Computer Aided Verification
CAV a.a. 2014/15

Giorgio Delzanno

DIBRIS, Università di Genova

Temporal Logic

Temporal Logic: A Class of Modal Logics

- Modal Logic: alternative notions of truth like *is it possible/necessary that φ is true?*
- Temporal logic is a special type of modal logic in which the truth of a formula depends on the time in which it is evaluated
- Typical temporal operators are
  - Eventually Φ: in some future instant Φ is true
  - Always Φ: in all future instants Φ is true

Several Types of Temporal Logics

- Linear Temporal Logic (LTL) is linear in the future; properties are defined on a path

Several Types of Temporal Logics

- Linear Temporal Logic (LTL) is linear in the future; properties are defined on a path
- Computational Tree Logic (CTL) is branching in the future; properties are defined on a tree

Several Types of Temporal Logics

- Linear Temporal Logic (LTL) is linear in the future; properties are defined on a path
- Computational Tree Logic (CTL) is branching in the future; properties are defined on a tree
- LTL and CTL are incomparable logics: There exist formulas in one logic that are not expressible in the other

Several Types of Temporal Logics

- Linear Temporal Logic (LTL) is linear in the future; properties are defined on a path
- Computational Tree Logic (CTL) is branching in the future; properties are defined on a tree
- LTL and CTL are incomparable logics: There exist formulas in one logic that are not expressible in the other

Several Types of Temporal Logics

- Linear Temporal Logic (LTL) is linear in the future; properties are defined on a path
- Computational Tree Logic (CTL) is branching in the future; properties are defined on a tree
- LTL and CTL are incomparable logics: There exist formulas in one logic that are not expressible in the other
- LTL and CTL are submsumed by CTL*, which in turn, is subsumed by the μ-calculus (a fixpoint logic)

Local Model Checking Problem

- Fixed a (Kripke) model M (a transition system), an initial state s_0, and a temporal property φ

$$M, s_0 \models \varphi?$$

where $\models$ = satisfiability relation

Global Model Checking Problem

- Fixed a (Kripke) model M and a temporal property φ, compute all states s such that $M, s \models \varphi$.
- (Global solves Local)

Linear Temporal Logic

PLTL Syntax

- Atomic Proposition: predicate symbols $p, q, r, \ldots$

PLTL Syntax

- Atomic Proposition: predicate symbols $p, q, r, \ldots$
- Classical connectives:

$$\neg\psi \quad \varphi \wedge \psi \quad \varphi \vee \psi \quad \varphi \supset \psi$$

Temporal Operators

A formula withot modalities at the top level is evaluated in the current instant

Temporal operators are interpreted over an infinite path in which states are labeled by sets of propositions:

- $\mathbf{X}\varphi$: φ is true in the next time instant

Temporal Operators

A formula withot modalities at the top level is evaluated in the current instant

Temporal operators are interpreted over an infinite path in which states are labeled by sets of propositions:

- $\mathbf{X}\varphi$: φ is true in the next time instant
- $\mathbf{F}\varphi$: in the future/eventually φ

Temporal Operators

A formula withot modalities at the top level is evaluated in the current instant

Temporal operators are interpreted over an infinite path in which states are labeled by sets of propositions:

- $\mathbf{X}\varphi$: φ is true in the next time instant
- $\mathbf{F}\varphi$: in the future/eventually φ
- $\mathbf{G}\varphi$: globally/always φ

Temporal Operators

A formula withot modalities at the top level is evaluated in the current instant

Temporal operators are interpreted over an infinite path in which states are labeled by sets of propositions:

- $\mathbf{X}\varphi$: φ is true in the next time instant
- $\mathbf{F}\varphi$: in the future/eventually φ
- $\mathbf{G}\varphi$: globally/always φ
- $\varphi\mathbf{U}\psi$: φ until ψ

Logical equivalences

- The set $\{ \neg, \vee, \mathbf{X}, \mathbf{U} \}$ is sufficiently complete to define LTL formulas

Logical equivalences

- The set $\{ \neg, \vee, \mathbf{X}, \mathbf{U} \}$ is sufficiently complete to define LTL formulas
- Indeed,

$$\begin{aligned} \mathbf{F}\varphi &\equiv \mathit{true}\ \mathbf{U}\ \varphi \\ \mathbf{G}\varphi &= \neg\mathbf{F}\neg\varphi \end{aligned}$$

