LTL Model Checking

Lecture #3 of Principles of Model Checking

Joost-Pieter Katoen

Software Modeling and Verification Group

affiliated to University of Twente, Formal Methods and Tools

University of Twente, September 29, 2009
Content of this lecture

- **Linear temporal logic**
  - syntax, semantics, specifying properties

- **Equivalences and fairness**
  - weak until, release, fairness in LTL

- **LTL model checking**
  - GNBA, from LTL to GNBA, complexity
Content of this lecture

⇒ Linear temporal logic
  – syntax, semantics, specifying properties

• Equivalences and fairness
  – weak until, release, fairness in LTL

• LTL model checking
  – GNBA, from LTL to GNBA, complexity
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

$$P_{mutex} = \text{set of infinite words } A_0 A_1 A_2 \ldots \text{ with } \{c_1, c_2\} \not\subseteq A_i \text{ for all } 0 \leq i$$

• Starvation freedom is specified over $AP = \{c_1, w_1, c_2, w_2\}$ by

$$P_{nostarve} = \text{set of infinite words } A_0 A_1 A_2 \ldots \text{ such that:}$$

$$\left(\exists j. w_1 \in A_j \right) \Rightarrow \left(\exists j. c_1 \in A_j \right) \land \left(\exists j. w_2 \in A_j \right) \Rightarrow \left(\exists j. c_2 \in A_j \right)$$

such properties can be specified succinctly using logic
Linear Temporal Logic: Syntax

- **Propositional logic**
  - $a \in AP$
  - $\neg \phi$ and $\phi \land \psi$
  - atomic proposition
  - negation and conjunction

- **Temporal operators**
  - $\bigcirc \phi$
  - $\phi U \psi$
  - neXt state fulfills $\phi$
  - $\phi$ holds Until a $\psi$-state is reached

linear temporal logic is a logic for describing LT properties
Derived operators

\[ \phi \lor \psi \equiv \neg (\neg \phi \land \neg \psi) \]
\[ \phi \Rightarrow \psi \equiv \neg \phi \lor \psi \]
\[ \phi \leftrightarrow \psi \equiv (\phi \Rightarrow \psi) \land (\psi \Rightarrow \phi) \]
\[ \phi \oplus \psi \equiv (\phi \land \neg \psi) \lor (\neg \phi \land \psi) \]
\[ \text{true} \equiv \phi \lor \neg \phi \]
\[ \text{false} \equiv \neg \text{true} \]
\[ \Diamond \phi \equiv \text{true} \lor \phi \quad \text{“sometimes in the future”} \]
\[ \Box \phi \equiv \neg \Diamond \neg \phi \quad \text{“from now on for ever”} \]

precedence order: the unary operators bind stronger than the binary ones.
\[ \neg \] and \[ \bigcirc \] bind equally strong. \[ \lor \] takes precedence over \[ \land \], \[ \lor \], and \[ \Rightarrow \].
Intuitive semantics

atomic prop. \( a \)  
\[
\begin{array}{cccccc}
& a & \text{arbitrary} & \text{arbitrary} & \text{arbitrary} & \text{arbitrary} \\
\hline
\text{next step} &  & \text{arbitrary} & a & \text{arbitrary} & \text{arbitrary} & \text{arbitrary} & \text{arbitrary} \\
\text{until} & a \lor b & a \land \lnot b & a \land \lnot b & b & \text{arbitrary} \\
\text{eventually} & \lnot a & \lnot a & \lnot a & a & \text{arbitrary} \\
\text{always} & a & a & a & a & a & a \\
\end{array}
\]
LT properties

- Mutual exclusion is specified over $AP = \{ c_1, c_2 \}$ by

$$P_{mutex} = \text{set of infinite words } A_0 A_1 A_2 \ldots \text{ with } \{ c_1, c_2 \} \not\subseteq A_i \text{ for all } 0 \leq i$$

- In LTL: $\Box \neg (c_1 \land c_2)$

- Starvation freedom is specified over $AP = \{ c_1, w_1, c_2, w_2 \}$ by

$$P_{nostarve} = \text{set of infinite words } A_0 A_1 A_2 \ldots \text{ such that:}$$

$$\left( \exists j. w_1 \in A_j \right) \Rightarrow \left( \exists j. c_1 \in A_j \right) \land \left( \exists j. w_2 \in A_j \right) \Rightarrow \left( \exists j. c_2 \in A_j \right)$$

