# 3. Linear Temporal Logic

Huixing Fang

School of Information Engineering
Yangzhou University

# Outline

## Definition 1 (Syntax of LTL)

$$\varphi ::= \mathtt{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2$$

where $a \in AP$. Precedence order: $\neg = \bigcirc \ > \mathbf{U} > \ \wedge$.

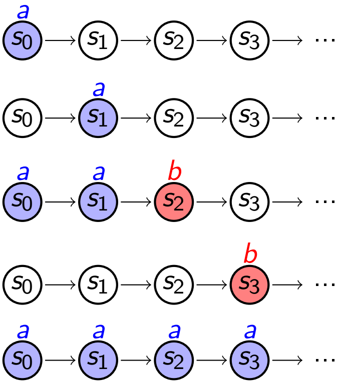
atomic prop. $a$

next operator $\bigcirc a$

until operator $a \mathbf{U} b$

eventually $\Diamond b$

always $\Box a$




The until operator allows to derive modalities:

1. $\Diamond$("eventually")
   $\Diamond\varphi = \mathtt{true}\mathbf{U}\varphi$
2. $\Box$ ("always")
   $\Box\varphi = \neg\Diamond\neg\varphi$

# 1 Syntax

## Example 2

| | | |
|---|---|---|
| mutual exclusion: | $\Box(\neg crit_1 \vee \neg crit_2)$ | (1) |
| railroad-crossing: | $\Box(train\_is\_near \rightarrow gate\_is\_closed)$ | (2) |
| progress property: | $\Box(request \rightarrow \Diamond response)$ | (3) |
| traffic light: | $\Box(yellow \vee \bigcirc\neg red)$ | (4) |
| infinitely often: | $\Box\Diamond\varphi$ | (5) |
| eventually forever: | $\Diamond\Box\varphi$ | (6) |
| unconditional fairness: | $\Box\Diamond crit_i$ | (7) |
| strong fairness: | $\Box\Diamond wait_i \rightarrow \Box\Diamond crit_i$ | (8) |
| weak fairness: | $\Diamond\Box wait_i \rightarrow \Box\Diamond crit_i$ | (9) |

# Outline

## Definition 3 (Semantics of LTL over Infinite Words)

The satisfaction relation between interpretation $\sigma = A_0 A_1 A_2 ... \in (2^{AP})^\omega$ and LTL formula is defined as follows:

1. $\sigma \models \texttt{true}$
2. $\sigma \models a$ iff $A_0 \models a$, i.e., $a \in A_0$
3. $\sigma \models \varphi_1 \wedge \varphi_2$ iff $\sigma \models \varphi_1$ and $\sigma \models \varphi_2$
4. $\sigma \models \neg\varphi$ iff $\sigma \not\models \varphi$
5. $\sigma \models \bigcirc\varphi$ iff $\mathit{suffix}(\sigma, 1) = A_1 A_2 A_3 ... \models \varphi$
6. $\sigma \models \varphi_1 \mathbf{U} \varphi_2$ iff there exists $j \geq 0$ such that
   $\mathit{suffix}(\sigma, j) = A_j A_{j+1} A_{j+2} ... \models \varphi_2$ and
   $\mathit{suffix}(\sigma, i) = A_j A_{j+1} A_{j+2} ... \models \varphi_1$, for $0 \leq i < j$

**LT property** of LTL formula $\varphi$: $\mathit{Words}(\varphi) = \{\sigma \in (2^{AP})^\omega \mid \sigma \models \varphi\}$.

# 2 Semantics

## Review of execution, paths and traces

For transition system $TS$ with labeling function $L : S \to 2^{AP}$,

1. execution: states + actions, $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$
2. paths: sequences of states, $\pi = s_0\ s_1\ s_2\ \ldots$
3. traces: sequences of sets of atomic propositions

$$trace(\pi) = L(s_0)L(s_1)L(s_2)\ldots \in (2^{AP})^{\omega}$$

## Semantics of $\Diamond$ and $\Box$ over Infinite Words

For $\sigma = A_0A_1A_2\ldots \in (2^{AP})^{\omega}$, and LTL formula $\varphi$

1. $\sigma \models \Diamond\varphi$ iff there exists $j \geq 0$ such that $A_jA_{j+1}A_{j+2}\ldots \models \varphi$
2. $\sigma \models \Box\varphi$ iff for all $j \geq 0$ we have $A_jA_{j+1}A_{j+2}\ldots \models \varphi$

## Definition 4 (Semantics of LTL over Paths and States)

Let $TS = (S, Act, \rightarrow, S_0, AP, L)$ without terminal states, and let $\varphi$ be an LTL formula over $AP$.