Traffic Light

- once green, next state is not red

$$\mathbf{G}(\mathit{green} \supset \neg \mathbf{X}\,\mathit{red})$$

Traffic Light

- once green, next state is not red

$$\mathbf{G}(\mathit{green} \supset \neg \mathbf{X}\,\mathit{red})$$

- eventually, it becomes green

$$\mathbf{F}\ \mathit{green}$$

Traffic Light

- once green, next state is not red

$$\mathbf{G}(\mathit{green} \supset \neg\mathbf{X}\mathit{red})$$

- eventually, it becomes green

$$\mathbf{F}\ \mathit{green}$$

- once green, becomes red after being yellow for some time

$$\mathbf{G}(\mathit{green} \supset ((\mathit{green}\ \mathbf{U}\ \mathit{yellow})\ \mathbf{U}\ \mathit{red}))$$

Semantics

- An LTL model is an infinite path $\sigma = \langle S, R, L \rangle$ defined as
  - S is a denumerable, non empty set of states

Semantics

- An LTL model is an infinite path $\sigma = \langle S, R, L \rangle$ defined as
  - S is a denumerable, non empty set of states
  - $R : S \rightarrow S$ assigns a unique successor $R(s)$ to each state s ($s\ R(s)\ R(R(s))\ \ldots$ is an infinite sequence of states)

Semantics

- An LTL model is an infinite path $\sigma = \langle S, R, L \rangle$ defined as
  - S is a denumerable, non empty set of states
  - $R : S \rightarrow S$ assigns a unique successor $R(s)$ to each state s ($s\ R(s)\ R(R(s))\ \dots$ is an infinite sequence of states)
  - $L : S \rightarrow 2^{AP}$ assigns to $s \in S$ the set of propositions formulas that are true in s

Semantics

- An LTL model is an infinite path $\sigma = \langle S, R, L \rangle$ defined as
  - S is a denumerable, non empty set of states
  - $R : S \rightarrow S$ assigns a unique successor $R(s)$ to each state s ($s\ R(s)\ R(R(s))\ \ldots$ is an infinite sequence of states)
  - $L : S \rightarrow 2^{AP}$ assigns to $s \in S$ the set of propositions formulas that are true in s
  - We can represent it as a finite graph (labels over 2^{AP}!)

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as

  - $\sigma, s \models p$ if $p \in L(s)$

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as
  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$

Satisfiability

- For $\sigma = \langle S, R, L\rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as
  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$
  - $\sigma, s \models \varphi \vee \psi$ if $\sigma, s \models \varphi$ or $s \models \psi$

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as

  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$
  - $\sigma, s \models \varphi \vee \psi$ if $\sigma, s \models \varphi$ or $s \models \psi$
  - $\ldots$

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as
  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$
  - $\sigma, s \models \varphi \vee \psi$ if $\sigma, s \models \varphi$ or $s \models \psi$
  - $\ldots$
  - $\sigma, s \models \mathbf{X}\varphi$ if $\sigma, R(s) \models \varphi$

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as

  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$
  - $\sigma, s \models \varphi \vee \psi$ if $\sigma, s \models \varphi$ or $s \models \psi$
  - $\ldots$
  - $\sigma, s \models \mathbf{X}\varphi$ if $\sigma, R(s) \models \varphi$

  - $\sigma, s \models \varphi_1 \; \mathbf{U} \; \varphi_2$ if
    $\exists j \geq 0.\ \sigma, R^j(s) \models \varphi_2,\ (\forall 0 \leq k < j.\ \sigma, R^k(s) \models \varphi_1)$

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as
  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$
  - $\sigma, s \models \varphi \vee \psi$ if $\sigma, s \models \varphi$ or $s \models \psi$
  - $\ldots$
  - $\sigma, s \models \mathbf{X}\varphi$ if $\sigma, R(s) \models \varphi$
  - $\sigma, s \models \varphi_1 \ \mathbf{U} \ \varphi_2$ if
    $\exists j \geq 0.\ \sigma, R^j(s) \models \varphi_2,\ (\forall 0 \leq k < j.\ \sigma, R^k(s) \models \varphi_1)$
  - $\sigma, s \models \mathbf{F}\varphi$ if $\exists j.\ \sigma, R^j(s) \models \varphi$

Satisfiability

- For $\sigma = \langle S, R, L \rangle$, let $R^j(s) = \underbrace{R(\ldots R(s)\ldots)}_{j}$

  The relation $\sigma, s \models \varphi$ (σ satisfies φ in s) is defined as
  - $\sigma, s \models p$ if $p \in L(s)$
  - $\sigma, s \models \neg\phi$ if $\sigma, s \not\models \phi$
  - $\sigma, s \models \varphi \vee \psi$ if $\sigma, s \models \varphi$ or $s \models \psi$
  - $\ldots$
  - $\sigma, s \models \mathbf{X}\varphi$ if $\sigma, R(s) \models \varphi$
  - $\sigma, s \models \varphi_1 \ \mathbf{U} \ \varphi_2$ if
    $\exists j \geq 0.\ \sigma, R^j(s) \models \varphi_2,\ (\forall 0 \leq k < j.\ \sigma, R^k(s) \models \varphi_1)$
  - $\sigma, s \models \mathbf{F}\varphi$ if $\exists j.\ \sigma, R^j(s) \models \varphi$
  - $\sigma, s \models \mathbf{G}\varphi$ if $\forall j.\ \sigma, R^j(s) \models \varphi$