- In LTL: $(\Box \Diamond w_1 \Rightarrow \Box \Diamond c_1) \land (\Box \Diamond w_2 \Rightarrow \Box \Diamond c_2)$
Semantics over words

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 relation satisfying:

- $\sigma \models true$
- $\sigma \models a$ 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 $\sigma \not\models \varphi$
- $\sigma \models \bigcirc \varphi$ iff $\sigma[1..] = A_1A_2A_3\ldots \models \varphi$
- $\sigma \models \varphi_1 U \varphi_2$ iff $\exists j \geq 0. \sigma[j..] \models \varphi_2$ and $\sigma[i..] \models \varphi_1$, $0 \leq i < j$

for $\sigma = A_0A_1A_2\ldots$ we have $\sigma[i..] = A_iA_{i+1}A_{i+2}\ldots$ is the suffix of $\sigma$ from index $i$ on
Semantics of $\Box$, $\Diamond$, $\Box\Diamond$ and $\Diamond\Box$

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

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

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

$\sigma \models \Diamond \Box \varphi$ iff $\exists j \geq 0. \forall i \geq j. \sigma[i..] \models \varphi$
Semantics over paths and states

Let $TS = (S, Act, \rightarrow, I, AP, L)$ and $\varphi$ 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 $\text{Traces}(TS) \subseteq \text{Words}(\varphi)$
Example

\[
\begin{align*}
TS & \models \Box a \\
TS & \models \Box (\neg b \Rightarrow \Box (a \land \neg b)) \\
TS & \not\models \Diamond (a \land b) \\
TS & \not\models b \cup (a \land \neg b)
\end{align*}
\]
Practical properties in LTL

- **Reachability**
  - simple reachability
  - conditional reachability

- **Safety**
  - invariant

- **Liveness**

- **Fairness**
  - \(\Box (\phi \Rightarrow \Diamond \psi)\) and others
  - \(\Box \Diamond \phi\) and others
Semantics of negation

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

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

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

It holds: $TS \models \lnot \varphi$ implies $TS \not\models \varphi$. Not always the reverse!

Note that:

$$TS \not\models \varphi \iff Traces(TS) \nsubseteq Words(\varphi)$$
$$\iff Traces(TS) \setminus Words(\varphi) \neq \emptyset$$
$$\iff Traces(TS) \cap Words(\lnot \varphi) \neq \emptyset.$$
A transition system for which $TS \not\models \Diamond a$ and $TS \not\models \neg\Diamond a$
Content of this lecture

- **Linear temporal logic**
  - syntax, semantics, specifying properties

  \[ \Rightarrow \text{Equivalences and fairness} \]
  - weak until, release, fairness in LTL

- **LTL model checking**
  - GNBA, from LTL to GNBA, complexity
Equivalence

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

$$\text{Words}(\phi) = \text{Words}(\psi)$$
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:
\[ \Diamond (\phi \cup \psi) \equiv (\Diamond \phi) \cup (\Diamond \psi) \]
\[ \Diamond (\phi \vee \psi) \equiv \Diamond \phi \vee \Diamond \psi \]
\[ \Box (\phi \land \psi) \equiv \Box \phi \land \Box \psi \]

but .......
\[ \Diamond (\phi \cup \psi) \not\equiv (\Diamond \phi) \cup (\Diamond \psi) \]
\[ \Diamond (\phi \land \psi) \not\equiv \Diamond \phi \land \Diamond \psi \]
\[ \Box (\phi \vee \psi) \not\equiv \Box \phi \vee \Box \psi \]
Distributive laws

\[ \Diamond (a \land b) \not\equiv \Diamond a \land \Diamond b \quad \text{and} \quad \Box (a \lor b) \not\equiv \Box a \lor \Box b \]

\[ TS \not\models \Diamond (a \land b) \quad \text{and} \quad TS \models \Diamond a \land \Diamond b \]
Expansion laws