1. For infinite path fragment $\pi$ of $TS$, the $\models$ relation is defined by

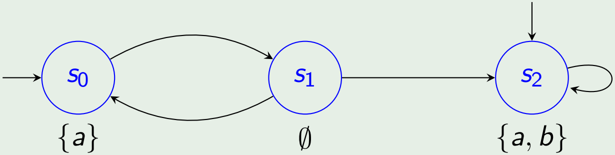
$$\begin{aligned} \pi = s_0 s_1 s_2 \ldots \models \varphi \quad & \text{iff} \quad trace(\pi) \models \varphi \\ & \text{iff} \quad trace(\pi) \in Words(\varphi) = \{\sigma \in (2^{AP})^\omega \mid \sigma \models \varphi\} \end{aligned}$$

2. For state $s \in S$, the $\models$ relation is defined by

$$\begin{aligned} s \models \varphi \quad & \text{iff} \quad \forall \pi \in Paths(s).\ \pi \models \varphi \\ & \text{iff} \quad s \models Words(\varphi) \\ & \text{iff} \quad Traces(s) \subseteq Words(\varphi) \end{aligned}$$

- $Paths(s)$ = set of all maximal path fragments starting in state $s$
- $Traces(s) = \{trace(\pi) \mid \pi \in Paths(s)\}$

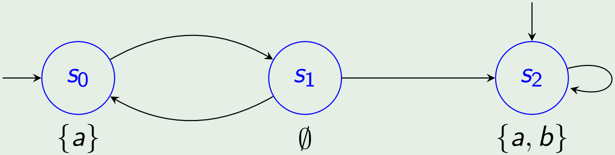
## Example 5 (LTL-semantics over paths)




- $AP = \{a, b\}$
- $\pi = s_0 s_1 s_2 s_2 \ldots$
- $trace(\pi)$
  $= L(s_0)L(s_1)L(s_2)\ldots$
  $= \{a\}\emptyset\{a, b\}^\omega$

- $\pi \models a,\ \pi \not\models b$
- $\pi \models \bigcirc(\neg a \wedge \neg b)$
- $\pi \models \bigcirc\bigcirc(a \wedge b)$
- $\pi \models (\neg b)\mathbf{U}(a \wedge b)$

## Example 6 (LTL-semantics over paths)




- $AP = \{a, b\}$
- $\pi = s_0 s_1 s_0 s_1 \ldots$
- $trace(\pi)$
  $= L(s_0)L(s_1)L(s_0)L(s_1)\ldots$
  $= \{a\}\emptyset\{a\}\emptyset$

- $\pi \models a \mathbf{U}\, b$?
- $\pi \models \Diamond b \rightarrow (a \mathbf{U}\, b)$?
- $\pi \models \bigcirc\bigcirc\neg b$?
- $\pi \models \Box a$?
- $\pi \models \Box\Diamond a$?
- $\pi \models \Diamond\Box a$?

**Definition 7 (Interpretation of LTL formulas over TS)**

Let $TS = (S, Act, \to, S_0, AP, L)$ without terminal states, and let $\varphi$ be an LTL formula over $AP$.

$$
\begin{aligned}
TS \models \varphi \text{ iff } & s_0 \models \varphi \text{ for all } s_0 \in S_0 \\
\text{iff } & trace(\pi) \models \varphi \text{ for all } \pi \in Paths(TS) \\
\text{iff } & Traces(TS) \subseteq Words(\varphi) \\
\text{iff } & TS \models Words(\varphi)
\end{aligned}
$$

**Review-1:** An LT property over $AP$ is a language $E$ of infinite words over the alphabet $\Sigma = 2^{AP}$, i.e., $E \subseteq (2^{AP})^{\omega}$.
**Review-2:** Satisfaction relation $\models$ for $TS$ and LT property $E$, $TS \models E$ iff $Traces(TS) \subseteq E$.

