Linear-Time Logic

Hao Zheng

Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620
Email: zheng@cse.usf.edu
Phone: (813)974-4757
Fax: (813)974-5456
Overview

1. Linear Time Logic: Syntax & Semantics (Section 5.1.1 - 5.1.3)
2. Linear Time Logic: Equivalences (Section 5.1.4)
3. Linear Time Logic: Additional Operators (Section 5.1.5)
4. Linear Time Logic: Specifying Fairness (Section 5.1.6)
5. Automata-Based LTL Model Checking
LT Properties

- An LT property is a set of infinite traces over $AP$.
- Specifying such sets explicitly is often inconvenient.
- Mutual exclusion is specified over $AP = \{ c_1, c_2 \}$ by
  \[ P_{mutex} = \text{set of infinite words } A_0 A_1 A_2 \ldots \text{ with } \{ c_1, c_2 \} \not\subseteq A_i \text{ for all } i \geq 0 \]
- Starvation freedom is specified over $AP = \{ c_1, w_1, c_2, w_2 \}$ by
  \[ P_{nostarve} = \text{set of infinite words } A_0 A_1 A_2 \ldots \text{ such that:} \]
  \[ (\exists j. w_1 \in A_j) \Rightarrow (\exists j. c_1 \in A_j) \land (\exists j. w_2 \in A_j) \Rightarrow (\exists j. c_2 \in A_j) \]

Such properties can be specified succinctly using linear temporal logic.
1. Linear Time Logic: Syntax & Semantics (Section 5.1.1 - 5.1.3)

2. Linear Time Logic: Equivalences (Section 5.1.4)

3. Linear Time Logic: Additional Operators (Section 5.1.5)

4. Linear Time Logic: Specifying Fairness (Section 5.1.6)

5. Automata-Based LTL Model Checking
5.1.1 Linear Temporal Logic (LTL): Syntax

- **Linear temporal logic** is a logic for describing LT properties.
  - An extension of propositional logic with temporal modalities.
- Modal logic over infinite sequences [Pnueli 1977].
- **Propositional logic:**
  - $a \in AP$
  - $\neg \phi$ and $\phi \land \psi$
- **Temporal operators:**
  - $\bigcirc \phi$
  - $\phi U \psi$
- **Syntax of LTL over $AP$**
  
  \[
  \varphi ::= \text{true} \mid a \mid \varphi \land \varphi \mid \neg \varphi \mid \bigcirc \varphi \mid \varphi U \varphi
  \]

  where $a \in AP$ is an atomic proposition.
LTL Derived Operators

\[ \phi \lor \psi \equiv \neg (\neg \phi \land \neg \psi) \]
\[ \phi \rightarrow \psi \equiv \neg \phi \lor \psi \]
\[ \phi \leftrightarrow \psi \equiv (\phi \rightarrow \psi) \land (\psi \rightarrow \phi) \]
\[ \phi \oplus \psi \equiv (\phi \land \neg \psi) \lor (\neg \phi \land \psi) \]

true \equiv \phi \lor \neg \phi
false \equiv \neg \text{true}

\[ \Diamond \phi \equiv \text{true} \mathrel{U} \phi \quad \text{“eventually in the future”} \]
\[ \Box \phi \equiv \neg \Diamond \neg \phi \quad \text{“globally true”} \]

Precedence order:

- The unary operators bind stronger than the binary ones.
- \( \neg \) and \( \bigcirc \) bind equally strong.
- \( \mathrel{U} \) takes precedence over \( \land, \lor, \) and \( \rightarrow \).
LTL Intuitive Semantics

- **a** (atomic prop.)
  - Initial state: {a}
  - All subsequent states: \{a\}

- **\(\bigcirc a\) (next step)**
  - Initial state: {\emptyset}
  - Next state: \{a\}

- **a \land \neg b\ (until)**
  - Initial state: {a}
  - Until state: {b}

- **\neg a\ (eventually)**
  - Initial state: {\emptyset}
  - Eventual state: {a}

- **\Box a\ (globally)**
  - Initial state: {a}
  - Global state: {a}
LTL Intuitive Semantics

Let $\sigma = A_0A_1A_2 \ldots \in (2^{AP})^\omega$.

$\sigma \models a$ iff $a \in A_0$ (i.e., $A_0 \models a$)

Let $\sigma = A_0A_1A_2 \ldots \in (2^{AP})^\omega$.