Expansion:

\[ \phi \mathcal{U} \psi \equiv \psi \lor (\phi \land \Box (\phi \mathcal{U} \psi)) \]

\[ \Diamond \phi \equiv \phi \lor \Box \Diamond \phi \]

\[ \lozenge \phi \equiv \phi \land \Box \lozenge \phi \]

proof on the black board
Expansion for until

\[ P = \text{Words}(\varphi \cup \psi) \] satisfies:

\[ P = \text{Words}(\psi) \cup \left\{ A_0A_1A_2 \ldots \in \text{Words}(\varphi) \mid A_1A_2 \ldots \in P \right\} \]

and is the \textit{smallest} LT-property such that:

\[ \text{Words}(\psi) \cup \left\{ A_0A_1A_2 \ldots \in \text{Words}(\varphi) \mid A_1A_2 \ldots \in P \right\} \subseteq P \quad (*) \]

smallest LT-property satisfying condition (*) means that:

\[ P = \text{Words}(\varphi \cup \psi) \text{ satisfies (*) and } \text{Words}(\varphi \cup \psi) \subseteq P \text{ for each } P \text{ satisfying (*)} \]
Weak until

- The \textit{weak-until} (or: unless) operator: $\varphi W \psi \overset{\text{def}}{=} (\varphi U \psi) \lor \Box \varphi$
  - as opposed to until, $\varphi W \psi$ 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}:
  
  - $\Box \psi \equiv \psi W \text{ false and } \varphi U \psi \equiv (\varphi W \psi) \land \neg \Box \neg \psi$

- Until and weak until satisfy the \textit{same expansion law}
  
  - but until is the smallest, and weak until the largest solution!
Expansion for weak until

\[ P = \text{Words}(\varphi W \psi) \] satisfies:

\[ P = \text{Words}(\psi) \cup \{ A_0A_1A_2\ldots \in \text{Words}(\varphi) \mid A_1A_2\ldots \in P \} \]

and is the *largest* LT-property such that:

\[ \text{Words}(\psi) \cup \{ A_0A_1A_2\ldots \in \text{Words}(\varphi) \mid A_1A_2\ldots \in P \} \supseteq P \quad (***) \]

*largest LT-property satisfying condition (****) means that:*

\[ P \supseteq \text{Words}(\varphi W \psi) \] satisfies (***), and \[ \text{Words}(\varphi W \psi) \supseteq P \] for each \( P \) satisfying (***).
The release operator

- The release operator: \( \varphi R \psi \overset{\text{def}}{=} \neg(\neg \varphi U \neg \psi) \)
  - \( \psi \) always holds, a requirement that is released as soon as \( \varphi \) holds

- Until \( U \) and release \( R \) are dual:
  \[
  \varphi U \psi \equiv \neg(\neg \varphi R \neg \psi) \\
  \varphi R \psi \equiv \neg(\neg \varphi U \neg \psi)
  \]

- Until and release are equally expressive:
  - \( \square \psi \equiv \text{false} R \psi \) and \( \varphi U \psi \equiv \neg(\neg \varphi R \neg \psi) \)

- Release satisfies the expansion law: \( \varphi R \psi \equiv \psi \land (\varphi \lor \Box (\varphi R \psi)) \)
Semantics of release

\[ \sigma \models \varphi \land \psi \]

iff

\[ \neg \exists j \geq 0. \left( \sigma[j..] \models \neg \psi \land \forall i < j. \sigma[i..] \models \neg \varphi \right) \]

(* definition of \( R \) *)

iff

\[ \neg \exists j \geq 0. \left( \sigma[j..] \not\models \psi \land \forall i < j. \sigma[i..] \not\models \varphi \right) \]

(* semantics of negation *)

iff

\[ \forall j \geq 0. \neg \left( \sigma[j..] \not\models \psi \land \forall i < j. \sigma[i..] \not\models \varphi \right) \]

(* duality of \( \exists \) and \( \forall \) *)

iff

\[ \forall j \geq 0. \left( \neg (\sigma[j..] \not\models \psi) \lor \neg \forall i < j. \sigma[i..] \not\models \varphi \right) \]

(* de Morgan’s law *)

iff

\[ \forall j \geq 0. \left( \sigma[j..] \models \psi \lor \exists i < j. \sigma[i..] \models \varphi \right) \]

(* semantics of negation *)

iff

\[ \forall j \geq 0. \sigma[j..] \models \psi \lor \exists i \geq 0. \left( \sigma[i..] \models \varphi \right) \land \forall k \leq i. \sigma[k..] \models \psi \]
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”
LTL fairness assumption

- **LTL fairness assumption** = conjunction of LTL fairness constraints
  - the fairness constraints are of any arbitrary type

- **Strong fairness assumption**: $sfair = \wedge_{0 < i \leq k} \left( \square \diamond \Phi_i \implies \square \diamond \Psi_i \right)$
  - compare this to an action-based strong fairness constraint over $A$ with $|A| = k$