# Outline

# 3 Specifying Properties

LTL-formulas for MUTEX protocols, $AP = \{wait_1, crit_1, wait_2, crit_2\}$

1. the mutual exclusion property

$$\varphi_m = \Box(\neg crit_1 \vee \neg crit_2)$$

2. every process enters the critical section infinitely often

$$\varphi_\ell = \Box\Diamond crit_1 \wedge \Box\Diamond crit_2$$

3. every waiting process finally enters its critical section

$$\varphi_f = \Box(wait_1 \rightarrow \Diamond crit_1) \wedge \Box(wait_2 \rightarrow \Diamond crit_2)$$

# Outline

# 4 Equivalence of LTL Formulae

**Definition 8 (Equivalence of LTL formulas)**

LTL formulae $\varphi_1$, $\varphi_2$. $\varphi_1 \equiv \varphi_2$ iff $Words(\varphi_1) = Words(\varphi_2)$ iff for all transition systems $\mathcal{T}$, $\mathcal{T} \models \varphi_1 \Leftrightarrow \mathcal{T} \models \varphi_2$.

Duality Rule: $\neg\bigcirc\varphi \equiv \bigcirc\neg\varphi$

*Proof:*

$$\begin{aligned}
& A_0A_1\ldots \models \neg\bigcirc\varphi \\
\text{iff} \quad & A_0A_1\ldots \not\models \bigcirc\varphi \\
\text{iff} \quad & A_1A_2\ldots \not\models \varphi \\
\text{iff} \quad & A_1A_2\ldots \models \neg\varphi \\
\text{iff} \quad & A_0A_1A_2\ldots \models \bigcirc\neg\varphi
\end{aligned}$$

# 4 Equivalence of LTL Formulae

The expansion laws describe the temporal modalities $\mathbf{U}$, $\Diamond$, and $\Box$ by means of a recursive equivalence.

1. until: $\boxed{\varphi \mathbf{U} \psi} \equiv \psi \vee (\varphi \wedge \bigcirc \boxed{\varphi \mathbf{U} \psi})$ least fixed point
2. eventually: $\boxed{\Diamond \psi} \equiv \psi \vee \bigcirc \boxed{\Diamond \psi}$ least fixed point
3. always: $\boxed{\Box \psi} \equiv \psi \wedge \bigcirc \boxed{\Box \psi}$ greatest fixed point

Expansion laws are fixed point equations

# 4 Equivalence of LTL Formulae

## Until is the Least Solution of the Expansion Law (Lemma 5.18)

For LTL formuae $\varphi$ and $\psi$, $Words(\varphi \mathbf{U} \psi)$ is the least LT property $P \subseteq (2^{AP})^\omega$ such that:

$$Words(\psi) \cup \{A_0A_1A_2\ldots \in Words(\varphi) \mid A_1A_2\ldots \in P\} \subseteq P \qquad (*)$$

Moreover, $Words(\varphi \mathbf{U} \psi)$ agrees with the set

$$Words(\psi) \cup \{A_0A_1A_2\ldots \in Words(\varphi) \mid A_1A_2\ldots \in Words(\varphi \mathbf{U} \psi)\}$$

The formulation "least LT property satisfying condition (*)" means that the following conditions hold:

1. $P = Words(\varphi \mathbf{U} \psi)$ satisfies $(*)$
2. $Words(\varphi \mathbf{U} \psi) \subseteq P$ for all LT properties $P$ satisfying $(*)$

# Outline

The weak until operator **W** :

$$\varphi \ \mathbf{W} \ \psi = (\varphi \mathbf{U} \psi) \vee \Box\varphi$$

Deriving "always" and "until" from "weak until":

$$\Box\varphi \equiv \varphi \ \mathbf{W} \ \mathit{false}$$
$$\varphi \mathbf{U} \psi \equiv (\varphi \ \mathbf{W} \ \psi) \wedge \Diamond\psi$$