$\sigma \models a$ iff $a \in A_0$ (i.e., $A_0 \models a$)
Let $\sigma = A_0 A_1 A_2 \ldots \in (2^{AP})^\omega$.

$\sigma \models \Box a$ iff $A_1 \models a$
Let $\sigma = A_0 A_1 A_2 \ldots \in (2^{AP})^\omega$.

$$\sigma \models a \mathbin{U} b \iff \exists j \geq 0. \ A_j \models b \text{ and } \forall 0 \leq i < j. \ A_i \models a$$
LTL Intuitive Semantics

Let $\sigma = A_0A_1A_2 \ldots \in (2^{AP})^\omega$.

$$\sigma \models \diamond a \iff \exists i \geq 0. A_i \models a$$
LTL Intuitive Semantics

Let $\sigma = A_0A_1A_2\ldots \in (2^{AP})^\omega$.

$\sigma \models \Box a$ iff $\forall i \geq 0. A_i \models a$
Let $\sigma = A_0A_1A_2\ldots \in (2^{AP})^\omega$.

$\Box \Diamond \varphi$ “infinitely often” $\varphi$

$\sigma \models \Box \Diamond \varphi$ iff $\forall i \geq 0 \exists j \geq i. A_j \models \varphi$
Let $\sigma = A_0 A_1 A_2 \ldots \in (2^{AP})^\omega$.

$\Diamond \Box \varphi$ “eventually forever” $\varphi$

$$\sigma \models \Diamond \Box \varphi \iff \exists i \geq 0 \ \forall j \geq i. \ A_j \models \varphi$$
Traffic Light Properties

- Once red, the light cannot become green immediately
  \( \Box (\text{red} \rightarrow \neg \Diamond \text{green}) \)

- The light becomes green eventually:
  \( \Diamond \text{green} \)

- The light becomes green infinitely often:
  \( \Box \Diamond \text{green} \)

- Once red, the light becomes green eventually:
  \( \Box (\text{red} \rightarrow \Diamond \text{green}) \)

- Once red, the light always becomes green eventually after being yellow for some time in-between:
  \( \Box(\text{red} \rightarrow \Diamond (\text{red} \cup (\text{yellow} \land \Diamond (\text{yellow} \cup \text{green})))) \)

Note these properties assume European traffic light which goes red, red/yellow, green, yellow, repeat.
LTL General Semantics (5.1.2)

Let $\sigma = A_0 A_1 A_2 \ldots \in (2^{AP})^\omega$.

$\sigma \models \text{true}$

$\sigma \models a$ iff $a \in A_0$ (i.e., $A_0 \models a$)

$\sigma \models \varphi_1 \land \varphi_2$ iff $\sigma \models \varphi_1$ and $\sigma \models \varphi_2$

$\sigma \models \neg \varphi$ iff $\sigma \not\models \varphi$

$\sigma \models \Box \varphi$ iff $\sigma[1..] = A_1 A_2 A_3 \ldots \models \varphi$

$\sigma \models \varphi_1 U \varphi_2$ iff $\exists j \geq 0. \sigma[j..] \models \varphi_2$ and $\sigma[i..] \models \varphi_1$, $0 \leq i < j$

where $\sigma[i..] = A_i A_{i+1} A_{i+2} \ldots$ is suffix of $\sigma$ from index $i$ on.
Let $\sigma = A_0A_1A_2\ldots \in (2^{AP})^\omega$.

$\sigma | = \Diamond \varphi$ iff $\exists j \geq 0. \sigma[j..] | = \varphi$

$\sigma | = \Box \varphi$ iff $\forall j \geq 0. \sigma[j..] | = \varphi$

$\sigma | = \Box \Diamond \varphi$ iff $\forall j \geq 0. \exists i \geq j. \sigma[i...] | = \varphi$

$\sigma | = \Diamond \Box \varphi$ iff $\exists j \geq 0. \forall i \geq j. \sigma[i...] | = \varphi$

where $\sigma[i..] = A_i A_{i+1} A_{i+2}\ldots$ is suffix of $\sigma$ from index $i$ on.
The LT-property induced by LTL formula $\varphi$ over $AP$ is:

$$Words(\varphi) = \left\{ \sigma \in (2^{AP})^\omega \mid \sigma \models \varphi \right\},$$

where $\models$ is the smallest satisfaction relation.
Definition 5.7  Semantics Over Paths and States

Let $TS = (S, Act, \rightarrow, I, AP, L)$ be a transition system without terminal states, and let $\varphi$ be an LTL-formula over $AP$.