- **General format**: $fair = ufair \land sfair \land wfair$

- **Rules of thumb**:  
  - strong (or unconditional) fairness assumptions are useful for solving contentions
  - weak fairness suffices for resolving nondeterminism resulting from interleaving
**Fair satisfaction**

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

\[
\begin{align*}
\text{FairPaths}_{\text{fair}}(s) &= \left\{ \pi \in \text{Paths}(s) \mid \pi \models \text{fair} \right\} \\
\text{FairTraces}_{\text{fair}}(s) &= \left\{ \text{trace}(\pi) \mid \pi \in \text{FairPaths}_{\text{fair}}(s) \right\}
\end{align*}
\]

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

\[
\begin{align*}
s \models_{\text{fair}} \varphi & \quad \text{if and only if} \quad \forall \pi \in \text{FairPaths}_{\text{fair}}(s). \pi \models \varphi \quad \text{and} \\
TS \models_{\text{fair}} \varphi & \quad \text{if and only if} \quad \forall s_0 \in I. s_0 \models_{\text{fair}} \varphi
\end{align*}
\]

\( \models_{\text{fair}} \) is the *fair satisfaction relation* for LTL; \( \models \) the standard one for LTL
Randomized arbiter

$$TS_1 \parallel Arbiter \parallel TS_2 \not\models \square \diamond crit_1$$

But: $$TS_1 \parallel Arbiter \parallel TS_2 \models_{fair} \square \diamond crit_1 \land \square \diamond crit_2$$ with $$fair = \square \diamond head \land \square \diamond tail$$
Reducing $\models_{fair} \text{ to } \models$

For:

- transition system $TS$ without terminal states
- LTL formula $\varphi$, and
- LTL fairness assumption $fair$

it holds:

$TS \models_{fair} \varphi$ if and only if $TS \models (fair \rightarrow \varphi)$

verifying an LTL-formula under a fairness assumption can be done using standard verification algorithms for LTL
Content of this lecture

• Linear temporal logic
  – syntax, semantics, specifying properties

• Equivalences and fairness
  – weak until, release, fairness in LTL

⇒ LTL model checking
  – GNBA, from LTL to GNBA, complexity
LTL model-checking problem

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 $TS \not\models \varphi$
NBA for LTL-formulae
A first attempt

\[ TS \models \varphi \quad \text{if and only if} \quad \text{Traces}(TS) \subseteq \overline{\mathcal{L}_\omega(\mathcal{A}_\varphi)} \]

\[ \text{if and only if} \quad \text{Traces}(TS) \cap \mathcal{L}_\omega(\mathcal{A}_\varphi) = \emptyset \]

\[ \text{if and only if} \quad \text{Traces}(TS) \cap \overline{\mathcal{L}_\omega(\mathcal{A}_\varphi)} = \emptyset \]

but complementation of NBA is quadratically exponential

if \( \mathcal{A} \) has \( n \) states, \( \overline{\mathcal{A}} \) has \( c^{n^2} \) states in worst case
Observation

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

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

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

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

\textit{LTL model checking is thus reduced to persistence checking!}
Overview of LTL model checking

System → Model of system → LTL-formula $\neg \varphi$ → Generalised Büchi automaton $G_{\neg \varphi}$ → Büchi automaton $A_{\neg \varphi}$ → Product transition system $TS \otimes A_{\neg \varphi}$ → $TS \otimes A_{\neg \varphi} \models P_{pers}(A_{\neg \varphi})$ → ‘Yes’

Negation of property → Transition system $TS$ → Büchi automaton $A_{\neg \varphi}$ → Product transition system $TS \otimes A_{\neg \varphi}$ → $TS \otimes A_{\neg \varphi} \models P_{pers}(A_{\neg \varphi})$ → ‘No’ (counter-example)
Generalized Büchi automata

• NBA are as expressive as $\omega$-regular languages

• Variants of NBA exist that are equally expressive
  – Muller, Rabin, Streett automata, and generalized Büchi automata (GNBA)

• GNBA are like NBA, but have a distinct acceptance criterion
  – a GNBA requires to visit several sets $F_1, \ldots, F_k$ ($k \geq 0$) infinitely often
  – for $k=0$, all runs are accepting; for $k=1$ it behaves like an NBA