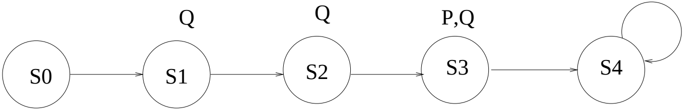
Example 1

- If $p\mathbf{U}q$ holds in s, then $\mathbf{F}q$ holds in s,
- The weak until operator:

$$p\ \mathbf{W}\ q \ \equiv\ \mathbf{G}p\ \vee\ (p\ \mathbf{U}\ q)$$

Example 2

Let M be the following model

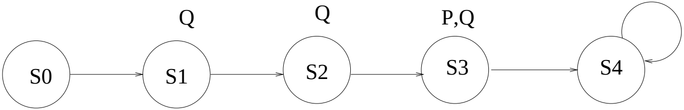


$L(s_0) = \emptyset$, $L(s_1) = \{q\}$, $L(s_2) = \{q\}$, $L(s_3) = \{p, q\}$, $L(s_4) = \emptyset$ then

- $\mathbf{X}p$ is true in s_2

Example 2

Let M be the following model

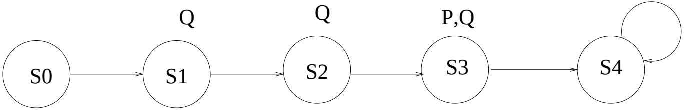


$L(s_0) = \emptyset$, $L(s_1) = \{q\}$, $L(s_2) = \{q\}$, $L(s_3) = \{p, q\}$, $L(s_4) = \emptyset$
then

- $\mathbf{X}p$ is true in s_2
- $\mathbf{F}p$ is true in s_0, s_1, s_2, s_3

Example 2

Let M be the following model

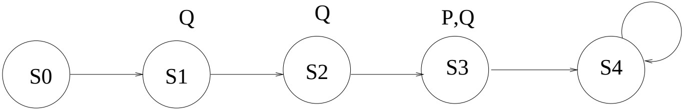


$L(s_0) = \emptyset$, $L(s_1) = \{q\}$, $L(s_2) = \{q\}$, $L(s_3) = \{p, q\}$, $L(s_4) = \emptyset$
then

- $\mathbf{X}p$ is true in s_2
- $\mathbf{F}p$ is true in s_0, s_1, s_2, s_3
- $\mathbf{G}p$ is never true

Example 2

Let M be the following model

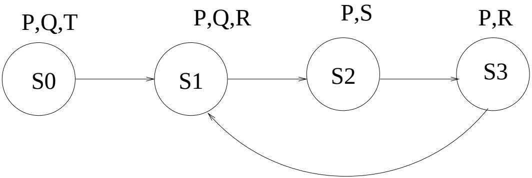


$L(s_0) = \emptyset$, $L(s_1) = \{q\}$, $L(s_2) = \{q\}$, $L(s_3) = \{p, q\}$, $L(s_4) = \emptyset$
then

- $\mathbf{X}p$ is true in s_2
- $\mathbf{F}p$ is true in s_0, s_1, s_2, s_3
- $\mathbf{G}p$ is never true
- $q\ \mathbf{U}\ p$ is true in s_1, s_2, s_3

Example II

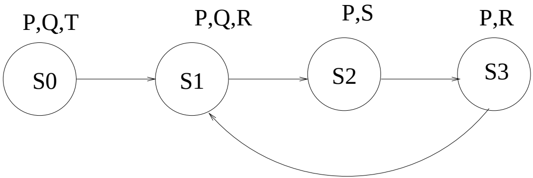
- Let M be the following model



- then
  - $\mathbf{F}t$ is true in s_0

Example II

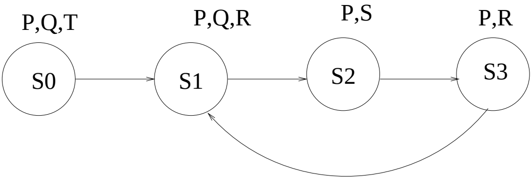
- Let M be the following model



- then
  - $\mathbf{F}t$ is true in s_0
  - $\mathbf{G}p$ is true in all states

Example II

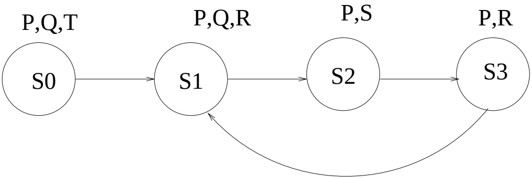
- Let M be the following model



- then
  - $\mathbf{F}t$ is true in s_0
  - $\mathbf{G}p$ is true in all states
  - $\mathbf{G}\ \mathbf{F}s$ is true in all states