- For infinite path fragment $\pi$ of $TS$:
  \[ \pi \models \varphi \iff \text{trace}(\pi) \models \varphi \]

- For state $s \in S$:
  \[ s \models \varphi \iff \forall \pi \in \text{Paths}(s). \pi \models \varphi \]

- $TS$ satisfies $\varphi$, denoted $TS \models \varphi$, iff $\text{Traces}(TS) \subseteq \text{Words}(\varphi)$
Semantics for Transition Systems

\[ TS \models \varphi \]

iff (* transition system semantics *)
\[ \text{Traces}(TS) \subseteq \text{Words}(\varphi) \]

iff (* definition of \( \models \) for LT-properties *)
\[ TS \models \text{Words}(\varphi) \]

iff (* Definition of \( \text{Words}(\varphi) \) *)
\[ \pi \models \varphi \text{ for all } \pi \in \text{Paths}(TS) \]

iff (* semantics of \( \models \) for states *)
\[ s_0 \models \varphi \text{ for all } s_0 \in I \]
LTL Examples

\[ TS \models \square a ? \]
LTL Examples

\[ \{ a, b \} \subseteq s_1 \]
\[ \{ a, b \} \subseteq s_2 \]
\[ \{ a \} \subseteq s_3 \]

\[ TS \models □ a \]

\[ TS \models □ (a \land b) ? \]
LTL Examples

\(TS \models \square a\)

\(TS \not\models \bigcirc (a \land b)\)

\(TS \models \square (\neg b \rightarrow \square (a \land \neg b))\)?
LTL Examples

\( \text{TS} \models \square a \)

\( \text{TS} \not\models \bigcirc (a \land b) \)

\( \text{TS} \models \square (\neg b \rightarrow \square (a \land \neg b)) \)
LTL Examples

\[ TS \models \Box a \]

\[ TS \not\models \Box (a \land b) \]

\[ TS \models \Box (\neg b \rightarrow \Box (a \land \neg b)) \]

\[ TS \models b \mathbf{U} (a \land \neg b) ? \]
LTL Examples

\[ TS \models □a \]
\[ TS \not\models □(\neg b \rightarrow □(a \land \neg b)) \]
\[ TS \not\models \Diamond_a \Diamond (a \land \neg b) \]
Semantics of Negation

- For paths, it holds $\pi \models \varphi$ if and only if $\pi \not\models \neg \varphi$ since:

$$Words(\neg \varphi) = (2^{AP})^\omega \setminus Words(\varphi).$$

- But: $TS \not\models \varphi$ and $TS \models \neg \varphi$ are not equivalent in general.

- It holds: $TS \models \neg \varphi$ implies $TS \not\models \varphi$, not always the reverse!

- Note that:

$$TS \not\models \varphi \quad \text{iff} \quad \text{Traces}(TS) \not\subseteq Words(\varphi)$$

$$\quad \quad \quad \quad \quad \quad \text{iff} \quad \text{Traces}(TS) \setminus Words(\varphi) \neq \emptyset$$

$$\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \text{iff} \quad \text{Traces}(TS) \cap Words(\neg \varphi) \neq \emptyset.$$

- $TS$ neither satisfies $\varphi$ nor $\neg \varphi$ if there are paths $\pi_1$ and $\pi_2$ in $TS$ such that $\pi_1 \models \varphi$ and $\pi_2 \models \neg \varphi$. 
A transition system for which $\text{TS} \notmodels \Diamond a$ and $\text{TS} \notmodels \neg \Diamond a$. 
Example 5.13 Leader Election

- $N$ processes, each of which has a unique identity. Leader process is the one that has the largest ID.
Example 5.13 Leader Election

- $N$ processes, each of which has a unique identity. Leader process is the one that has the largest ID.
- **There is always one leader**

\[\Box( \bigvee_{1 \leq i \leq N} \text{leader}_i \land \bigvee_{1 \leq j \leq N, j \neq i} \neg \text{leader}_j)\]

\[\Box\Diamond( \bigvee_{1 \leq i \leq N} \text{leader}_i \land \bigvee_{1 \leq j \leq N, j \neq i} \neg \text{leader}_j)\]

\[\Box\Diamond( \bigvee_{1 \leq i \leq N} \text{leader}_i)\]

\[\Diamond\Box( \bigvee_{1 \leq i \leq N} \text{leader}_i)\]
Example 5.13 Leader Election