• GNBA are useful to relate temporal logic and automata
Generalized Büchi automata

A generalized NBA (GNBA) $G$ is a tuple $(Q, \Sigma, \delta, Q_0, F)$ where:

- $Q, \Sigma, \delta$ and $Q_0$ are as before, and

- $F = \{ F_1, \ldots, F_k \}$ is a (possibly empty) subset of $2^Q$
Language of a GNBA

- GNBA $\mathcal{G} = (Q, \Sigma, \delta, Q_0, \mathcal{F})$ and word $\sigma = A_0A_1A_2\ldots \in \Sigma^\omega$

- A **accepted run** for $\sigma$ in $\mathcal{G}$ is an infinite sequence $q_0q_1q_2\ldots$ such that:
  - $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for all $0 \leq i$, and
  - for all $F \in \mathcal{F}$: $q_i \in F$ for infinitely many $i$

- $L_\omega(\mathcal{G}) = \{ \sigma \in \Sigma^\omega \mid$ there exists an accepting run for $\sigma$ in $\mathcal{G} \}$
Example

A GNBA for the property "both processes are infinitely often in their critical section"

\[ \mathcal{F} = \{ \{ q_1 \}, \{ q_2 \} \} \]
From GNBA to NBA

For any GNBA $\mathcal{G}$ there exists an NBA $\mathcal{A}$ with:

$\mathcal{L}_\omega(\mathcal{G}) = \mathcal{L}_\omega(\mathcal{A})$ and $|\mathcal{A}| = O(|\mathcal{G}| \cdot |\mathcal{F}|)$

where $\mathcal{F}$ denotes the set of acceptance sets in $\mathcal{G}$

Sketch of transformation GNBA (with $|\mathcal{F}| = k$) into equivalent NBA:

- make $k$ copies of the GNBA
- initial states of NBA := the initial states in the first copy
- final states of NBA := accept set $F_1$ in the first copy
- on visiting in $i$-th copy a state in $F_i$, then move to the $(i+1)$-st copy
Example
From LTL to GNBA

GNBA $\mathcal{G}_\varphi$ over $2^{AP}$ for LTL-formula $\varphi$ with $L_\omega(\mathcal{G}_\varphi) = \text{Words}(\varphi)$:

- Assume $\varphi$ only contains the operators $\land$, $\neg$, $\circ$ and $U$

- States are elementary sets of sub-formulas in $\varphi$
  - for $\sigma = A_0A_1A_2 \ldots \in \text{Words}(\varphi)$, “expand” $A_i \subseteq AP$ with sub-formulas of $\varphi$
  - $\ldots$ to obtain $\bar{\sigma} = B_0B_1B_2 \ldots$ such that

  \[
  \psi \in B_i \text{ if and only if } \sigma^i = A_iA_{i+1}A_{i+2} \ldots \models \psi
  \]

- $\bar{\sigma}$ is intended to be a run in GNBA $\mathcal{G}_\varphi$ for $\sigma$

- Transitions are derived from semantics $\circ$ and expansion law for $U$

- Accept sets guarantee that: $\bar{\sigma}$ is an accepting run for $\sigma$ iff $\sigma \models \varphi$
From LTL to GNBA: the states (example)

- Let $\varphi = a \cup (\neg a \land b)$ and $\sigma = \{ a \} \{ a, b \} \{ b \} \ldots$
  - $B_i$ is a subset of $\{ a, b, \neg a \land b, \varphi \} \cup \{ \neg a, \neg b, \neg(\neg a \land b), \neg \varphi \}$
  - this set of formulas is also called the closure of $\varphi$

- Extend $A_0 = \{ a \}$, $A_1 = \{ a, b \}$, $A_2 = \{ b \}$, $\ldots$ as follows:
  - extend $A_0$ with $\neg b$, $\neg(\neg a \land b)$, and $\varphi$ as they hold in $\sigma^0 = \sigma$ (and no others)
  - extend $A_1$ with $\neg(\neg a \land b)$ and $\varphi$ as they hold in $\sigma^1$ (and no others)
  - extend $A_2$ with $\neg a$, $\neg a \land b$ and $\varphi$ as they hold in $\sigma^2$ (and no others)
  - $\ldots$ and so forth
  - this is not effective and is performed on the automaton (not on words)

