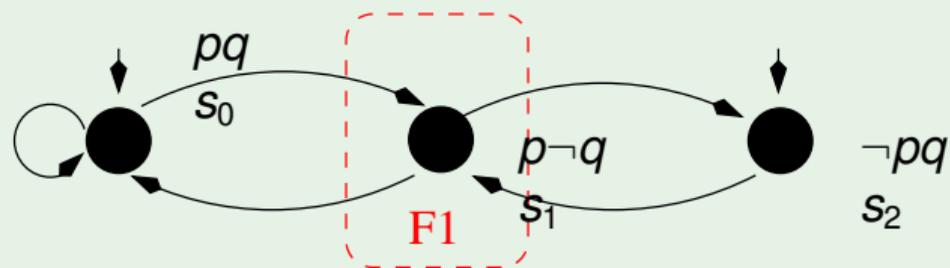


For each of the following facts, say if it is true or false in CTL.

- (a) $M \models \mathbf{AF}\neg p$
[Solution: true]
- (b) $M \models \mathbf{A}(p\mathbf{U}\neg q)$
[Solution: true]
- (c) $M \models \mathbf{AX}\neg q$
[Solution: false]
- (d) $M \models \mathbf{AGAF}\neg p$
[Solution: true]

Ex: Fair CTL Model Checking

Consider the following *fair* Kripke Model M :



For each of the following facts, say if it is true or false in CTL.

- (a) $M \models \mathbf{EF}(p \wedge q)$
[Solution: true]
- (b) $M \models \mathbf{AGAF}p$
[Solution: true]
- (c) $M \models \mathbf{AF}\neg q$
[Solution: true]
- (d) $M \models \mathbf{AG}(\neg p \vee \neg q)$
[Solution: false]

Ex: Symbolic LTL Model Checking

Given the following LTL formula: $\varphi \stackrel{\text{def}}{=} \neg((\mathbf{GF}p \wedge \mathbf{GF}q) \rightarrow \mathbf{GF}r)$

(a) Compute the Negative Normal Form of φ ($NNF(\varphi)$).

$$\begin{aligned}
 \varphi &\iff \neg((\mathbf{GF}p \wedge \mathbf{GF}q) \rightarrow \mathbf{GF}r) \\
 \text{[Solution: } &\iff \neg(\neg(\mathbf{GF}p \wedge \mathbf{GF}q) \vee \mathbf{GF}r) \\
 &\iff (\mathbf{GF}p \wedge \mathbf{GF}q \wedge \neg\mathbf{GF}r) \\
 &\iff (\mathbf{GF}p \wedge \mathbf{GF}q \wedge \mathbf{FG}\neg r) \iff NNF(\varphi) \\
 \text{]} &
 \end{aligned}$$

(b) Compute the set of elementary subformulas of φ .

[Solution: First write the formula in terms of **X** and **U**'s (write "**F** ψ " for "**TU** ψ "):]

$$\begin{aligned}
 \varphi &\iff \neg((\mathbf{GF}p \wedge \mathbf{GF}q) \rightarrow \mathbf{GF}r) \\
 &\iff \neg((\neg\mathbf{F}\neg\mathbf{F}p \wedge \neg\mathbf{F}\neg\mathbf{F}q) \rightarrow \neg\mathbf{F}\neg\mathbf{F}r)
 \end{aligned}$$

$$el(\mathbf{F}\neg\mathbf{F}p) = \{\mathbf{XF}\neg\mathbf{F}p\} \cup el(\neg\mathbf{F}p) = \{\mathbf{XF}\neg\mathbf{F}p\} \cup \{\mathbf{XF}p\} \cup el(p) = \{\mathbf{XF}\neg\mathbf{F}p, \mathbf{XF}p, p\}.$$

$$\begin{aligned}
 \text{Hence: } el(\varphi) &= el(\neg((\neg\mathbf{F}\neg\mathbf{F}p \wedge \neg\mathbf{F}\neg\mathbf{F}q) \rightarrow \neg\mathbf{F}\neg\mathbf{F}r)) \\
 &= el(\mathbf{F}\neg\mathbf{F}p) \cup el(\mathbf{F}\neg\mathbf{F}q) \cup el(\mathbf{F}\neg\mathbf{F}r) \\
 &= \{\mathbf{XF}\neg\mathbf{F}p, \mathbf{XF}p, p, \mathbf{XF}\neg\mathbf{F}q, \mathbf{XF}q, q, \mathbf{XF}\neg\mathbf{F}r, \mathbf{XF}r, r\}
 \end{aligned}$$

(c) What is the (maximum) number of states of a fair Kripke Model representing φ ?

[Solution: By definition it is $2^{|el(\varphi)|} = 2^9 = 512$.]

Ex: Symbolic LTL Model Checking

Given the following LTL formula $\psi \stackrel{\text{def}}{=} \neg \mathbf{F} \neg p$, compute and draw the tableau \mathcal{T}_ψ of ψ . [Solution:

(i) The set of elementary subformulas of ψ is $el(\psi) \stackrel{\text{def}}{=} \{p, \mathbf{X}\mathbf{F}\neg p\}$. Hence, the set of states is

$$\{s_1 : (p, \neg \mathbf{X}\mathbf{F}\neg p), s_2 : (p, \mathbf{X}\mathbf{F}\neg p), s_3 : (\neg p, \neg \mathbf{X}\mathbf{F}\neg p), s_4 : (\neg p, \mathbf{X}\mathbf{F}\neg p)\}$$

(ii) The set of initial states of \mathcal{T}_ψ is $sat(\psi) \stackrel{\text{def}}{=} S \setminus (sat(\neg p) \cup sat(\mathbf{X}\mathbf{F}\neg p)) = \{s_1\}$.