Example II

- Let M be the following model



- then
  - $\mathbf{F}t$ is true in s_0
  - $\mathbf{G}p$ is true in all states
  - $\mathbf{G}\ \mathbf{F}s$ is true in all states
  - $\mathbf{X}(r \supset (q\ \mathbf{U}\ s))$ is true in s_0, s_1, s_3

Interesting LTL formulas

- $\mathbf{G}p$: always p (safety property)

Interesting LTL formulas

- $\mathbf{G}p$: always p (safety property)
- $p \supset \mathbf{F}q$: if p holds initially, eventually q holds (reachability)

Interesting LTL formulas

- $\mathbf{G}p$: always p (safety property)
- $p \supset \mathbf{F}q$: if p holds initially, eventually q holds (reachability)
- $\mathbf{G}\ \mathbf{F}p$: p is true infinitely often (fairness)

Interesting LTL formulas

- $\mathbf{G}p$: always p (safety property)
- $p \supset \mathbf{F}q$: if p holds initially, eventually q holds (reachability)
- $\mathbf{G}\ \mathbf{F}p$: p is true infinitely often (fairness)
- $\mathbf{F}\ \mathbf{G}p$: when p becomes true, it remains true forever (eventually permanently)

Interesting LTL formulas

- $\mathbf{G}p$: always p (safety property)
- $p \supset \mathbf{F}q$: if p holds initially, eventually q holds (reachability)
- $\mathbf{G}\ \mathbf{F}p$: p is true infinitely often (fairness)
- $\mathbf{F}\ \mathbf{G}p$: when p becomes true, it remains true forever (eventually permanently)
- $\mathbf{G}(p \supset \mathbf{F}q)$: globally, if p holds then eventually q holds (responsiveness)

Interesting LTL formulas

- $\mathbf{GF}p \; \not\equiv \; \mathbf{F}p$

Interesting LTL formulas

- $\mathbf{GF}p \not\equiv \mathbf{F}p$
- $\mathbf{FG}p \not\equiv \mathbf{G}p$

Axioms for LTL: Duality

$$\neg \mathbf{G}\varphi \equiv \mathbf{F}\neg\varphi$$

$$\neg \mathbf{F}\varphi \equiv \mathbf{G}\neg\varphi$$

$$\neg \mathbf{X}\varphi \equiv \mathbf{X}\neg\varphi$$

Axioms for LTL: Expansion

$$\varphi \,\mathbf{U}\, \psi \;\equiv\; \psi \;\vee\; [\varphi \;\wedge\; \mathbf{X}(\varphi \,\mathbf{U}\, \psi)]$$

$$\mathbf{F}\varphi \;\equiv\; \varphi \;\vee\; \mathbf{X}\;\mathbf{F}\varphi$$

$$\mathbf{G}\varphi \;\equiv\; \varphi \;\wedge\; \mathbf{X}\;\mathbf{G}\varphi$$

Axioms for LTL: Idempotence

$$\mathbf{G}\ \mathbf{G}\varphi \equiv \mathbf{G}\varphi$$

$$\mathbf{F}\ \mathbf{F}\varphi \equiv \mathbf{F}\varphi$$

$$\varphi\ \mathbf{U}\ (\varphi \mathbf{U} \psi) \equiv \varphi\ \mathbf{U}\ \psi$$

$$(\varphi\ \mathbf{U}\ \psi)\mathbf{U}\psi \equiv \varphi\ \mathbf{U}\ \psi$$

Axioms for LTL: Absorbtion

$$\mathbf{F}\ \mathbf{G}\ \mathbf{F}\varphi \equiv \mathbf{G}\ \mathbf{F}\varphi$$

$$\mathbf{G}\ \mathbf{F}\ \mathbf{G}\varphi \equiv \mathbf{F}\ \mathbf{G}\varphi$$

Axioms for LTL: Commutation

$$\mathbf{X}(\varphi \ \mathbf{U} \ \psi) \quad \equiv \quad (\mathbf{X}\varphi) \ \mathbf{U} \ (\mathbf{X} \ \psi)$$

