Introduction to Model Checking

René Thiemann

Institute of Computer Science
University of Innsbruck

WS 2011/2012

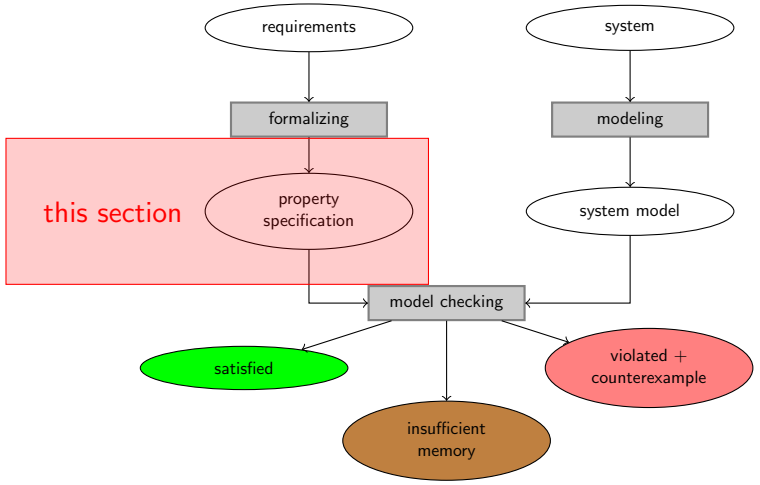
Outline

- LTL - Linear Time Logic
  - Syntax and Semantics
  - Equivalences
- LTL Model Checking
  - Overview
  - Transforming LTL into GNBAs
  - Complexity of LTL Model Checking

Outline

- LTL - Linear Time Logic
  - Syntax and Semantics
  - Equivalences
- LTL Model Checking
  - Overview
  - Transforming LTL into GNBAs
  - Complexity of LTL Model Checking

Model checking overview



Motivation

- so far, we used GNBAs as requirements specification

$\Rightarrow$ larger specification become unreadable
(think of an intersection GNBA of four properties)

$\Rightarrow$ need nicer way to precisely state requirements

$\Rightarrow$ LTL

Syntax of Linear Temporal Logic

modal logic over infinite sequences [Pnueli 1977]

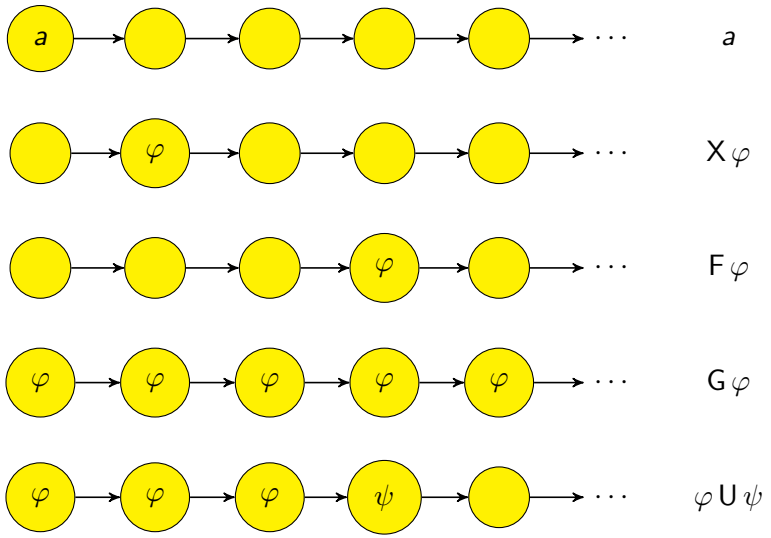
- propositional logic
  - true
  - $a, \text{paid}, \text{sprite}, \ldots \in AP$ — atomic proposition
  - $\neg\varphi$ and $\varphi \wedge \psi$ — negation and conjunction
- temporal operators
  - $\mathsf{X}\,\varphi$ — neXt step fulfills φ
  - $\mathsf{F}\,\varphi$ — sometimes in the Future φ will hold
  - $\mathsf{G}\,\varphi$ — φ Globally holds
  - $\varphi\,\mathsf{U}\,\psi$ — φ holds Until ψ holds

linear temporal logic is a logic for describing linear time properties

precedence order: the unary operators bind stronger than the binary ones.
$\neg$ and X bind equally strong. U takes precedence over $\wedge$, $\vee$, and $\Rightarrow$

$$\neg, \mathsf{X}, \mathsf{F}, \mathsf{G} \quad > \quad \mathsf{U} \quad > \quad \wedge, \ \vee \ , \Rightarrow, \Leftrightarrow$$

Intuitive semantics



Derived Boolean operators

$$\begin{aligned}
\text{false} &\equiv \neg\text{true} \\
\varphi \vee \psi &\equiv \neg(\neg\varphi \wedge \neg\psi) \\
\varphi \Rightarrow \psi &\equiv \neg\varphi \vee \psi \\
\varphi \Leftrightarrow \psi &\equiv (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi) \\
\varphi \oplus \psi &\equiv \neg(\varphi \Leftrightarrow \psi)
\end{aligned}$$

Practical properties in LTL

- Reachability $\mathsf{F}\,\psi$
- Safety $\mathsf{G}\,\neg\varphi$
- Liveness $\mathsf{G}\,(\varphi \Rightarrow \mathsf{F}\,\psi)$ and others
- Fairness $\mathsf{G}\,\mathsf{F}\,\varphi$ and others

Properties of a traffic light

- the light is never red and green: $\neg\,(\text{red} \wedge \text{green})$
- whenever the light is red, it cannot become green immediately afterwards:

$$\text{red} \;\Rightarrow\; \neg\,\mathsf{X}\,\text{green}$$

- eventually, the light becomes green: $\mathsf{F}\,\text{green}$
- green holds until red appears problem in req.: should mention orange in between and at sometime orange appears; moreover, the red is later than the orange

$$(\text{green}\;\mathsf{U}\;\text{red}) \wedge \mathsf{F}\,(\text{orange} \wedge \mathsf{X}\,\mathsf{F}\,\text{red})$$

consider $\{\text{green}\}\,\{\text{red}\}\,\{\text{orange}\}\,\{\text{red}\}\ldots$

$$\text{green}\;\mathsf{U}\,(\text{orange} \wedge \mathsf{X}\,(\text{orange}\;\mathsf{U}\;\text{red}))$$

most of these requirements do not correspond to the specifications

Requirements from Chapter 2, slide 27

1. always at most one traffic light is showing green

$$\mathsf{G}\,(\neg\text{green1} \vee \neg\text{green2})$$

2. green cannot be directly followed by red

$$\mathsf{G}\,(\text{green} \Rightarrow \mathsf{X}\,\neg\text{red})$$

3. we will see green infinitely often

$$\mathsf{G}\,\mathsf{F}\,\text{green}$$

4. whenever we select sprite then later on we will get a sprite

$$\mathsf{G}\,(\text{select} \Rightarrow \mathsf{X}\,\mathsf{F}\,\text{get})$$

Semantics over words

the language induced by LTL formula φ over $AP = \{a_1, \ldots, a_n\}$ is:

$$\mathcal{L}(\varphi) \;=\; \left\{ w \in \left(2^{AP}\right)^{\omega} \mid w \models \varphi \right\}, \text{where } \models \text{ is defined as follows:}$$

(let $w = A_0A_1A_2\ldots$ and $w[i..] = A_iA_{i+1}A_{i+2}\ldots$ is the suffix of w from index i on)

$$\begin{array}{lcll}
w & \models & \text{true} & \\
w & \models & a_i & \text{iff} \quad a_i \in A_0 \\
w & \models & \varphi_1 \wedge \varphi_2 & \text{iff} \quad w \models \varphi_1 \text{ and } w \models \varphi_2 \\
w & \models & \neg\,\varphi & \text{iff} \quad w \not\models \varphi \\
w & \models & \mathsf{X}\,\varphi & \text{iff} \quad w[1..] = A_1A_2A_3\ldots \models \varphi \\
w & \models & \varphi_1\,\mathsf{U}\,\varphi_2 & \text{iff} \quad \exists j \geqslant 0.\ w[j..] \models \varphi_2 \text{ and } \forall 0 \leqslant i < j : w[i..] \models \varphi_1 \\
w & \models & \mathsf{F}\,\varphi & \text{iff} \quad \exists j \geqslant 0.\ w[j..] \models \varphi \\
w & \models & \mathsf{G}\,\varphi & \text{iff} \quad \forall j \geqslant 0.\ w[j..] \models \varphi
\end{array}$$

Often used constructs

- $w \models \mathsf{G}\,\mathsf{F}\,\varphi$ iff $\forall i : w[i..] \models \mathsf{F}\,\varphi$
  iff $\forall i\ \exists j : w[i..][j..] \models \varphi$
  iff $\forall i\ \exists j : w[i+j..] \models \varphi$
  iff $\forall i\ \exists j \geqslant i : w[j..] \models \varphi$
  iff infinitely often φ is satisfied in w

- $w \models \mathsf{F}\,\mathsf{G}\,\varphi$ iff $\exists i\ \forall j \geqslant i : w[j..] \models \varphi$ iff
  from some point onwards φ is always satisfied in w

Semantics for Transition Systems

semantics is defined via set inclusion as indicated in previous section, so a system satisfies a formula iff all traces are allowed w.r.t. the formula:

$$TS \models \varphi \text{ iff } \mathit{Traces}(TS) \subseteq \mathcal{L}(\varphi)$$

A note on negations

for trace w, it holds $w \models \varphi$ iff $w \not\models \neg\varphi$ since

$$\mathcal{L}(\neg\varphi) = (2^{AP})^{\omega} \setminus \mathcal{L}(\varphi)$$

but: $TS \not\models \varphi$ and $TS \models \neg\varphi$ are not equivalent in general
usually it holds: $TS \models \neg\varphi$ implies $TS \not\models \varphi$ but not always the reverse!
usually $=$ TS must contain at least one trace

example: a transition system for which $TS \not\models \mathsf{F}\,a$ and $TS \not\models \neg\mathsf{F}\,a$

Equivalence of LTL formulas

Definition (Equivalence)

LTL formulas φ, ψ are equivalent, denoted $\varphi \equiv \psi$, iff

$$\mathcal{L}(\varphi) = \mathcal{L}(\psi)$$

equivalences are useful to

- simplify formulas
- reduce the number of operators in algorithms / proofs

proving a new equivalence

- is a proof of set-equality $\mathcal{L}(\varphi) = \mathcal{L}(\psi)$
- can be done by showing for all words w that $w \in \mathcal{L}(\varphi)$ iff $w \in \mathcal{L}(\psi)$

Deriving F and G

- $\mathsf{F}\,\varphi \equiv \text{true}\,\mathsf{U}\,\varphi$

$$
\begin{array}{ll}
 & w \models \mathsf{F}\,\varphi \\
\text{iff} & \exists j \geqslant 0.\ w[j..] \models \varphi \\
\text{iff} & \exists j \geqslant 0.\ w[j..] \models \varphi \text{ and } \forall 0 \leqslant i < j : w[i..] \models \text{true} \\
\text{iff} & w \models \text{true}\,\mathsf{U}\,\varphi
\end{array}
$$

- $\mathsf{G}\,\varphi \equiv \neg(\mathsf{F}\,\neg\varphi) \equiv \neg(\text{true}\,\mathsf{U}\,\neg\varphi)$

$$
\begin{array}{ll}
 & w \models \mathsf{G}\,\varphi \\
\text{iff} & \forall j \geqslant 0.\ w[j..] \models \varphi \\
\text{iff} & \textit{not}\ (\textit{not}\ \forall j \geqslant 0.\ w[j..] \models \varphi) \\
\text{iff} & \textit{not}\ \exists j \geqslant 0.\ w[j..] \not\models \varphi \\
\text{iff} & \textit{not}\ w[j..] \models \mathsf{F}\,\neg\varphi \\
\text{iff} & w \models \neg\mathsf{F}\,\neg\varphi
\end{array}
$$

$\Rightarrow$ the temporal operators X and U suffice to express every LTL formula

Duality and idempotence laws

duality:

$$
\begin{array}{rcl}
\neg\mathsf{G}\,\varphi & \equiv & \mathsf{F}\,\neg\varphi \\
\neg\mathsf{F}\,\varphi & \equiv & \mathsf{G}\,\neg\varphi \\
\neg\mathsf{X}\,\varphi & \equiv & \mathsf{X}\,\neg\varphi
\end{array}
$$

idempotency:

$$
\begin{array}{rcl}
\mathsf{G}\,\mathsf{G}\,\varphi & \equiv & \mathsf{G}\,\varphi \\
\mathsf{F}\,\mathsf{F}\,\varphi & \equiv & \mathsf{F}\,\varphi \\
\varphi\,\mathsf{U}\,(\varphi\,\mathsf{U}\,\psi) & \equiv & \varphi\,\mathsf{U}\,\psi \\
(\varphi\,\mathsf{U}\,\psi)\,\mathsf{U}\,\psi & \equiv & \varphi\,\mathsf{U}\,\psi
\end{array}
$$

Absorption and distributive laws

absorption:

$$
\begin{array}{rcl}
\mathsf{F}\,\mathsf{G}\,\mathsf{F}\,\varphi & \equiv & \mathsf{G}\,\mathsf{F}\,\varphi \\
\mathsf{G}\,\mathsf{F}\,\mathsf{G}\,\varphi & \equiv & \mathsf{F}\,\mathsf{G}\,\varphi
\end{array}
$$

distribution:

$$
\begin{array}{rcl}
\mathsf{X}\,(\varphi\,\mathsf{U}\,\psi) & \equiv & \mathsf{X}\,\varphi\,\mathsf{U}\,\mathsf{X}\,\psi \\
\mathsf{F}\,(\varphi \vee \psi) & \equiv & \mathsf{F}\,\varphi \vee \mathsf{F}\,\psi \\
\mathsf{G}\,(\varphi \wedge \psi) & \equiv & \mathsf{G}\,\varphi \wedge \mathsf{G}\,\psi
\end{array}
$$

but:

$$
\begin{array}{rcl}
\mathsf{F}\,(\varphi \wedge \psi) & \not\equiv & \mathsf{F}\,\varphi \wedge \mathsf{F}\,\psi \\
\mathsf{G}\,(\varphi \vee \psi) & \not\equiv & \mathsf{G}\,\varphi \vee \mathsf{G}\,\psi
\end{array}
$$

Expansion laws

expansion:

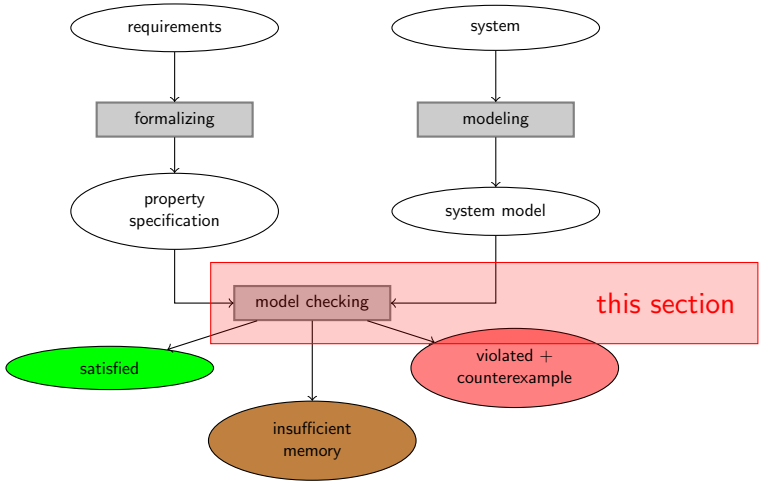
$$
\begin{array}{rcl}
\mathsf{F}\,\varphi & \equiv & \varphi \vee \mathsf{X}\,\mathsf{F}\,\varphi \\
\mathsf{G}\,\varphi & \equiv & \varphi \wedge \mathsf{X}\,\mathsf{G}\,\varphi \\
\varphi\,\mathsf{U}\,\psi & \equiv & \psi \vee (\varphi \wedge \mathsf{X}(\varphi\,\mathsf{U}\,\psi))
\end{array}
$$

$$
\begin{array}{ll}
 & w \models \mathsf{F}\,\varphi \\
\text{iff} & \exists i \geqslant 0 : w[i..] \models \varphi \\
\text{iff} & w[0..] \models \varphi \text{ or } \exists i > 0 : w[i..] \models \varphi \\
\text{iff} & w \models \varphi \text{ or } \exists j \geqslant 0 : w[1+j..] \models \varphi \\
\text{iff} & w \models \varphi \text{ or } \exists j \geqslant 0 : w[1..][j..] \models \varphi \\
\text{iff} & w \models \varphi \text{ or } w[1..] \models \mathsf{F}\,\varphi \\
\text{iff} & w \models \varphi \text{ or } w \models \mathsf{X}\,\mathsf{F}\,\varphi \\
\text{iff} & w \models \varphi \vee \mathsf{X}\,\mathsf{F}\,\varphi
\end{array}
$$

Outline

- LTL - Linear Time Logic
  - Syntax and Semantics
  - Equivalences
- LTL Model Checking
  - Overview
  - Transforming LTL into GNBAs
  - Complexity of LTL Model Checking

Model checking overview



LTL Model Checking using GNBAs

transition system TS and LTL formula φ given

$$
\begin{aligned}
& TS \models \varphi \\
\text{iff} \quad & \mathit{Traces}(TS) \subseteq \mathcal{L}(\varphi) \\
\text{iff} \quad & \mathit{Traces}(TS) \setminus \mathcal{L}(\varphi) = \mathit{Traces}(TS) \cap \mathcal{L}(\neg\varphi) = \varnothing
\end{aligned}
$$

$\Rightarrow$ LTL model checking can be done in four steps

1. calculate GNBA $\mathcal{A}_{TS}$ with $\mathit{Traces}(TS) = \mathcal{L}(\mathcal{A}_{TS})$ (Chapter 2)
2. calculate GNBA $\mathcal{A}_{\neg\varphi}$ with $\mathcal{L}(\neg\varphi) = \mathcal{L}(\mathcal{A}_{\neg\varphi})$ (this section)
3. calculate GNBA $\mathcal{A}$ for intersection of $\mathcal{A}_{TS}$ and $\mathcal{A}_{\neg\varphi}$ (Chapter 2)
4. perform non-emptyness test for $\mathcal{L}(\mathcal{A})$ (Chapter 2)

Fischer Ladner Closure

let φ be an LTL formula over $AP = \{a_1, \ldots, a_n\}$.

Definition
the Fischer Ladner closure $cl(\varphi)$ is the list of sub-formulas of φ (starting from small formulas and ending with φ):

$$a_1, \ldots, a_n, \ldots, \varphi$$

Example

$$cl(\neg b \wedge (\mathrm{X}\, a\, \mathrm{U}\, b)) = a, b, \neg b, \mathrm{X}\, a, \mathrm{X}\, a\, \mathrm{U}\, b, \neg b \wedge (\mathrm{X}\, a\, \mathrm{U}\, b)$$

φ-Expansion

idea:

- expand word by new row for each $\psi \in cl(\varphi)$ which is not an atomic proposition
- write truth-values of ψ in i-th column for subword $w[i..]$

Definition
for $w \in (2^n)^\omega$ and LTL-formula φ with $cl(\varphi) = \varphi_1, \ldots, \varphi_m$ define the φ-expansion as word $v \in (2^m)^\omega$:

$$v[i]^j = 1 \text{ iff } w[i..] \models \varphi_j$$

($v[i]$ is the i-th letter of the infinite word v,
and $v[i]^j$ is the j-th component of the vector $v[i]$)

Example

φ-expansion for $\varphi = \neg b \wedge (\mathsf{X}\, a\, \mathsf{U}\, b)$

explain that first two line define word and are given

a	0 1 1 0 1 0 0 1 1 1 ...
b	0 0 1 1 0 1 0 1 0 1 ...
$\neg b$	1 1 0 0 1 0 1 0 1 0 ...
$\mathsf{X}\, a$	1 1 0 1 0 0 1 1 1 1 ...
$\mathsf{X}\, a\, \mathsf{U}\, b$	1 1 1 1 0 1 1 1 1 1 ...
$\neg b \wedge (\mathsf{X}\, a\, \mathsf{U}\, b)$	1 1 0 0 0 0 1 0 1 0 ...

Idea of LTL to GNBA Translation

- GNBA guesses the φ-expansion of w
- ... and checks that guesses are correct (mostly done locally!)
- ... and demands that value for whole formula is 1 for whole word w

Definition (Consistency Checks)
let $cl(\varphi) = \varphi_1, \ldots, \varphi_m$; a sequence of two successive vectors $(b_1, \ldots, b_m)^T$ $(c_1, \ldots, c_m)^T$ is consistent w.r.t. $cl(\varphi)$ iff whenever

$\varphi_j = \text{true}$	then b_j
$\varphi_j = \neg\varphi_{j_1}$	then $b_j \Leftrightarrow \neg b_{j_1}$
$\varphi_j = \varphi_{j_1} \wedge \varphi_{j_2}$	then $b_j \Leftrightarrow (b_{j_1} \wedge b_{j_2})$
$\varphi_j = \mathsf{X}\,\varphi_{j_1}$	then $b_j \Leftrightarrow c_{j_1}$
$\varphi_j = \varphi_{j_1}\, \mathsf{U}\, \varphi_{j_2}$	then $b_j \Leftrightarrow (b_{j_2} \vee (b_{j_1} \wedge c_j))$

(for last check recall expansion law: $\varphi_{j_1}\, \mathsf{U}\, \varphi_{j_2} \equiv \varphi_{j_2} \vee (\varphi_{j_1} \wedge \mathsf{X}(\varphi_{j_1}\, \mathsf{U}\, \varphi_{j_2}))$)

Consistency Checks and LTL-Models

Lemma
$w \models \varphi$ *iff there exists an expansion* $v \in (2^m)^\omega$ *of* w *such that*

1. $v[i]$ $v[i+1]$ *is consistent for all* i
2. $v[0]^m = 1$
3. *whenever* $\varphi_j = \varphi_{j_1}\, \mathsf{U}\, \varphi_{j_2}$ *and* $v[i]^j = 1$ *then there exists* $i' \geqslant i$ *such that* $v[i']^{j_2} = 1$

Example (3. is necessary NOT IN HANDOUT)

Let $w =$

a	1	1	1	1	1	1	1	...
b	0	0	0	0	0	0	0	...

and $\varphi = a\, \mathsf{U}\, b$.

Choose $v =$

a	1	1	1	1	1	1	1	...
b	0	0	0	0	0	0	0	... .
$a\, \mathsf{U}\, b$	1	1	1	1	1	1	1	...

v satisfies 1 and 2, but $v \not\models a\, \mathsf{U}\, b$.

Translating LTL to GNBA

Definition (GNBA for an LTL formula φ)

let $cl(\varphi) = a_1, \ldots, a_n, \varphi_{n+1}, \ldots, \varphi_m$ where $\varphi_m = \varphi$
define $\mathcal{A}_\varphi = (2^m \uplus \{q_0\}, 2^n, q_0, \delta, F_1, \ldots, F_k)$ where

- $(c_1, \ldots, c_m)^T \in \delta((b_1, \ldots, b_m)^T, (d_1, \ldots, d_n)^T)$ iff
  1. $c_j \Leftrightarrow d_j$ for all $j \leqslant n$ (expansion)
  2. $(b_1, \ldots, b_m)^T$ $(c_1, \ldots, c_m)^T$ is consistent (consistent expansion)
- $(c_1, \ldots, c_m)^T \in \delta(q_0, (d_1, \ldots, d_n)^T)$ iff
  1. $c_j \Leftrightarrow d_j$ for all $j \leqslant n$ (expansion)
  2. c_m (φ is satisfied)
- if $\varphi_j = \varphi_{j_1} \,\mathsf{U}\, \varphi_{j_2}$ is i-th U-subformula in $cl(\varphi)$ then

$$F_i = \{(b_1, \ldots, b_m)^T \mid \neg b_j \vee b_{j_2}\}$$

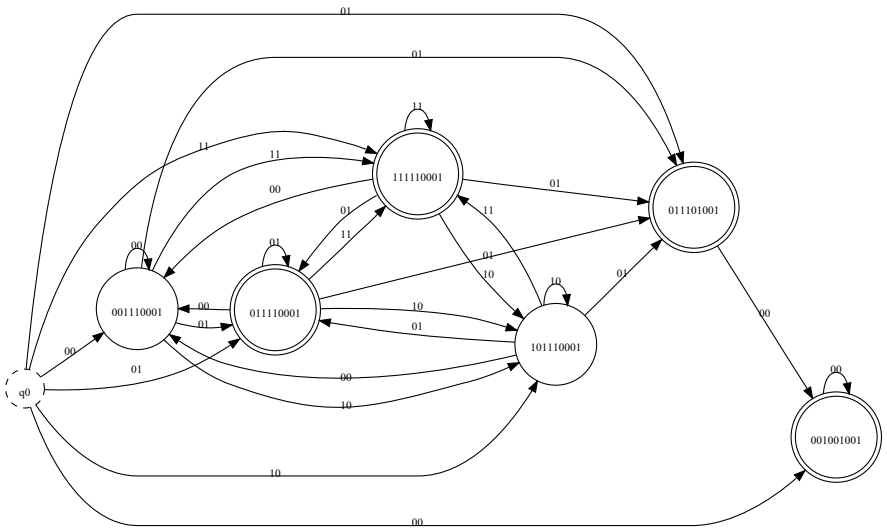
Example

$\varphi = \mathsf{G}(\text{req} \Rightarrow \mathsf{X}\,\mathsf{F}\,\text{resp})$
$\equiv \neg(\text{true}\,\mathsf{U}\,(\text{req} \wedge \neg\mathsf{X}(\text{true}\,\mathsf{U}\,\text{resp}))) = \psi$

$cl(\psi)$	expansion	consistency
1 : *req*	$d_1 \Leftrightarrow c_1$	b_3
2 : *resp*	$d_2 \Leftrightarrow c_2$	$b_4 \Leftrightarrow (b_2 \vee (b_3 \wedge c_4))$
3 : true		$b_5 \Leftrightarrow c_4$
4 : true U *resp*		$b_6 \Leftrightarrow \neg b_5$
5 : X (true U *resp*)		$b_7 \Leftrightarrow (b_1 \wedge b_6)$
6 : $\neg$X (true U *resp*)		$b_8 \Leftrightarrow (b_7 \vee (b_3 \wedge c_8))$
7 : *req* $\wedge \neg$X (true U *resp*)		$b_9 \Leftrightarrow \neg b_8$
8 : true U (*req* $\wedge \neg$X (true U *resp*))		(513*st*/2304*tr*)
9 : $\neg$(true U (*req* $\wedge \neg$X (true U *resp*)))		(7/28*compressed*)

- $(c_1, \ldots, c_9)^T \in \delta((b_1, \ldots, b_9)^T, (d_1, d_2)^T)$ iff expansion and consistent
- $(c_1, \ldots, c_9)^T \in \delta(q_0, (d_1, d_2)^T)$ iff expansion and c_9
- $F_1 = \{(b_1, \ldots, b_9) \mid \neg b_4 \vee b_2\}$ $F_2 = \{(b_1, \ldots, b_9) \mid \neg b_8 \vee b_7\}$

Example (parts of GNBA)



Soundness of Translation

Theorem

for every LTL formula φ

$$\mathcal{L}(\varphi) = \mathcal{L}(\mathcal{A}_\varphi)$$

Proof of Lemma.

by induction on φ using the consistency checks ■

Proof of Theorem.

- construction of $\mathcal{A}_\varphi$ directly corresponds to requirements 1 and 2 in lemma
- remaining difficulty:
  show that visiting F_i infinitely often is the same as requirement 3 in lemma for i-th U-subformula $\varphi_j = \varphi_{j_1} \,\mathsf{U}\, \varphi_{j_2}$ ■

Optimizing the Translation

- observation: many states do not have outgoing transitions
- reason: several inconsistencies due to Boolean conditions
  example: if $\varphi_j = \varphi_{j_1} \wedge \varphi_{j_2}$ then b_j cannot be freely chosen;
  value of b_j is determined by b_{j_1} and b_{j_2}
- idea: take reduced Fischer-Ladner closure $cl'(\varphi)$ which only contains
  - atomic propositions
  - X-formulas
  - U-formulas
  - ... and no other formula

  and then incorporate the Boolean connectives directly into consistency, final states, ...

Towards an Improved Translation

let $cl'(\varphi) = \varphi_1, \ldots, \varphi_m$ over $AP = \{a_1, \ldots, a_n\}$

Definition (Unwinding)

the unwinding of a subformula ψ of φ w.r.t. a vector $B = (b_1, \ldots, b_m)$ is defined as $\mathcal{U}_B(\psi)$ where

$$\begin{aligned}
\mathcal{U}_B(a_i) &= b_i \text{ for all atomic propositions } a_i \\
\mathcal{U}_B(\text{true}) &= \text{true} \\
\mathcal{U}_B(\neg\psi) &= \neg\mathcal{U}_B(\psi) \\
\mathcal{U}_B(\psi_1 \wedge \psi_2) &= \mathcal{U}_B(\psi_1) \wedge \mathcal{U}_B(\psi_2) \\
\mathcal{U}_B(\mathsf{X}\,\psi) &= b_j \text{ where } n < j \leqslant m \text{ is the index s.t. } \varphi_j = \mathsf{X}\,\psi \\
\mathcal{U}_B(\psi\,\mathsf{U}\,\chi) &= b_j \text{ where } n < j \leqslant m \text{ is the index s.t. } \varphi_j = \psi\,\mathsf{U}\,\chi
\end{aligned}$$

Towards an Improved Translation (2)

Definition (Compressed Consistency Checks)

let $cl'(\varphi) = \varphi_1, \ldots, \varphi_m$; a sequence of two successive vectors $B = (b_1, \ldots, b_m)^T$ and $C = (c_1, \ldots, c_m)^T$ is consistent w.r.t. $cl'(\varphi)$ iff whenever

$$\begin{aligned}
&\varphi_j = \mathsf{X}\,\psi \quad \text{then } b_j \Leftrightarrow \mathcal{U}_C(\psi) \\
&\varphi_j = \psi\,\mathsf{U}\,\chi \quad \text{then } b_j \Leftrightarrow (\mathcal{U}_B(\chi) \vee (\mathcal{U}_B(\psi) \wedge c_j))
\end{aligned}$$

Improved Translation

Definition

let $cl'(\varphi) = a_1, \ldots, a_n, \varphi_{n+1}, \ldots, \varphi_m$ define
$\mathcal{A}_\varphi = (2^m \uplus \{q_0\}, 2^n, q_0, \delta, F_1, \ldots, F_k)$ where

- $(c_1, \ldots, c_m)^T \in \delta((b_1, \ldots, b_m)^T, (d_1, \ldots, d_n)^T)$ iff
  1. $c_j \Leftrightarrow d_j$ for all $j \leqslant n$ (expansion)
  2. $(b_1, \ldots, b_m)^T\ (c_1, \ldots, c_m)^T$ is consistent (consistent expansion)
- $C = (c_1, \ldots, c_m)^T \in \delta(q_0, (d_1, \ldots, d_n)^T)$ iff
  1. $c_j \Leftrightarrow d_j$ for all $j \leqslant n$ (expansion)
  2. $\mathcal{U}_C(\varphi)$ (φ is satisfied)
- if $\varphi_j = \psi\,\mathsf{U}\,\chi$ is i-th U-subformula in $cl'(\varphi)$ then

$$F_i = \{B = (b_1, \ldots, b_m)^T \mid \neg b_j \vee \mathcal{U}_B(\chi)\}$$

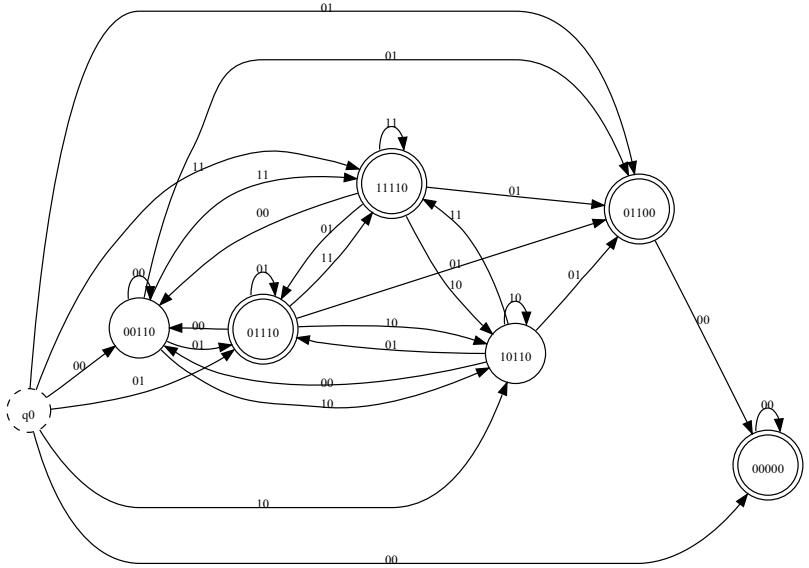
Example NOT IN HANDOUT

$$\varphi = \mathsf{G}\,(\text{req} \Rightarrow \mathsf{X}\,\mathsf{F}\,\text{resp})$$
$$\equiv \neg(\text{true}\,\mathsf{U}\,(\text{req} \wedge \neg\mathsf{X}\,(\text{true}\,\mathsf{U}\,\text{resp}))) = \psi$$

$cl'(\psi)$	expansion	consistency
1 : req	$d_1 \Leftrightarrow c_1$	$b_3 \Leftrightarrow (b_2 \vee (\text{true} \wedge c_3))$
2 : resp	$d_2 \Leftrightarrow c_2$	$b_4 \Leftrightarrow c_3$
3 : true U resp		$b_5 \Leftrightarrow ((b_1 \wedge (\neg b_4)) \vee (\text{true} \wedge c_5))$
4 : X (true U resp)		$(33st/144tr)$
5 : true U (req $\wedge \neg$X (true U resp))		$(7/28 compressed)$

- $(c_1, \ldots, c_5)^T \in \delta((b_1, \ldots, b_5)^T, (d_1, d_2)^T)$ iff expansion and consistent
- $(c_1, \ldots, c_5)^T \in \delta(q_0, (d_1, d_2)^T)$ iff expansion and $\neg c_5$
- $F_1 = \{(b_1, \ldots, b_5) \mid \neg b_3 \vee b_2\}$ $\quad F_2 = \{(b_1, \ldots, b_5) \mid \neg b_5 \vee (b_1 \wedge \neg b_4)\}$

Example GNBA



Soundness of the Improved Transformation

Theorem

both transformations yield essentially the same automata; the only difference is that

- *whenever $\varphi_i = \varphi_{i_1} \wedge \varphi_{i_2}$ then the i-th component is missing in the improved transformation; the state without the i-th component of the improved transformation is corresponding to the state $(\ldots, b_{i_1}, \ldots, b_{i_2}, \ldots, b_i, \ldots)$ where $b_i = b_{i_1} \wedge b_{i_2}$; moreover, all other states in the non-improved translation where $b_i \neq b_{i_1} \wedge b_{i_2}$ have no outgoing states*
- *a similar property is true for negations*

$\Rightarrow$ *soundness of the non-improved translation implies soundness of the improved one*

Complexity of LTL Model Checking

LTL model checking: given TS and φ check

$$\mathcal{L}(\mathcal{A}_3) = \varnothing$$

where

$$\mathcal{A}_1 = \mathcal{A}_{TS}$$
$$\mathcal{A}_2 = \mathcal{A}_{\neg\varphi}$$
$$\mathcal{A}_3 = \mathcal{A}_{\mathcal{A}_1 \cap \mathcal{A}_2}$$

number of states:

$\mathcal{A}_1 : |TS| + 1$

$\mathcal{A}_2 : 2^{|AP|+|\varphi|} + 1$ where $|\varphi|$ is number of temporal operators in φ

$\mathcal{A}_3 : (|TS| + 1) \cdot (2^{|AP|+|\varphi|} + 1)$

$\Rightarrow$ total complexity of $\mathcal{O}(|TS| \cdot 2^{|AP|+|\varphi|})$ linear in TS, exp. in φ

Lower bound

Theorem
there exists a family of LTL formulas φ_n over $AP = \{a\}$ with $|\varphi_n| = \mathcal{O}(poly(n))$ such that every NBA $\mathcal{A}_n$ with $\mathcal{L}(\mathcal{A}_n) = \mathcal{L}(\varphi_n)$ has at least 2^n states

Proof (1)

let $AP = \{a\}$ and:

$$\mathcal{L}_n \;=\; \{\, b_1 \ldots b_n\, b_1 \ldots b_n\, w \mid b_i \in \{0,1\}, w \in \{0,1\}^\omega \,\}, \qquad \text{for } n \geqslant 0$$

it follows $\mathcal{L}_n = \mathcal{L}(\varphi_n)$ where $\varphi_n = \bigwedge_{0 \leqslant i < n} (\mathsf{X}^i\, a \Leftrightarrow \mathsf{X}^{n+i}\, a)$

φ_n is an LTL formula of polynomial length: $|\varphi_n| \in \mathcal{O}(n^2)$

however, any NBA $\mathcal{A}$ with $\mathcal{L}_\omega(\mathcal{A}) = \mathcal{L}_n$ has at least 2^n states

Proof (2)

claim: any NBA $\mathcal{A}$ for $\bigwedge_{0 \leqslant i < n} (\mathsf{X}^i\, a \Leftrightarrow \mathsf{X}^{n+i}\, a)$ has at least 2^n states

words of the form $b_1 \ldots b_n\, b_1 \ldots b_n\, 0\,0\,0 \ldots$ are accepted by $\mathcal{A}$.
thus, for every word $w = b_1 \ldots b_n$ of length n there is a state $q(w)$ in $\mathcal{A}$ which can be reached from the initial state by consuming $b_1 \ldots b_n$.
Moreover, from $q(w)$, it is possible to visit an accept state infinitely often by reading the word $w\,0\,0\,0 \ldots$.
if for every two different words w and w' of length n, the states $q(w)$ and $q(w')$ are different, then there are at least 2^n states. Otherwise, there are two words w and w' where $w \neq w'$, but $q(w) = q(w')$. But then $\mathcal{A}$ cannot accept the language since $w\,w'\,0\,0\,0 \ldots$ does not satisfy the formula, but has an accepting run (go from the initial state to $q(w) = q(w')$ by reading w, and then continue from $q(w')$ with the accepting run for $w'\,0\,0\,0 \ldots$

Is Bad Complexity a Result of Automata Approach? No!

Theorem (Sistla, Clarke)
the LTL model checking problem $TS \models \varphi$ is PSPACE-hard

Theorem
the LTL model checking problem $TS \models \varphi$ is PSPACE-complete

Summary

- LTL is a logic for specifying the allowed traces of a system; it extends propositional logic by temporal operators like X and U
- LTL model checking can be done by constructing a GNBA and then checking whether this GNBA accepts at least one word, the counterexample
- LTL model checking is PSPACE-complete