- $N$ processes, each of which has a unique identity. Leader process is the one that has the largest ID.
- *There must always be at most one leader*
Example 5.13 Leader Election

- $N$ processes, each of which has a unique identity. Leader process is the one that has the largest ID.
- *There must always be at most one leader*

\[
\square \bigwedge_{1 \leq i \leq N} (leader_i \rightarrow \bigwedge_{1 \leq j \leq N, j \neq i} \neg leader_j)
\]
Example 5.13 Leader Election

- $N$ processes, each of which has a unique identity. Leader process is the one that has the largest ID.
- There must always be at most one leader

$$\square \bigwedge_{1 \leq i \leq N} (leader_i \rightarrow \bigwedge_{1 \leq j \leq N, j \neq i} \neg leader_j)$$

- A correct leader will be elected eventually.
Contents

1. Linear Time Logic: Syntax & Semantics (Section 5.1.1 - 5.1.3)
2. Linear Time Logic: Equivalences (Section 5.1.4)
3. Linear Time Logic: Additional Operators (Section 5.1.5)
4. Linear Time Logic: Specifying Fairness (Section 5.1.6)
5. Automata-Based LTL Model Checking
5.1.4 Equivalence

LTL formulas $\phi, \psi$ are equivalent, denoted $\phi \equiv \psi$, if:

$$\text{Words}(\phi) = \text{Words}(\psi)$$

- Recall that the time complexity for invariant checking is:

$$\mathcal{O}(N \ast (1 + |\Phi|) + M)$$

where

- $N$ is the number of reachable states,
- $M$ is the number of transitions in the reachable fragment of $TS$, and
- $|\Phi|$ is the length of $\Phi$ - number of logic connectives in $\Phi$
Duality and Idempotence Laws

Duality:

\[ \neg \Box \phi \equiv \Diamond \neg \phi \]
\[ \neg \Diamond \phi \equiv \Box \neg \phi \]
\[ \neg \lozenge \phi \equiv \lozenge \neg \phi \]

Idempotency:

\[ \Box \Box \phi \equiv \Box \phi \]
\[ \Diamond \Diamond \phi \equiv \Diamond \phi \]
\[ \phi \cup (\phi \cup \psi) \equiv \phi \cup \psi \]
\[ (\phi \cup \psi) \cup \psi \equiv \phi \cup \psi \]
Absorption and Distributive Laws

Absorption:  \[\Diamond \Box \Diamond \phi \equiv \Box \Diamond \phi\]
\[\Box \Diamond \Box \phi \equiv \Diamond \Box \phi\]

Distribution:  \[\lozenge (\phi \cup \psi) \equiv (\lozenge \phi) \cup (\lozenge \psi)\]
\[\lozenge (\phi \vee \psi) \equiv \lozenge \phi \vee \lozenge \psi\]
\[\Box (\phi \land \psi) \equiv \Box \phi \land \Box \psi\]

but \ldots:  \[\Diamond (\phi \cup \psi) \neq (\Diamond \phi) \cup (\Diamond \psi)\]
\[\Diamond (\phi \land \psi) \neq \Diamond \phi \land \Diamond \psi\]
\[\Box (\phi \lor \psi) \neq \Box \phi \lor \Box \psi\]
Distributive Laws

\[ (a \land b) \land a = a \land (b \lor a) \]

Chris J. Myers (Lecture 5: LTL) Verification of Cyber-Physical Systems

\[ TS \not\models \diamond(a \land b) \text{ and } TS \models (\diamond a \land \diamond b) \]

Hao Zheng (CSE, USF) Comp Sys Verification
Define $U$, $\diamond$, and $\Box$ by recursion.