Duality of **U** and **W** :

$$\neg(\varphi \mathbf{U} \psi) \equiv (\neg\psi) \ \mathbf{W} \ (\neg\varphi \wedge \neg\psi)$$
$$\neg(\varphi \ \mathbf{W} \ \psi) \equiv (\neg\psi) \mathbf{U} (\neg\varphi \wedge \neg\psi)$$

# 5 Weak Until, Release, and Positive Normal Form

Expansion laws for **U** and **W** :

$$\varphi \mathbf{U} \psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi \mathbf{U} \psi))$$
$$\varphi \mathbf{W} \psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi \mathbf{W} \psi))$$

Weak-Until is the Greatest Solution of the Expansion Law (Lemma 5.19)

- $Words(\varphi \mathbf{U} \psi)$ is the smallest LT-property $P$ such that

$$Words(\psi) \cup \{A_0 A_1 A_2 ... \in Words(\varphi) \mid A_1 A_2 ... \in P\} \subseteq P$$

- $Words(\varphi \mathbf{W} \psi)$ is the largest LT-property $P$ such that

$$P \subseteq Words(\psi) \cup \{A_0 A_1 A_2 ... \in Words(\varphi) \mid A_1 A_2 ... \in P\}$$

# 5 Weak Until, Release, and Positive Normal Form

## Positive Normal Form for LTL (Weak-until PNF)

For $a \in AP$, the set of LTL formulae in **weak-until positive normal form (weak-until PNF)** is given by:

$$\varphi ::= \texttt{true} \mid \texttt{false} \mid a \mid \neg a \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \bigcirc\varphi \mid \varphi_1 \mathbf{U} \varphi_2 \mid \varphi_1 \ \mathbf{W} \ \varphi_2$$

PNF also sometimes called negation normal form (NNF)

$\Diamond$ and $\Box$ can be derived:

$$\Diamond\varphi = \texttt{true}\mathbf{U}\varphi$$
$$\Box\varphi = \varphi \ \mathbf{W} \ \texttt{false}$$

Each LTL formula can be transformed into an equivalent LTL formula in PNF

Each LTL formula can be transformed into an equivalent LTL formula in PNF by using the following transformations:

$$
\begin{aligned}
\neg \texttt{true} &\rightsquigarrow \texttt{false} \\
\neg \texttt{false} &\rightsquigarrow \texttt{true} \\
\neg\neg\varphi &\rightsquigarrow \varphi \\
\neg(\varphi \wedge \psi) &\rightsquigarrow \neg\varphi \vee \neg\psi \\
\neg\bigcirc\varphi &\rightsquigarrow \bigcirc\neg\varphi \\
\neg(\varphi \mathbf{U} \psi) &\rightsquigarrow (\varphi \wedge \neg\psi)\ \mathbf{W}\ (\neg\varphi \wedge \neg\psi) \\
&\vdots
\end{aligned}
$$

# Outline

# 6 Fairness in LTL

Fairness Constraints

1. Unconditional fairness: e.g.,"Every process gets its turn infinitely often."
2. Strong fairness: e.g., "Every process that is enabled infinitely often gets its turn infinitely often."
3. Weak fairness: e.g., "Every process that is continuously enabled from a certain time instant on gets its turn infinitely often."

# 6 Fairness in LTL

## Definition 9 (Unconditional, Strong, and Weak Fairness)

For transition system $TS = (S, Act, \rightarrow, I, AP, L)$ without terminal states, $A \subseteq Act$, and infinite execution fragment $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \ldots$ of $TS$:

1. $\rho$ is unconditionally $A$-fair whenever $\overset{\infty}{\exists}\, j.\ \alpha_j \in A$.
2. $\rho$ is strongly $A$-fair whenever $(\overset{\infty}{\exists}\, j.\ Act(s_j) \cap A \neq \emptyset) \Rightarrow (\overset{\infty}{\exists}\, j.\ \alpha_j \in A)$.
3. $\rho$ is weakly $A$-fair whenever $(\overset{\infty}{\forall}\, j.\ Act(s_j) \cap A \neq \emptyset) \Rightarrow (\overset{\infty}{\exists}\, j.\ \alpha_j \in A)$.

$\overset{\infty}{\exists}\, j$ : there are infinitely many $j$.
$\overset{\infty}{\forall}\, j$ : for nearly all $j$, for all, except for finitely many $j$.
The variable $j$ ranges over the natural numbers.

For state $s$, let $Act(s)$ denote the set of actions that are executable in state $s$,

$$Act(s) = \{\alpha \in Act \mid \exists s' \in S.\ s \xrightarrow{\alpha} s'\}$$

## Definition 10 (Fairness Assumption)

A fairness assumption for *Act* is a triple

$$\mathcal{F} = (\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak})$$

with $\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak} \subseteq 2^{Act}$. Execution $\rho$ is **$\mathcal{F}$-fair** if

1. it is unconditionally $A$-fair for **all** $A \in \mathcal{F}_{ucond}$,
2. it is strongly $A$-fair for **all** $A \in \mathcal{F}_{strong}$, and
3. it is weakly $A$-fair for **all** $A \in \mathcal{F}_{weak}$.

Remark: $A$ is a set of actions.

1. $FairPaths_{\mathcal{F}}(s)$ : the set of $\mathcal{F}$-paths of $s$ (i.e., infinite $\mathcal{F}$-fair path fragments that start in state $s$).
2. $FairPaths_{\mathcal{F}}(TS)$ : set of $\mathcal{F}$-fair paths that start in some initial state of $TS$.
3. $FairTraces_{\mathcal{F}}(s) = trace(FairPaths_{\mathcal{F}}(s))$

**Definition 11 (Fair Satisfaction Relation for LT Properties)**

Let $P$ be an LT property over $AP$ and $F$ a fairness assumption over $Act$. Transition system $TS = (S, Act, \rightarrow, I, AP, L)$ fairly satisfies $P$, notation $TS \models_{\mathcal{F}} P$, iff $FairTraces_{\mathcal{F}}(TS) \subseteq P$.

In case a transition system has traces that are not $\mathcal{F}$-fair, then in general we are confronted with a situation

$$TS \models_{\mathcal{F}} P \qquad \text{whereas} \qquad TS \not\models P$$

By restricting the validity of a property to the set of fair paths, the verification can be restricted to "realistic" executions.

## Definition 12 (LTL Fairness Constraints and Assumptions)

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

1. An unconditional LTL fairness constraint is an LTL formula of the form
$$ufair = \Box\Diamond\Psi.$$
2. A strong LTL fairness condition is an LTL formula of the form
$$sfair = \Box\Diamond\Phi \rightarrow \Box\Diamond\Psi.$$
3. A weak LTL fairness constraint is an LTL formula of the form
$$wfair = \Diamond\Box\Phi \rightarrow \Box\Diamond\Psi.$$

An LTL fairness assumption is a conjunction of LTL fairness constraints (of any arbitrary type).

# 6 Fairness in LTL

Notations:

1. LTL fairness assumptions are a conjunction of unconditional, strong, and weak fairness assumptions: $\mathit{fair} = \mathit{ufair} \wedge \mathit{sfair} \wedge \mathit{wfair}$.
2. Set of all fair paths starting in $s$,
   $\mathit{FairPaths}(s) = \{\pi \in \mathit{Paths}(s) \mid \pi \models \mathit{fair}\}$
3. Set of all traces induced by fair paths starting in $s$,
   $\mathit{FairTraces}(s) = \{\mathit{trace}(\pi) \mid \pi \in \mathit{FairPaths}(s)\}$

## Definition 13 (Satisfaction Relation for LTL with Fairness)

For state $s$ in transition system $TS$ (over $AP$) without terminal states, LTL formula $\varphi$, and LTL fairness assumption *fair* let

$$\begin{aligned} s \models_{\mathit{fair}} \varphi \quad & \text{iff} \quad \forall \pi \in \mathit{FairPaths}(s).\ \pi \models \varphi \quad \text{and} \\ TS \models_{\mathit{fair}} \varphi \quad & \text{iff} \quad \forall s_0 \in I. s_0 \models_{\mathit{fair}} \varphi. \end{aligned}$$

$TS$ satisfies $\varphi$ under *fair* if $\varphi$ holds for all **fair paths** that originate from some initial state.