(iii) Since s_1 is the only state in $sat(\neg \mathbf{F}\neg p)$, then s_1 is the only successor of itself, so that the only relevant transition is a self-loop over s_1 .

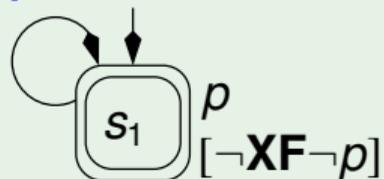
(One can also —un-necessarily— draw all transitions from states where $\neg \mathbf{X}\mathbf{F}\neg p$ holds into $\{s_1\}$ and from from states where $\mathbf{X}\mathbf{F}\neg p$ holds into $\{s_2, s_3, s_4\}$.)

(iv) There is one **U**-subformula, $\mathbf{F}\neg p$, so that there is one fairness condition defined as $sat(\neg \mathbf{F}\neg p \vee \neg p)$. Since $\mathbf{F}\neg p$ is false in s_1 , then s_1 is part of the fairness condition. [Alternatively: there is no **positive U**-subformula, so that we must add a **AGAF^T** fairness condition, which is equivalent to say that all states belong to the fairness condition.]

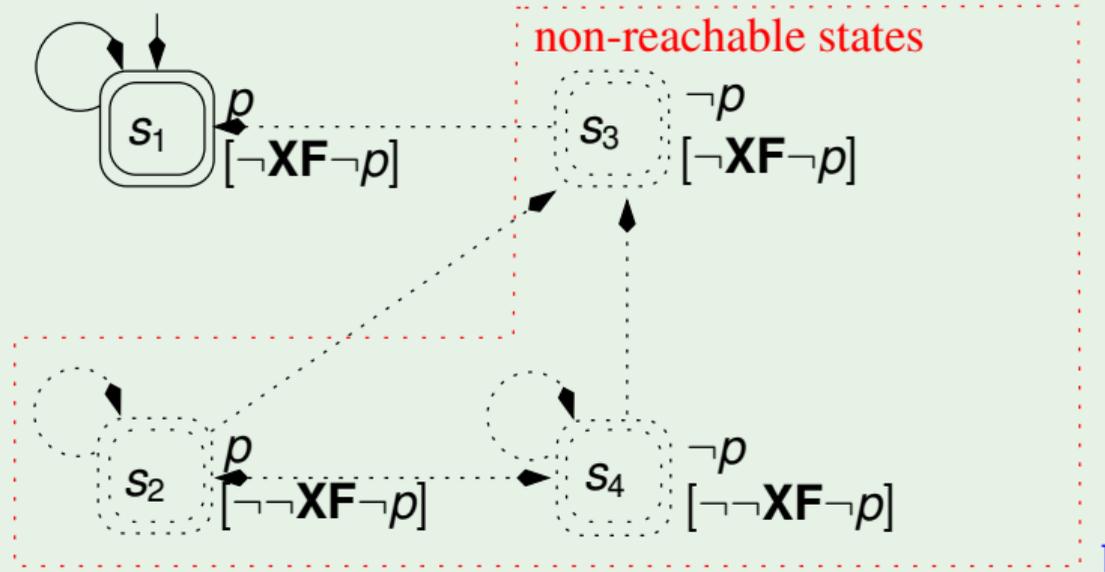
]

Ex: Symbolic LTL Model Checking (cont.)

[Solution:



or, alternatively without simplifications:



Ex: Symbolic LTL Model Checking

Given the following LTL formula $\psi \stackrel{\text{def}}{=} \mathbf{G}p$, compute and draw the tableau \mathcal{T}_ψ of ψ . [Without converting anything into **X**, **U**].
[Solution:

(i) The set of elementary subformulas of ψ is $el(\psi) \stackrel{\text{def}}{=} \{p, \mathbf{XG}p\}$. Hence, the set of states is

$$\{s_1 : (p, \mathbf{XG}p), s_2 : (p, \neg\mathbf{XG}p), s_3 : (\neg p, \mathbf{XG}p), s_4 : (\neg p, \neg\mathbf{XG}p)\}$$

(ii) The set of initial states of \mathcal{T}_ψ is $sat(\psi) \stackrel{\text{def}}{=} sat(p) \cap sat(\mathbf{XG}p) = \{s_1\}$.

(iii) Since s_1 is the only state in $sat(\mathbf{G}p)$, then s_1 is the only successor of itself, so that the only relevant transition is a self-loop over s_1 .

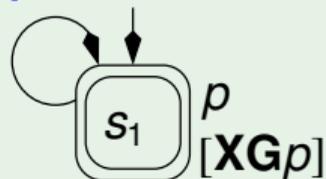
(One can also —un-necessarily— draw all transitions from states where $\mathbf{XG}p$ holds into $\{s_1\}$ and from from states where $\neg\mathbf{XG}p$ holds into $\{s_2, s_3, s_4\}$.)

(iv) Since there is no “**U**” subformula, we must add a **AGAF** \top fairness condition, which is equivalent to say that all states belong to the fairness condition.

]

Ex: Symbolic LTL Model Checking (cont.)

[Solution:



or, alternatively without simplifications:

