Introduction to Model Checking

René Thiemann

Institute of Computer Science
University of Innsbruck

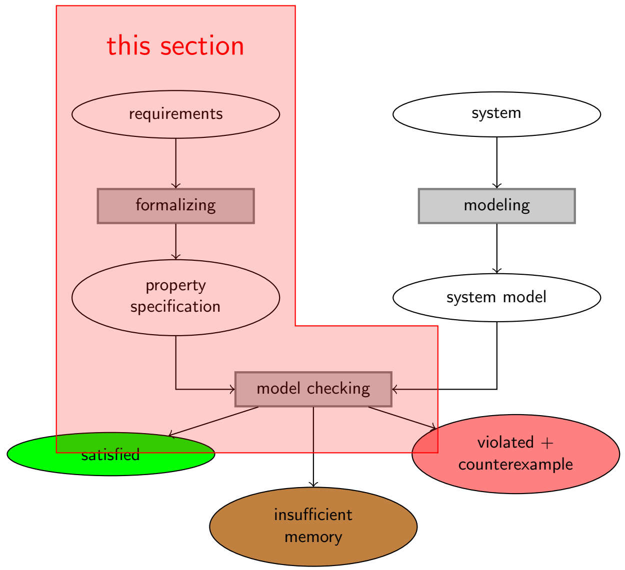
WS 2009/2010

Outline

Outline

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

Model checking overview



Requirements ≠ Specification

requirements

- high-level description (consider scheduler for exclusive access)
  - (the scheduler should be correct)
  - no two clients get access at the same time
  - the scheduler should be fair
  - there is no deadlock
- what we observe from system: $\mathit{Traces}(TS) \subseteq (2^{AP})^\omega$

$\Rightarrow$ how to answer question "does system satisfy requirements"? problem: to imprecise

$\Rightarrow$ we need requirements in a precise, i.e., mathematical specification

Linear Time Properties

one main idea to specify requirements: describe allowed traces

- specification is set $\mathcal{S} \subseteq (2^{AP})^\omega$ (linear time property)
- system TS satisfies $\mathcal{S}$ iff every trace of TS is allowed w.r.t. $\mathcal{S}$:

$$\mathit{Traces}(TS) \subseteq \mathcal{S}$$

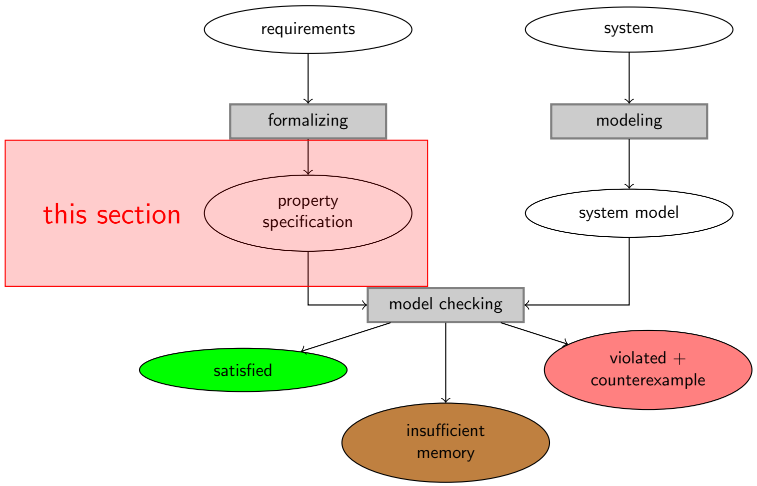
- model checking of linear time properties: given $\mathit{Traces}(TS)$ and $\mathcal{S}$, answer $\mathit{Traces}(TS) \subseteq \mathcal{S}$

$\Rightarrow$ precise formulation, no ambiguity

- upcoming problems
  - how to specify sets $\mathcal{S}$ conveniently ...
  - ... such that $\mathit{Traces}(TS) \subseteq \mathcal{S}$ can be decided

Outline

Model checking overview



Syntax of Linear Temporal Logic

modal logic over infinite sequences [Pnueli 1977]

- propositional logic
  - true
  - a, paid, sprite, ... $\in AP$ — 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

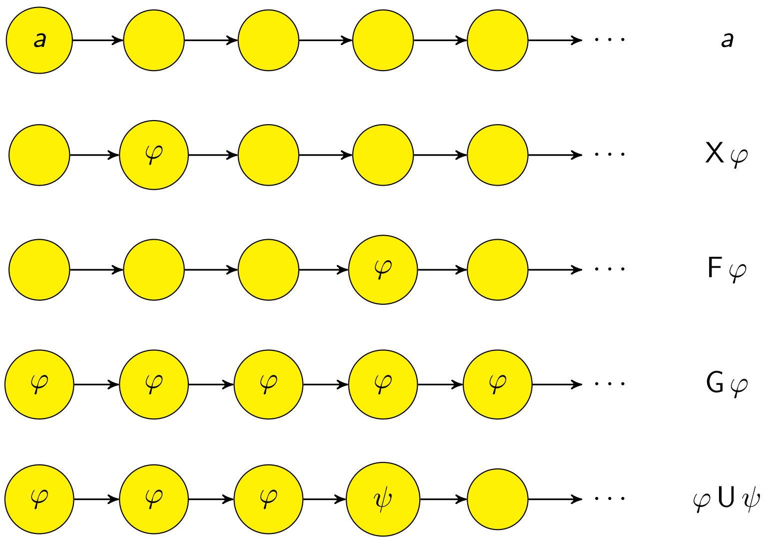
Derived operators

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

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



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 and at sometime orange appears; moreover, the red is later than the orange

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

most of these requirements do not correspond to the specifications

Practical properties in LTL

- Reachability
  - reachability $\mathsf{F}\,\psi$
  - conditional reachability $\varphi\,\mathsf{U}\,\psi$
  - reachability from any state not expressible
- Safety $\mathsf{G}\,\neg\varphi$
- Liveness $\mathsf{G}\,(\varphi \Rightarrow \mathsf{F}\,\psi)$ and others
- Fairness $\mathsf{G}\,\mathsf{F}\,\varphi$ and others

Semantics over words

the language induced by LTL formula φ over $AP = \{a_1, \dots, a_n\}$ 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 $w = A_0A_1A_2\dots$ and $w[i..] = A_iA_{i+1}A_{i+2}\dots$ is the suffix of w from index i on)

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

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

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

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!

example:

- let w_1 and w_2 be two different traces of TS such that $w_1 \models \varphi$ and $w_2 \not\models \varphi$
- due to w_2 we know $TS \not\models \varphi$
- due to w_1 we know $w_1 \not\models \neg\,\varphi$ and hence $TS \not\models \neg\,\varphi$

$\Rightarrow$ $TS \not\models \varphi$ and $TS \not\models \neg\,\varphi$

Example

Equivalence of LTL formulas, Deriving F and G

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

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

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

Often used constructs

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

Duality and idempotence laws

duality:

$$\neg\,\mathsf{G}\,\varphi \equiv \mathsf{F}\,\neg\,\varphi$$
$$\neg\,\mathsf{F}\,\varphi \equiv \mathsf{G}\,\neg\,\varphi$$
$$\neg\,\mathsf{X}\,\varphi \equiv \mathsf{X}\,\neg\,\varphi$$

idempotency:

$$\mathsf{G}\,\mathsf{G}\,\varphi \equiv \mathsf{G}\,\varphi$$
$$\mathsf{F}\,\mathsf{F}\,\varphi \equiv \mathsf{F}\,\varphi$$
$$\varphi\,\mathsf{U}\,(\varphi\,\mathsf{U}\,\psi) \equiv \varphi\,\mathsf{U}\,\psi$$
$$(\varphi\,\mathsf{U}\,\psi)\,\mathsf{U}\,\psi \equiv \varphi\,\mathsf{U}\,\psi$$

Absorption and distributive laws

absorption:
$$\mathsf{F}\,\mathsf{G}\,\mathsf{F}\,\varphi \equiv \mathsf{G}\,\mathsf{F}\,\varphi$$
$$\mathsf{G}\,\mathsf{F}\,\mathsf{G}\,\varphi \equiv \mathsf{F}\,\mathsf{G}\,\varphi$$

distribution:
$$\mathsf{X}\,(\varphi\,\mathsf{U}\,\psi) \equiv \mathsf{X}\,\varphi\,\mathsf{U}\,\mathsf{X}\,\psi$$
$$\mathsf{F}\,(\varphi \vee \psi) \equiv \mathsf{F}\,\varphi \vee \mathsf{F}\,\psi$$
$$\mathsf{G}\,(\varphi \wedge \psi) \equiv \mathsf{G}\,\varphi \wedge \mathsf{G}\,\psi$$

but:
$$\mathsf{F}\,(\varphi \wedge \psi) \not\equiv \mathsf{F}\,\varphi \wedge \mathsf{F}\,\psi$$
$$\mathsf{G}\,(\varphi \vee \psi) \not\equiv \mathsf{G}\,\varphi \vee \mathsf{G}\,\psi$$

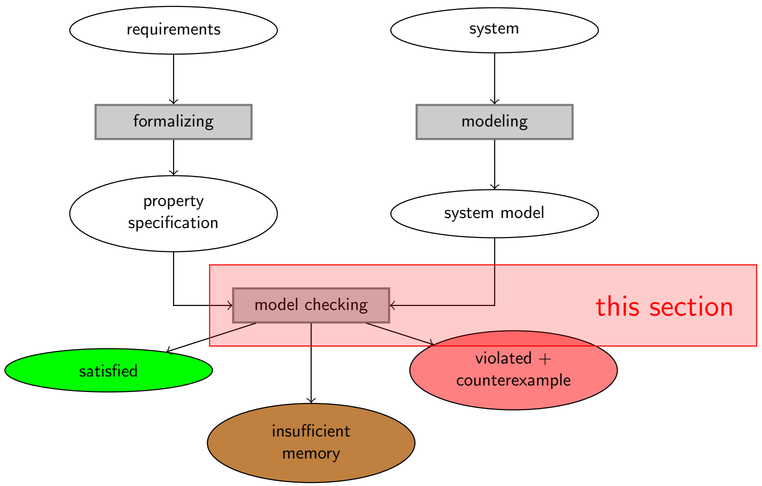
Example

Expansion laws

expansion:
$$\mathsf{F}\,\varphi \equiv \varphi \vee \mathsf{X}\,\mathsf{F}\,\varphi$$
$$\mathsf{G}\,\varphi \equiv \varphi \wedge \mathsf{X}\,\mathsf{G}\,\varphi$$
$$\varphi\,\mathsf{U}\,\psi \equiv \psi \vee (\varphi \wedge \mathsf{X}(\varphi\,\mathsf{U}\,\psi))$$

Outline

Model checking overview



The requirements of model checking

essentially we need a mechanism to represent $\mathcal{L}(\varphi)$ for LTL formula φ

- possible classes: finite, regular, context-free, context-sensitive, ...
- model checking requires checking $Traces(TS) \subseteq \mathcal{L}(\varphi)$ or equivalently: $Traces(TS) \cap \mathcal{L}(\neg\varphi) = \varnothing$

$\Rightarrow$ requirements on class of language
  - closure under intersection
  - emptyness decidable
  - expressive enough to represent $Traces(TS)$ and $\mathcal{L}(\varphi)$
- use regular languages, they are closed under all boolean operations
- possible representations of regular languages
  - regular expressions
  - non-recursive grammars
  - finite automata

Are GNBA's expressive enough to express LTL?

Safety properties: (refutation by a finite prefix of an ω-word)

1. always at most one traffic light is showing green
   $\mathsf{G}\,\neg(\mathsf{green}_1 \wedge \mathsf{green}_2)$
2. green cannot be directly followed by red
   $\neg\mathsf{F}\,(\mathsf{green} \wedge \mathsf{X}\,\mathsf{red})$

Liveness properties: (refutation only by whole ω-word)

3. we will see green infinitely often
   $\mathsf{G}\,\mathsf{F}\,\mathsf{green}$
4. whenever we select sprite then later on we will get a sprite
   $\mathsf{G}\,(\mathsf{sel\_sprite} \Rightarrow \mathsf{X}\,\mathsf{F}\,\mathsf{get\_sprite})$

$\Rightarrow$ many interesting properties can be expressed by GNBAs (and indeed every LTL formula can be translated into equivalent GNBA)

Properties as GNBAs

Model Checking with 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 $AP = \{a_1, \dots, a_n\}$.

Definition

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

$$a_1, \dots, a_n, \dots, \varphi$$

Example

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 $cl(\varphi) = \varphi_1, \dots, \varphi_m$ 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 $\varphi = \neg b \wedge (\mathsf{X}\, a \,\mathsf{U}\, b)$

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 $v[i']^{j_2} = 1$*

Translating LTL to GNBA

Definition (GNBA for an LTL formula φ)

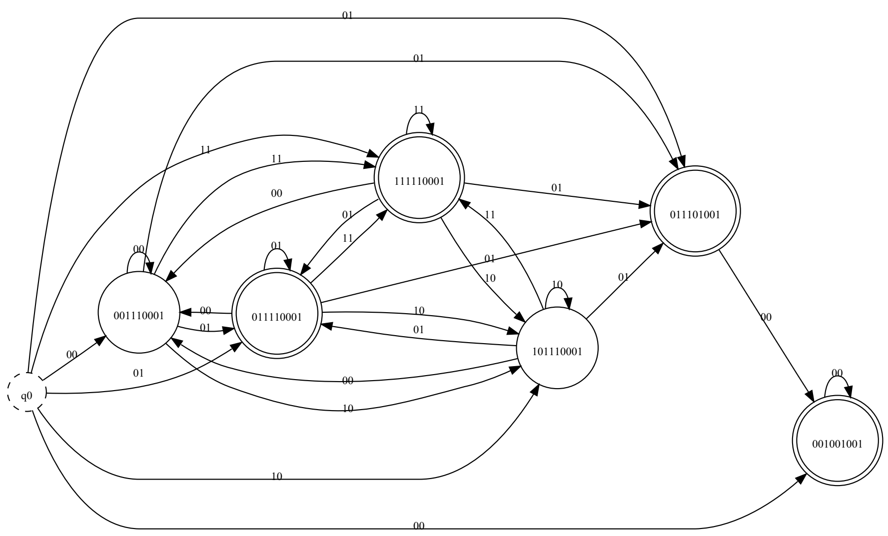
let $cl(\varphi) = a_1, \ldots, a_n, \varphi_{n+1}, \ldots, \varphi_m$ where $\varphi_m = \varphi$
define $\mathcal{A}_\varphi = (2^m \uplus \{q_0\}, 2^n, q_0, \delta, F_1, \ldots, F_k)$ where

- $(c_1, \ldots, c_m)^T \in \delta((b_1, \ldots, b_m)^T, (d_1, \ldots, d_n)^T)$ 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

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

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

Towards an Improved Translation (2)

Definition (Compressed Consistency Checks)

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

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

Improved Translation

Definition

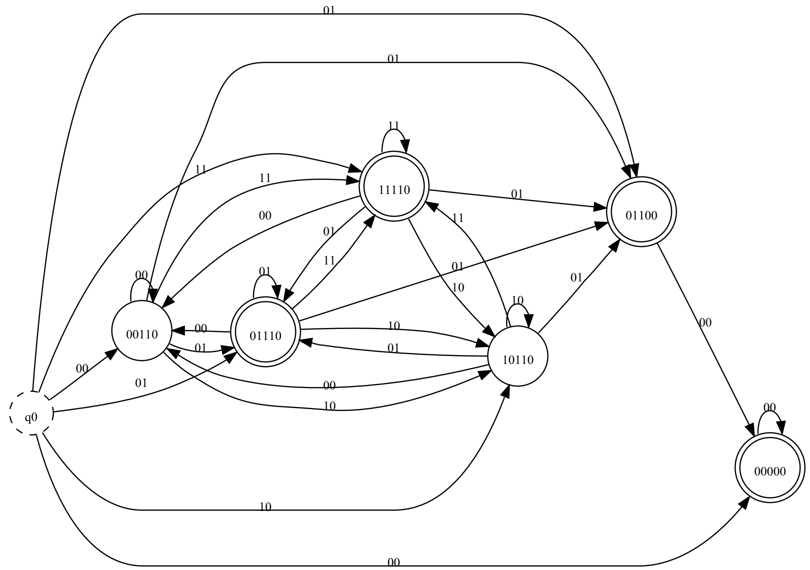
let $cl'(\varphi) = a_1, \dots, a_n, \varphi_{n+1}, \dots, \varphi_m$ define $\mathcal{A}_\varphi = (2^m \uplus \{q_0\}, 2^n, q_0, \delta, F_1, \dots, F_k)$ where

- $(c_1, \dots, c_m)^T \in \delta((b_1, \dots, b_m)^T, (d_1, \dots, d_n)^T)$ iff
  1. $c_j \Leftrightarrow d_j$ for all $j \leqslant n$ (expansion)
  2. $(b_1, \dots, b_m)^T\ (c_1, \dots, c_m)^T$ is consistent (consistent expansion)
- $C = (c_1, \dots, c_m)^T \in \delta(q_0, (d_1, \dots, d_n)^T)$ 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

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

Example

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* $(\dots, b_{i_1}, \dots, b_{i_2}, \dots, b_i, \dots)$ *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 $\mathcal{O}(|TS| \cdot 2^{|AP|+|\varphi|})$

Lower bound

Theorem

there exists a family of LTL formulas φ_n *over* $AP = \{\mathrm{a}\}$ *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)

Proof (2)

Is Bad Complexity a Result of Automata Approach? No!

Theorem

the inverted LTL model checking problem

$$TS \not\models \varphi$$

is NP-hard

Corollary

the LTL model checking problem $TS \models \varphi$ *is coNP-hard*

⇒ assuming $P \neq NP$, LTL model checking cannot be done in polynomial time

Proving NP-hardness

to prove NP-hardness of problem p one can use a reduction:

1. find another problem q which is known to be NP-hard
2. reduce q to p in polynomial time, i.e.,
   find a mapping $\mu : q \to p$ such that
   - q has a positive answer iff $\mu(q)$ has a positive answer
   - μ can be computed in polynomial time

NP-Hardness of "$TS \not\models \varphi$"

Example

Complexity is Even Worse ...

Theorem (Sistla, Clarke)

the LTL model checking problem $TS \models \varphi$ *is PSPACE-hard*

... But There is a Limit

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