**Theorem 14 (Reduction of $\models_{fair}$ to $\models$)**

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

$$TS \models_{fair} \varphi \quad \text{iff} \quad TS \models fair \rightarrow \Phi.$$

# Outline

For transition system $TS$ and LTL formula $\varphi$, let $\mathcal{A}$ be an NBA (Nondeterministic Finite Automaton) with $\mathcal{L}_\omega(\mathcal{A}) = Words(\neg\varphi)$ :

$$
\begin{aligned}
TS \models \varphi \quad & \text{iff} \quad Traces(TS) \subseteq Words(\varphi) \\
& \text{iff} \quad Traces(TS) \cap ((2^{AP})\omega \setminus Words(\varphi)) = \emptyset \\
& \text{iff} \quad Traces(TS) \cap Words(\neg\varphi) = \emptyset \\
& \text{iff} \quad Traces(TS) \cap \mathcal{L}_\omega(\mathcal{A}) = \emptyset
\end{aligned}
$$

# 7 Automata-Based LTL Model Checking

**Algorithm 1:** Automaton-based LTL model checking

**Input:** finite transition system $TS$ and LTL formula $\varphi$ (both over $AP$)
**Output:** "yes" if $TS \models \varphi$; otherwise, "no" plus a counterexample

Construct an NBA $A_{\neg\varphi}$ such that $\mathcal{L}_\omega(A_{\neg\varphi}) = Words(\neg\varphi)$
Construct the product transition system $TS \otimes A$

**if** $\exists \pi \in Pahts(TS \otimes A)$ *satisfying the accepting condition of* $A$ **then**
  return "no" and an expressive prefix of $\pi$
**else**
  return "yes"
**end**

$$
\begin{array}{ll}
 & TS \models \text{LTL-formula } \varphi \\
\text{iff} & Traces(TS) \cap \mathcal{L}_\omega(\mathcal{A}) = \emptyset \\
\text{iff} & \text{there is NO path } \langle s_0, q_0 \rangle \langle s_1, q_1 \rangle \langle s_2, q_2 \rangle \ldots \\
 & \text{ in } TS \otimes A \text{ s.t. } q_i \in F \text{ for infinitely many } i \in \mathbb{N} \\
\text{iff} & TS \otimes A \models \Diamond \Box \neg F
\end{array}
$$

## Definition 15 (Nondeterministic Büchi Automaton (NBA))

A Büchi automaton (NBA) $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ where

- $Q$ is a finite set of states,
- $\Sigma$ is an alphabet,
- $\delta : Q \times \Sigma \to 2^Q$ is a transition function,
- $Q_0 \subseteq Q$ is a set of initial states, and
- $F \subseteq Q$ is a set of accept states, called the acceptance set.

- A **run** for $\sigma = A_0 A_1 A_2 ... \in \Sigma^\omega$ denotes an infinite sequence $q_0 q_1 q_2 ...$ of states in $\mathcal{A}$ such that $q_0 \in Q_0$ and $q_i \xrightarrow{A_i} q_{i+1}$ for $i \geq 0$.
- Run $q_0 q_1 q_2 ...$ is **accepting** if $q_i \in F$ for infinitely many indices $i \in \mathbb{N}$.
- The **accepted language** of $\mathcal{A}$ is
  $\mathcal{L}_\omega(\mathcal{A}) = \{\sigma \in \Sigma^\omega \mid \text{there exists an accepting run for } \sigma \text{ in } \mathcal{A}\}$

## Definition 16 (Nonblocking NBA)

Let $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ be an NBA. $\mathcal{A}$ is called **nonblocking** if $\delta(q, a) \neq \emptyset$ for all states $q$ and all symbols $a \in \Sigma$.

**Remark:** For each NBA $\mathcal{A}$ there exists a nonblocking NBA $trap(\mathcal{A})$ with $|trap(\mathcal{A})| = O(|\mathcal{A}|)$ and $\mathcal{A} \equiv trap(\mathcal{A})$.