- Result:
  - $\bar{\sigma} = \underbrace{\{ a, \neg b, \neg(\neg a \land b), \varphi \}}_{B_0} \underbrace{\{ a, b, \neg(\neg a \land b), \varphi \}}_{B_1} \underbrace{\{ \neg a, b, \neg a \land b, \varphi \}}_{B_2} \ldots$
Closure

For LTL-formula $\varphi$, the set $\text{closure}(\varphi)$ consists of all sub-formulas $\psi$ of $\varphi$ and their negation $\neg\psi$

(where $\psi$ and $\neg\neg\psi$ are identified)

for $\varphi = a \cup (\neg a \land b)$, $\text{closure}(\varphi) = \{ a, b, \neg a, \neg b, \neg a \land b, \neg(\neg a \land b), \varphi, \neg \varphi \}$

can we take $B_i$ as any subset of $\text{closure}(\varphi)$? no! they must be elementary
Elementary sets of formulae

$B \subseteq \text{closure}(\varphi)$ is elementary if:

1. $B$ is logically consistent if for all $\varphi_1 \land \varphi_2, \psi \in \text{closure}(\varphi)$:
   - $\varphi_1 \land \varphi_2 \in B \iff \varphi_1 \in B$ and $\varphi_2 \in B$
   - $\psi \in B \Rightarrow \neg \psi \notin B$
   - $\text{true} \in \text{closure}(\varphi) \Rightarrow \text{true} \in B$

2. $B$ is locally consistent if for all $\varphi_1 \cup \varphi_2 \in \text{closure}(\varphi)$:
   - $\varphi_2 \in B \Rightarrow \varphi_1 \cup \varphi_2 \in B$
   - $\varphi_1 \cup \varphi_2 \in B$ and $\varphi_2 \notin B \Rightarrow \varphi_1 \in B$

3. $B$ is maximal, i.e., for all $\psi \in \text{closure}(\varphi)$:
   - $\psi \notin B \Rightarrow \neg \psi \in B$
Examples
The GNBA of LTL-formula $\varphi$

For LTL-formula $\varphi$, let $G_\varphi = (Q, 2^{AP}, \delta, Q_0, \mathcal{F})$ where

- $Q =$ all elementary sets $B \subseteq \text{closure}(\varphi)$, $Q_0 = \{B \in Q \mid \varphi \in B\}$
- $\mathcal{F} = \{\{B \in Q \mid \varphi_1 \cup \varphi_2 \notin B \text{ or } \varphi_2 \in B\} \mid \varphi_1 \cup \varphi_2 \in \text{closure}(\varphi)\}$
- The transition relation $\delta : Q \times 2^{AP} \to 2^Q$ is given by:
  
  - If $A \neq B \cap AP$ then $\delta(B, A) = \emptyset$
  - $\delta(B, B \cap AP)$ is the set $B'$ of all elementary sets of formulas satisfying:
    (i) For every $\bigcirc \psi \in \text{closure}(\varphi)$: $\bigcirc \psi \in B \iff \psi \in B'$, and
    (ii) For every $\varphi_1 \cup \varphi_2 \in \text{closure}(\varphi)$:
      $$\varphi_1 \cup \varphi_2 \in B \iff (\varphi_2 \in B \lor (\varphi_1 \in B \land \varphi_1 \cup \varphi_2 \in B'))$$
GNBA for LTL-formula $\bigcirc a$

\begin{itemize}
  \item $Q_0 = \{ B_1, B_3 \}$ since $\bigcirc a \in B_1$ and $\bigcirc a \in B_3$
  \item $\delta(B_2, \{ a \}) = \{ B_3, B_4 \}$ as $B_2 \cap \{ a \} = \{ a \}$, $\neg \bigcirc a = \bigcirc \neg a \in B_2$, and $\neg a \in B_3, B_4$
  \item $\delta(B_1, \{ a \}) = \{ B_1, B_2 \}$ as $B_1 \cap \{ a \} = \{ a \}$, $\bigcirc a \in B_1$ and $a \in B_1, B_2$
  \item $\delta(B_4, \{ a \}) = \emptyset$ since $B_4 \cap \{ a \} = \emptyset \neq \{ a \}$
\end{itemize}