**Expansion:**

\[
\Box \phi \equiv \phi \land \Box \Box \phi \\
\diamond \phi \equiv \phi \lor \Box \diamond \phi \\
\phi U \psi \equiv \psi \lor (\phi \land \Box (\phi U \psi))
\]
Contents

1. Linear Time Logic: Syntax & Semantics (Section 5.1.1 - 5.1.3)

2. Linear Time Logic: Equivalences (Section 5.1.4)

3. Linear Time Logic: Additional Operators (Section 5.1.5)

4. Linear Time Logic: Specifying Fairness (Section 5.1.6)

5. Automata-Based LTL Model Checking
5.1.5 Weak Until

- The \textit{weak-until} (or: unless) operator:

\[
\varphi W \psi \overset{\text{def}}{=} (\varphi U \psi) \lor \Box \varphi
\]

- \(\varphi W \psi\) does not require a \(\psi\)-state to be reached.

- Until \(U\) and weak until \(W\) are \textit{dual}:

\[
\neg (\varphi U \psi) \equiv (\varphi \land \neg \psi) W (\neg \varphi \land \neg \psi)
\]

\[
\neg (\varphi W \psi) \equiv (\varphi \land \neg \psi) U (\neg \varphi \land \neg \psi)
\]

- Until and weak until are \textit{equally expressive}:

\[
\Box \psi \equiv \psi W \text{ false}
\]

\[
\varphi U \psi \equiv (\varphi W \psi) \land \neg \Box \neg \psi
\]
The Release Operator

- The *release* operator:

\[
\varphi \text R \psi \quad \overset{\text{def}}{=} \quad \neg (\neg \varphi \cup \neg \psi)
\]

\[
\overset{\text{def}}{=} \quad (\neg \varphi \wedge \psi) W (\varphi \wedge \psi)
\]

- \(\psi\) always holds, a requirement that is released as soon as \(\varphi\) holds.
Recall Action-Based Fairness Constraints

For set $A$ of actions and infinite run $\rho$:

- **Unconditional fairness**
  Some action in $A$ occurs infinitely often along $\rho$.

- **Strong fairness**
  If actions in $A$ are *infinitely often* enabled (not necessarily always!) then some action in $A$ has to occur infinitely often in $\rho$.

- **Weak fairness**
  If actions in $A$ are *continuously enabled* (no temporary disabling!) then it has to occur infinitely often in $\rho$.

This chapter uses *state-based* fairness assumptions (and constraints).
5.1.6 LTL Fairness Constraints

Let $\Phi$ and $\Psi$ be propositional logic formulas over $AP$.

1. An **unconditional LTL fairness constraint** is of the form:

   $$ ufair = \Box\Diamond\Psi $$

2. A **strong LTL fairness condition** is of the form:

   $$ sfair = \Box\Diamond\Phi \rightarrow \Box\Diamond\Psi $$

3. A **weak LTL fairness constraint** is of the form:

   $$ wfair = \Diamond\Box\Phi \rightarrow \Box\Diamond\Psi $$

$\Phi$ stands for “something is enabled”; $\Psi$ for “something is taken”
Fair Satisfaction

For state $s$ in transition system $TS$ (over $AP$) without terminal states, let

$$FairPaths_{\text{fair}}(s) = \{ \pi \in \text{Paths}(s) \mid \pi \models \text{fair} \}$$

$$FairTraces_{\text{fair}}(s) = \{ \text{trace}(\pi) \mid \pi \in FairPaths_{\text{fair}}(s) \}$$

For LTL-formula $\varphi$, and LTL fairness assumption $\text{fair}$:

$$s \models_{\text{fair}} \varphi \text{ if and only if } \forall \pi \in FairPaths_{\text{fair}}(s). \pi \models \varphi \text{ and }$$

$$TS \models_{\text{fair}} \varphi \text{ if and only if } \forall s_0 \in I. s_0 \models_{\text{fair}} \varphi$$

$\models_{\text{fair}}$ is the fair satisfaction relation for LTL; $\models$ the standard one for LTL
Example 5.27  Randomized Arbiter

\[
\begin{align*}
TS_1 \parallel \text{Arbiter} \parallel TS_2 & \not\models \Box \Diamond \text{crit}_1 \\
\text{But: } TS_1 \parallel \text{Arbiter} \parallel TS_2 & \models_{\text{fair}} \Box \Diamond \text{crit}_1 \land \Box \Diamond \text{crit}_2 \\
& \text{with } \text{fair} = \Box \Diamond \text{head} \land \Box \Diamond \text{tail}
\end{align*}
\]
Semaphore-Based Mutual Exclusion