In nonblocking NBA $trap(\mathcal{A})$,

$$\delta'(q, A) = \begin{cases} \delta(q, A) & \text{if } q \in Q \text{ and } \delta(q, A) \neq \emptyset \\ \{q_{trap}\} & \text{otherwise} \end{cases}$$

$trap(\mathcal{A})$ is obtained from $\mathcal{A}$ by inserting a nonaccept trapping state $q_{trap}$ equipped with a self-loop for each symbol of $\Sigma$.

# 7 Automata-Based LTL Model Checking

## Definition 17 (Product of Transition System and NBA)

Let $TS = (S, Act, \rightarrow, I, AP, L)$ be a transition system without terminal states and $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ a nonblocking NBA. Then, $TS \otimes \mathcal{A}$ is the following transition system:

$$TS \otimes \mathcal{A} = (S \times Q, Act, \rightarrow', I', AP', L')$$

where $\rightarrow'$ is the smallest relation defined by the rule

$$\frac{s \xrightarrow{\alpha} t \wedge p \xrightarrow{L(t)} q}{\langle s, p \rangle \xrightarrow{\alpha}{}' \langle t, q \rangle}$$

and

- $I' = \{\langle s_0, q \rangle \mid s_o \in I \wedge \exists q_0 \in Q_0.\ q_0 \xrightarrow{L(s_0)} q\}$,
- $AP' = Q$ and $L' : S \times Q \rightarrow 2^Q$ is given by $L'(\langle s, q \rangle) = \{q\}$.

# 7.1 From LTL to NBA

Construction of an NBA $\mathcal{A}_\varphi$ satisfying $\mathcal{L}_\omega(\mathcal{A}_\varphi) = \mathit{Words}(\varphi)$ for the LTL formula $\varphi$

$$\text{LTL formula } \varphi$$
$$\downarrow$$
$$\text{Generalised Nondeterministic Büchi automaton } G_\varphi$$
$$\downarrow$$
$$\text{Nondeterministic Büchi Automaton } \mathcal{A}_\varphi$$

## Definition 18 (Generalized NBA (GNBA))

A generalized NBA is a tuple $G = (Q, \Sigma, \delta, Q_0, \mathcal{F})$ where $Q, \Sigma, \delta, Q_0$ are defined as for an NBA, and $\mathcal{F}$ is a (possibly empty) subset of $2^Q$.

The elements $F \in \mathcal{F}$ are called **acceptance sets**. The infinite run $q_0 q_1 q_2 \ldots \in Q^\omega$ is called **accepting** if

$$\forall F \in \mathcal{F}.\ (\overset{\infty}{\exists}\ j \in \mathbb{N}.\ q_j \in F)\ .$$

The accepted language of $G$ is:

$$\mathcal{L}_\omega(G) = \{\sigma \in \Sigma^\omega \mid \text{there exists an accepting run for } \sigma \text{ in } G\}$$

**Review:** infinite words can be defined as functions $\sigma : \mathbb{N} \to \Sigma$ and the notation $\sigma = A_1 A_2 A_3 \ldots$ means that $\sigma(i) = A_i$ for all $i \in \mathbb{N}$.

Let infinite word $\sigma = A_0 A_1 A_2 ... \in Words(\varphi)$, an infinite words $\bar{\sigma} = B_0 B_1 B_2 ...$ satisfies

$$\psi \in B_i \quad \text{iff} \quad A_i A_{i+1} A_{i+2} ... \models \psi \qquad (\psi \in closure(\varphi))$$

## Example 19

If $\varphi = a\mathbf{U}(\neg \wedge b)$, $\sigma = \{a\}\{a, b\}\{b\}...$ then

$$\begin{aligned} closure(\varphi) &= \{a, b, \neg a, \neg b, \neg a \wedge b, \neg(\neg a \wedge b), \varphi, \neg\varphi\} \\ B_0 &= \{a, \neg b, \neg(\neg a \wedge b), \varphi\} \\ B_1 &= \{a, b, \neg(\neg a \wedge b), \varphi\} \\ B_2 &= \{\neg a, b, \neg a \wedge b, \varphi\} \\ &... \end{aligned}$$

The GNBA $G_\varphi$ is constructed such that the sets $Bi$ constitute its states

### Definition 20 (Closure of $\varphi$)

The closure of LTL formula $\varphi$ is the set *closure*$(\varphi)$ consisting of all subformulae $\psi$ of $\varphi$ and their negation $\neg\psi$ (where $\psi$ and $\neg\neg\psi$ are identified).