The set $\mathcal{F}$ is empty, since $\varphi = \bigcirc a$ does not contain an until-operator.
GNBA for LTL-formula $a \cup b$

justification: on the black board
NBA are more expressive than LTL

Corollary: every LTL-formula expresses an $\omega$-regular property

But: there exist $\omega$-regular properties that cannot be expressed in LTL

Example: there is no LTL formula $\varphi$ with $\text{Words}(\varphi) = P$ for the LT-property:

$$ P = \left\{ A_0 A_1 A_2 \ldots \in \left( 2^{\{ a \}} \right)^\omega \mid a \in A_{2i} \text{ for } i \geq 0 \right\} $$

But there exists an NBA $\mathcal{A}$ with $\mathcal{L}_\omega(\mathcal{A}) = P$

$\Rightarrow$ there are $\omega$-regular properties that cannot be expressed in LTL!
For any LTL-formula $\varphi$ (over $AP$) there exists an NBA $A_\varphi$ with $\text{Words}(\varphi) = \mathcal{L}_\omega(A_\varphi)$ and which can be constructed in time and space in $2^{\mathcal{O}(|\varphi| \cdot \log |\varphi|)}$
Time and space complexity

• States GNBA $G_\varphi$ are elementary sets of formulae in $closure(\varphi)$
  
  – sets $B$ can be represented by bit vectors with single bit per subformula $\psi$ of $\varphi$

• The number of states in $G_\varphi$ is bounded by $2^{|subf(\varphi)|}$
  
  – where $subf(\varphi)$ denotes the set of all subformulae of $\varphi$
  – $|subf(\varphi)| \leq 2 \cdot |\varphi|$; so, the number of states in $G_\varphi$ is bounded by $2^{O(|\varphi|)}$

• The number of accepting sets of $G_\varphi$ is bounded above by $O(|\varphi|)$

• The number of states in NBA $A_\varphi$ is thus bounded by $2^{O(|\varphi|)} \cdot O(|\varphi|)$

• $2^{O(|\varphi|)} \cdot O(|\varphi|) = 2^{O(|\varphi| \log |\varphi|)}$  \hspace{1cm} qed
There exists a family of LTL formulas $\varphi_n$ with $|\varphi_n| = O(poly(n))$ such that every NBA $A_{\varphi_n}$ for $\varphi_n$ has at least $2^n$ states.
Proof (1)

Let $AP$ be non-empty, that is, $|2^{AP}| \geq 2$ and:

$$\mathcal{L}_n = \left\{ A_1 \ldots A_n A_1 \ldots A_n \sigma \mid A_i \subseteq AP \land \sigma \in \left( 2^{AP} \right)^\omega \right\}, \quad \text{for } n \geq 0$$

It follows $\mathcal{L}_n = \text{Words}(\varphi_n)$ where

$$\varphi_n = \bigwedge_{a \in AP} \bigwedge_{0 \leq i < n} (\bigcirc^i a \leftrightarrow \bigcirc^{n+i} a)$$

$\varphi_n$ is an LTL formula of polynomial length: $|\varphi_n| \in \mathcal{O}\left(|AP| \cdot n\right)$

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 $\bigwedge_{a \in AP} \bigwedge_{0 \leq i < n} (\circ^i a \leftrightarrow \circ^{n+i} a)$ has at least $2^n$ states.

Words of the form $A_1 \ldots A_n A_1 \ldots A_n \varnothing \varnothing \varnothing \ldots$ are accepted by $\mathcal{A}$.

$\mathcal{A}$ thus has for every word $A_1 \ldots A_n$ of length $n$, a state $q(A_1 \ldots A_n)$, say, which can be reached from an initial state by consuming $A_1 \ldots A_n$.

From $q(A_1 \ldots A_n)$, it is possible to visit an accept state infinitely often by accepting the suffix $A_1 \ldots A_n \varnothing \varnothing \varnothing \ldots$.

If $A_1 \ldots A_n \neq A_1' \ldots A_n'$ then

$$A_1 \ldots A_n A_1' \ldots A_n' \varnothing \varnothing \varnothing \ldots \notin \mathcal{L}_n = \mathcal{L}_\omega(\mathcal{A})$$

Therefore, the states $q(A_1 \ldots A_n)$ are all pairwise different.

Given $|2^{AP}|$ possible sequences $A_1 \ldots A_n$, NBA $\mathcal{A}$ has $\geq \left( |2^{AP}| \right)^n \geq 2^n$ states.
Complexity for LTL model checking

The time and space complexity of LTL model checking is in $\mathcal{O}(\mid TS\mid \cdot 2^{|\phi|})$.
The LTL model-checking problem is PSPACE-complete