A Deduction System for LTL

Axioms:

$$
\begin{array}{l}
\textit{All tautologies} \\
\mathbf{X}(\neg\varphi) \equiv \neg\mathbf{X}\varphi \\
\mathbf{X}(\varphi \supset \psi) \supset \mathbf{X}\varphi \supset \mathbf{X}\psi \\
\mathbf{G}\varphi \supset (\varphi \wedge \mathbf{X}\mathbf{G}\varphi)
\end{array}
$$

Rules:

Modus ponens	$\varphi, \varphi \supset \psi \vdash \psi$
Next	$\varphi \vdash \mathbf{X}\varphi$
Indution	$\varphi \supset \psi, \varphi \supset \mathbf{X}\varphi \vdash \varphi \supset \mathbf{G}\psi$

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$
- $(\neg(A \supset B)) \supset \neg B$ [taut]

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$
- $(\neg(A \supset B)) \supset \neg B$ [taut]
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}\neg B$ [as before]

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$
- $(\neg(A \supset B)) \supset \neg B$ [taut]
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}\neg B$ [as before]
- $\mathbf{X}\neg B \supset \neg\mathbf{X}B$

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$
- $(\neg(A \supset B)) \supset \neg B$ [taut]
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}\neg B$ [as before]
- $\mathbf{X}\neg B \supset \neg\mathbf{X}B$
- $(\neg\mathbf{X}(A \supset B)) \supset \neg\mathbf{X}B$

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$
- $(\neg(A \supset B)) \supset \neg B$ [taut]
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}\neg B$ [as before]
- $\mathbf{X}\neg B \supset \neg\mathbf{X}B$
- $(\neg\mathbf{X}(A \supset B)) \supset \neg\mathbf{X}B$
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}A \wedge \neg\mathbf{X}B \equiv \neg(\mathbf{X}A \supset \mathbf{X}B)$

Derivation

$$(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$$

- $(\neg(A \supset B)) \supset A$ [taut]
- $\mathbf{X}((\neg(A \supset B)) \supset A)$ [nex]
- $[\mathbf{X}((\neg(A \supset B)) \supset A)] \supset [\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A]$ (ax)
- $\mathbf{X}\neg(A \supset B) \supset \mathbf{X}A$ (mp)
- $\mathbf{X}\neg(A \supset B) \equiv \neg\mathbf{X}(A \supset B)$ (ax)
- $\neg\mathbf{X}(A \supset B) \supset \mathbf{X}A$
- $(\neg(A \supset B)) \supset \neg B$ [taut]
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}\neg B$ [as before]
- $\mathbf{X}\neg B \supset \neg\mathbf{X}B$
- $(\neg\mathbf{X}(A \supset B)) \supset \neg\mathbf{X}B$
- $(\neg\mathbf{X}(A \supset B)) \supset \mathbf{X}A \wedge \neg\mathbf{X}B \equiv \neg(\mathbf{X}A \supset \mathbf{X}B)$
- $(\mathbf{X}A \supset \mathbf{X}B) \supset \mathbf{X}(A \supset B)$

Other Theorem

$$(A \wedge \mathbf{XF}A) \supset \mathbf{F}A$$

- ...

Expressiveness of LTL formulas

- A is true only at the even states $s_0 s_2 \dots$ (false at odd ones)

Expressiveness of LTL formulas

- A is true only at the even states $s_0 s_2 \ldots$ (false at odd ones)

$$A \wedge \mathbf{G}(A \leftrightarrow \neg \mathbf{X} A)$$

Expressiveness of LTL formulas

- A is true only at the even states $s_0 s_2 \ldots$ (false at odd ones)

$$A \wedge \mathbf{G}(A \leftrightarrow \neg \mathbf{X} A)$$

- *even*(A): A is true at the even states (don't care at odd ones)

Expressiveness of LTL formulas

- A is true only at the even states $s_0 s_2 \dots$ (false at odd ones)

$$A \wedge \mathbf{G}(A \leftrightarrow \neg \mathbf{X} A)$$

- $even(A)$: A is true at the even states (don't care at odd ones)

$$A \wedge \mathbf{G}(A \supset \mathbf{XX} A) \quad \textbf{Is it ok}?$$

Expressiveness of LTL formulas

- A is true only at the even states $s_0 s_2 \ldots$ (false at odd ones)

$$A \wedge \mathbf{G}(A \leftrightarrow \neg \mathbf{X} A)$$

- *even*(A): A is true at the even states (don't care at odd ones)

$$A \wedge \mathbf{G}(A \supset \mathbf{XX} A) \quad \textbf{Is it ok}?$$

- No, it is false if A is true in s_1 and false in s_3!
  Actually, it can not be expressed in LTL

Lemma

Assumptions

- Let $M(i) = p^i(\neg p)p^\omega$ be the model in which p is true in $s_0 s_1 \ldots s_i$, false in state s_{i+1}, and true in s_j for $j > i + 1$.
- $M(0), M(2), M(4), \ldots$ all satisfy $even(p)$, while $M(1), M(3), \ldots$ don't