\[ \langle n_1, n_2, y = 1 \rangle \]
\[ \langle w_1, n_2, y = 1 \rangle \]
\[ \langle n_1, w_2, y = 1 \rangle \]
\[ \langle c_1, n_2, y = 0 \rangle \]
\[ \langle w_1, w_2, y = 1 \rangle \]
\[ \langle n_1, c_2, y = 0 \rangle \]
\[ \langle c_1, w_2, y = 0 \rangle \]

\[ sfair_1 = \Box \Diamond wait_1 \rightarrow \Box \Diamond crit_1 \]
Semaphore-Based Mutual Exclusion

\[ \langle n_1, n_2, y=1 \rangle \]

\[ \langle w_1, n_2, y=1 \rangle \]

\[ \langle n_1, w_2, y=1 \rangle \]

\[ \langle c_1, n_2, y=0 \rangle \]

\[ \langle w_1, w_2, y=1 \rangle \]

\[ \langle n_1, c_2, y=0 \rangle \]

\[ \langle c_1, w_2, y=0 \rangle \]

\[ \langle w_1, c_2, y=0 \rangle \]

\[ \text{fair} = sfair_1 \land sfair_2 \]
Semaphore-Based Mutual Exclusion

\[
\langle n_1, n_2, y = 1 \rangle \\
\langle w_1, n_2, y = 1 \rangle \quad \langle n_1, w_2, y = 1 \rangle \\
\langle c_1, n_2, y = 0 \rangle \quad \langle w_1, c_2, y = 0 \rangle \\
\langle c_1, w_2, y = 0 \rangle \quad \langle w_1, w_2, y = 1 \rangle
\]

\[\text{fair} = \text{sfair}_1 \land \text{sfair}_2\]

\[\mathcal{T}S_{\text{Sem}} \models \text{fair} \quad \square \Diamond \text{crit}_1 \land \square \Diamond \text{crit}_2\]
Theorem 5.30  Reducing $\models_{\text{fair}}$ to $\models$

For:
- A transition system $TS$ without terminal states
- LTL formula $\varphi$, and
- LTL fairness assumption $\text{fair}$

It holds:

$$TS \models_{\text{fair}} \varphi \quad \text{if and only if} \quad TS \models (\text{fair} \to \varphi)$$

Verifying an LTL-formula under a fairness assumption can be done using standard verification algorithms for LTL.
The following decision problem:

Given finite transition system $TS$ and LTL-formula $\varphi$:
yields “yes” if $TS \models \varphi$, and “no” (plus a counterexample) if $TS \not\models \varphi$

*See section 5.2 for details.*
A First Attempt

\[ TS \models \varphi \quad \text{if and only if} \quad \text{Traces}(TS) \subseteq \underbrace{\text{Words}(\varphi)}_{L_\infty(A_\varphi)} \]

\[ \text{if and only if} \quad \text{Traces}(TS) \cap L_\infty(A_{\overline{\varphi}}) = \emptyset \]

But complementation of NBA is quadratically exponential.

If \( A \) has \( n \) states, \( \overline{A} \) has \( c^{n^2} \) states in worst case!

Use the fact that \( L_\infty(A_{\overline{\varphi}}) = L_\infty(A_{\overline{\neg \varphi}}) \)!
Observation

\[ TS \models \varphi \text{ if and only if } Traces(TS) \subseteq Words(\varphi) \]

\[ \text{if and only if } Traces(TS) \cap ((2^{AP})^\omega \setminus Words(\varphi)) = \emptyset \]

\[ \text{if and only if } Traces(TS) \cap Words(\neg \varphi) = \emptyset \]

\[ \text{if and only if } TS \otimes A_{\neg \varphi} \models \Diamond \Box \neg F \text{ where } F \]

is the set of accepting states of \( A_{\neg \varphi} \).

\[ LTL \text{ model checking is thus reduced to persistence checking!} \]
Some Examples: LTL to BGA

\[ \Box \Diamond \text{green} \]

\[ \Box (a \rightarrow \Diamond b) \]

\[ \Diamond \Box a \]
Overview of LTL Model Checking

System

Model of system

Negation of property

LTL-formula $\neg \phi$

model checker

Transition system $TS$

Generalised Büchi automaton $G_{\neg \phi}$

Büchi automaton $A_{\neg \phi}$

Product transition system $TS \otimes A_{\neg \phi}$

$TS \otimes A_{\neg \phi} \models P_{\text{pers}}(A_{\neg \phi})$

‘Yes’

‘No’ (counter-example)