Lemma

- Let φ be a formula on predicate p with k occurrences of $\mathbf{X}$

Property
For $i > k$, the truth of φ on $M(i)$ is independent from i
i.e. φ has always the same value in $p^{k+1}(\neg p)p^{\omega}$, $p^{k+2}(\neg p)p^{\omega}, \ldots$
P. Wolper. Temporal Logic can be more expressive

Proof of the Lemma

By induction on the structure of φ.
Let f_i be the value of f in $M(i)$:

- φ atomic: $\varphi = p$ and p is true in $M(i)$ for all $i > 0$

Proof of the Lemma

By induction on the structure of φ.
Let f_i be the value of f in $M(i)$:

- φ atomic: $\varphi = p$ and p is true in $M(i)$ for all $i > 0$
- $\varphi = \mathbf{X}\psi$: φ in $M(i)$ evaluates as ψ in $M(i-1)$, but ψ has $k-1$ occurrences of next, so we can apply the ind. hyp.

Proof of the Lemma

By induction on the structure of φ.
Let f_i be the value of f in $M(i)$:

- φ atomic: $\varphi = p$ and p is true in $M(i)$ for all $i > 0$
- $\varphi = \mathbf{X}\psi$: φ in $M(i)$ evaluates as ψ in $M(i-1)$, but ψ has $k-1$ occurrences of next, so we can apply the ind. hyp.
- $\varphi = \mathbf{G}\psi$, i.e., $\varphi \equiv \psi \wedge \mathbf{XG}\psi$, then
  $\varphi_i \equiv \psi_i \wedge \psi_{i-1} \wedge \ldots \wedge \psi_{k+1} \wedge \varphi_k$, but by ind. hyp.
  $\psi_i \wedge \ldots \wedge \psi_{k+1} \equiv \psi_{k+1}$ have the same num. of occurrences of
  $\mathbf{X}$ as φ, i.e., $\varphi_i \equiv \psi_{k+1} \wedge \varphi_k$ (independent from i)

Proof of the Lemma

By induction on the structure of φ.
Let f_i be the value of f in $M(i)$:

- φ atomic: $\varphi = p$ and p is true in $M(i)$ for all $i > 0$
- $\varphi = \mathbf{X}\psi$: φ in $M(i)$ evaluates as ψ in $M(i-1)$, but ψ has $k-1$ occurrences of next, so we can apply the ind. hyp.
- $\varphi = \mathbf{G}\psi$, i.e., $\varphi \equiv \psi \wedge \mathbf{XG}\psi$, then $\varphi_i \equiv \psi_i \wedge \psi_{i-1} \wedge \ldots \wedge \psi_{k+1} \wedge \varphi_k$, but by ind. hyp. $\psi_i \wedge \ldots \wedge \psi_{k+1} \equiv \psi_{k+1}$ have the same num. of occurrences of $\mathbf{X}$ as φ, i.e., $\varphi_i \equiv \psi_{k+1} \wedge \varphi_k$ (independent from i)
- etc.

How to express *even*(p)?

- Automata: that recognizes $p, \mathit{true}, p, \mathit{true}, \ldots$ (infinite word)

How to express $even(p)$?

- Automata: that recognizes $p, true, p, true, \ldots$ (infinite word)
- ETL (Wolper): We can express $even(p)$ by quantifying over predicates:

$$\exists q.(q \wedge \mathbf{G}(q \supset \mathbf{X}\neg q) \wedge \mathbf{G}(\neg q \supset \mathbf{X}q) \wedge \mathbf{G}(q \supset p))$$

Model Checking

Let us consider models with several infinite paths.

- Model checking: Fixed a model M, a state s_0 and an LTL formula φ $\sigma, s_0 \models \varphi$ for every σ in M?

Model Checking

Let us consider models with several infinite paths.

- Model checking: Fixed a model M, a state s_0 and an LTL formula φ $\sigma, s_0 \models \varphi$ for every σ in M?
- Existential Model checking: Fixed a model M, a state s_0 and an LTL formula φ is there an infinite path σ in M s.t. $\sigma, s_0 \models \varphi$? (dual to the first problem)

Satisfiability/Validity

- Satisfiability problem: Fixed a formula φ, is there an infinite path σ, and a state s_0 s.t. $\sigma, s_0 \models \varphi$?

Satisfiability/Validity

- Satisfiability problem: Fixed a formula φ, is there an infinite path σ, and a state s_0 s.t. $\sigma, s_0 \models \varphi$?
- Validity problem: Fixed a formula φ, does $\sigma, s_0 \models \varphi$ hold for every infinite path σ and initial state s_0? (dual to satisfiability)

Satisfiability/Validity

- Satisfiability problem: Fixed a formula φ, is there an infinite path σ, and a state s_0 s.t. $\sigma, s_0 \models \varphi$?
- Validity problem: Fixed a formula φ, does $\sigma, s_0 \models \varphi$ hold for every infinite path σ and initial state s_0? (dual to satisfiability)
- The problems are decidable but with exponential-time in the size of φ

Model Checking Algorithm

- Tableau methods

Model Checking Algorithm

- Tableau methods
- Büchi Automata (SPIN)

Model Checking Algorithm

- Tableau methods
- Büchi Automata (SPIN)
- Alternating Büchi automata

Semantic Tableau in Classical Logic

(See slides on Aulaweb)

- Deductive method for checking a formula

Semantic Tableau in Classical Logic

(See slides on Aulaweb)

- Deductive method for checking a formula
- Goal: build a model using syntactic rules (semantic method)

LTL Model Checking Algorithm

Lichtenstein-Pnueli's Algorithm: Tableau

- For fixed M, s, φ, decide $M, s \models \psi$
- The algorithm is based on the construction of a tableau, i.e., the product of M with a syntactic model for the formula $\varphi = \neg\psi$ (solve existential model checking)
- The states of the tableau contains sets of subformulas of φ (Hintikka sets) from the closure of φ
- We restrict the construction to formulas with **X** and **U**

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it
- Formally, $cl(\varphi)$ is the smallest set of formulas that contains φ and such that:

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it
- Formally, $cl(\varphi)$ is the smallest set of formulas that contains φ and such that:
  - it contains the subformulas of φ

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it
- Formally, $cl(\varphi)$ is the smallest set of formulas that contains φ and such that:
  - it contains the subformulas of φ
  - it is closed under negation (we identify $\neg\neg F$ and F):

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it
- Formally, $cl(\varphi)$ is the smallest set of formulas that contains φ and such that:
  - it contains the subformulas of φ
  - it is closed under negation (we identify $\neg\neg F$ and F):
  - if $\neg \mathbf{X}\psi \in cl(\varphi)$, then $\mathbf{X}\neg\psi \in cl(\varphi)$

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it
- Formally, $cl(\varphi)$ is the smallest set of formulas that contains φ and such that:
  - it contains the subformulas of φ
  - it is closed under negation (we identify $\neg\neg F$ and F):
  - if $\neg \mathbf{X}\psi \in cl(\varphi)$, then $\mathbf{X}\neg\psi \in cl(\varphi)$
  - if $\psi_1 \mathbf{U} \psi_2 \in cl(\varphi)$, then $\{\psi_1, \psi_2, \mathbf{X}(\psi_1 \mathbf{U} \psi_2)\} \subseteq cl(\varphi)$ (we use the expansion axiom)

Closure of a formula

- The closure a formula of φ is the set of formulas useful to satisfy it
- Formally, $cl(\varphi)$ is the smallest set of formulas that contains φ and such that:
    - it contains the subformulas of φ
    - it is closed under negation (we identify $\neg\neg F$ and F):
    - if $\neg \mathbf{X}\psi \in cl(\varphi)$, then $\mathbf{X}\neg\psi \in cl(\varphi)$
    - if $\psi_1 \mathbf{U} \psi_2 \in cl(\varphi)$, then $\{\psi_1, \psi_2, \mathbf{X}(\psi_1 \mathbf{U} \psi_2)\} \subseteq cl(\varphi)$ (we use the expansion axiom)
- The cardinality of $cl(\varphi)$ is linear in φ

Example: Closure of a formula

$cl(p_1 \mathbf{U} p_2)$
contains
$p_1 \mathbf{U} p_2$,
p_1,
p_2,
$\mathbf{X}(p_1 \mathbf{U} p_2)$
and all negated formulas

Maximally Consistent Set

A set $X \subseteq cl(\varphi)$ is maximally consistent iff:

- (maximal) for each $\psi \in cl(\varphi)$, $\psi \in X$ or $\neg\psi \in X$

Maximally Consistent Set

A set $X \subseteq cl(\varphi)$ is maximally consistent iff:

- (maximal) for each $\psi \in cl(\varphi)$, $\psi \in X$ or $\neg\psi \in X$
- (logically consistent) for each $\psi \in cl(\varphi)$, $\psi \in X$ iff $\neg\psi \notin X$

Maximally Consistent Set

A set $X \subseteq cl(\varphi)$ is maximally consistent iff:

- (maximal) for each $\psi \in cl(\varphi)$, $\psi \in X$ or $\neg\psi \in X$
- (logically consistent) for each $\psi \in cl(\varphi)$, $\psi \in X$ iff $\neg\psi \notin X$
- (logically consistent) for each $\psi_1 \vee \psi_2 \in cl(\varphi)$, $\psi_1 \vee \psi_2 \in X$ iff $\psi_1 \in X$ or $\psi_2 \in X$

Maximally Consistent Set

A set $X \subseteq cl(\varphi)$ is maximally consistent iff:

- (maximal) for each $\psi \in cl(\varphi)$, $\psi \in X$ or $\neg\psi \in X$
- (logically consistent) for each $\psi \in cl(\varphi)$, $\psi \in X$ iff $\neg\psi \notin X$
- (logically consistent) for each $\psi_1 \vee \psi_2 \in cl(\varphi)$, $\psi_1 \vee \psi_2 \in X$ iff $\psi_1 \in X$ or $\psi_2 \in X$
- (locally consistent) for each $\psi_1 \ \mathbf{U}\ \psi_2 \in cl(\varphi)$, $\psi_1 \ \mathbf{U}\ \psi_2 \in X$ iff either $\psi_2 \in X$, or $(\{\psi_1, \mathbf{X}(\psi_1 \ \mathbf{U}\ \psi_2)\} \subseteq X)$

Tableau

Fixed a model M:

- Nodes = atoms of the form $\langle s_A, K_A \rangle$ where s_A is a state in M and K_A is a maximal consistent set compatible with the labelling of s_A
- Edge = From $\langle s_A, K_A \rangle \longrightarrow \langle s_B, K_B \rangle$ iff
  - $R(s_A) = s_B$ is a transition in M
  - For each formula $\mathbf{X}\varphi_1 \in CL(\varphi)$, $\mathbf{X}\varphi_1 \in K_A$ iff $\varphi_1 \in K_B$

Strongly Connected Component

A strongly connected component of a graph is a subgraph C s.t. for each pair $\langle n, n' \rangle$ of nodes in C there exists a path from n to n'

Self-fulfilling

A strongly connected component C is self-fulfilling iff:
for each atom B in C and for each formula $\varphi_1 \mathbf{U} \varphi_2 \in B$,
there exists an atom B' in C such that $\varphi_2 \in K_{B'}$

Theorem [Pnueli-Lichtenstein]

- Let G be the tableau associated to M and ψ
- $M, s \models_{\exists} \psi$ iff:
    - there exists an atom $A = \langle s, K \rangle$ in G s.t. $\psi \in K$
    - there exists a path in G from A to a self-fulfilling strongly connected component C of G

Depth-First Search Algorithm

- Build the tableau with size in $O((|M|) \cdot 2^{|\varphi|})$ (exponential in φ)

Depth-First Search Algorithm

- Build the tableau with size in $O((|M|) \cdot 2^{|\varphi|})$ (exponential in φ)
- Search for strongly connected components that are self-fulfilling (SFSCC) (e.g. by using Tarjan DFS-based algorithm, linear in the size of the graph)

Depth-First Search Algorithm

- Build the tableau with size in $O((|M|) \cdot 2^{|\varphi|})$ (exponential in φ)
- Search for strongly connected components that are self-fulfilling (SFSCC) (e.g. by using Tarjan DFS-based algorithm, linear in the size of the graph)

Depth-First Search Algorithm

- Build the tableau with size in $O((|M|) \cdot 2^{|\varphi|})$ (exponential in φ)
- Search for strongly connected components that are self-fulfilling (SFSCC) (e.g. by using Tarjan DFS-based algorithm, linear in the size of the graph)
- We can use again a DFS for checking reachability of a SFSCC

Complexity

- Complexity of the algorithm: $O(|M| \cdot 2^{|\varphi|})$

Complexity

- Complexity of the algorithm: $O(|M| \cdot 2^{|\varphi|})$
  - φ is often small

Complexity

- Complexity of the algorithm: $O(|M| \cdot 2^{|\varphi|})$
  - φ is often small
  - the problem is the size of M, it can be exponential in its description!

Complexity

- Complexity of the algorithm: $O(|M| \cdot 2^{|\varphi|})$
  - φ is often small
  - the problem is the size of M, it can be exponential in its description!
- The two DFS visits can be nested, and the search of an accepting state can be made on-the-fly (we build the tableau while searching for a lasso)

From Tableau to Automata

- The tableau algorithm can be viewed in terms of automata operation: product and emptiness test

From Tableau to Automata

- The tableau algorithm can be viewed in terms of automata operation: product and emptiness test
- Automata theory allows to exploit optimal algorithms for such operations

From Tableau to Automata

- The tableau algorithm can be viewed in terms of automata operation: product and emptiness test
- Automata theory allows to exploit optimal algorithms for such operations
- We need special automata that accept infinite-words