### Definition 21 (Elementary Sets of Formulae)

$B \subseteq$ *closure*$(\varphi)$ is elementary if it is consistent with respect to propositional logic, maximal, and locally consistent with respect to the until operator.

## Properties of elementary sets of formulae

1. $B$ is **consistent** with respect to propositional logic, for all $\varphi_1 \wedge \varphi_2, \psi \in closure(\varphi)$:
   - $\varphi_1 \wedge \varphi_2 \in B \Leftrightarrow \varphi_1 \in B$ and $\varphi_2 \in B$
   - $\psi \in B \Rightarrow \neg\psi \notin B$
   - $\mathtt{true} \in closure(\varphi) \Rightarrow \mathtt{true} \in B$.
2. $B$ is **locally consistent** with respect to the until operator, for all $\varphi_1 \mathbf{U} \varphi_2 \in closure(\varphi)$:
   - $\varphi_2 \in B \Rightarrow \varphi_1 \mathbf{U} \varphi_2 \in B$
   - $\varphi_1 \mathbf{U} \varphi_2 \in B$ and $\varphi_2 \notin B \Rightarrow \varphi_1 \in B$.
3. $B$ is **maximal**, for all $\psi \in closure(\varphi)$:
   - $\psi \notin B \Rightarrow \neg\psi \in B$.

# 7.1 From LTL to NBA

**Theorem 22 (GNBA for LTL Formula)**

*For any LTL formula $\varphi$ (over $AP$) there exists a GNBA $G_\varphi$ over the alphabet $2^AP$ such that*

1. $Words(\varphi) = \mathcal{L}_\omega(G_\varphi)$.
2. $G_\varphi$ *can be constructed in time and space* $2^{O(|\varphi|)}$.
3. *The number of accepting sets of* $G_\varphi$ *is bounded above by* $O(|\varphi|)$.

# 7.1 From LTL to NBA

Let $\varphi$ be an LTL formula over $AP$. Let $G_\varphi = (Q, 2^{AP}, \delta, Q_0, \mathcal{F})$ :

- $Q$ : set of all elementary sets of formulae $B \subseteq \mathit{closure}(\varphi)$,
- $Q_0 = \{B \in Q \mid \varphi \in B\}$,
- $\mathcal{F} = \{F_{\varphi_1 \mathbf{U} \varphi_2} \mid \varphi_1 \mathbf{U} \varphi_2 \in \mathit{closure}(\varphi)\}$, where $F_{\varphi_1 \mathbf{U} \varphi_2} = \{B \in Q \mid \varphi_1 \mathbf{U} \varphi_2 \notin B \text{ or } \varphi_2 \in B\}$.

The transition relation $\delta : Q \times 2^{AP} \rightarrow 2^Q$ is given by:

- If $A \neq B \cap AP$, then $\delta(B, A) = \emptyset$
- If $A = B \cap AP$, then $\delta(B, A)$ is the set of all elementary sets of formulae $B'$ satisfying
  1. for every $\bigcirc\psi \in \mathit{closure}(\varphi)$: $\bigcirc \in B \Leftrightarrow \psi \in B'$, and
  2. for every $\varphi_1 \mathbf{U} \varphi_2 \in \mathit{closure}(\varphi)$: $\varphi_1 \mathbf{U} \varphi_2 \in B \Leftrightarrow (\varphi_2 \in B \vee (\varphi_1 \in B \wedge \varphi_1 \mathbf{U} \varphi_2 \in B'))$.

**Theorem 23**

*From each GNBA $G$ there exists an NBA $\mathcal{A}$ with $\mathcal{L}_\omega(G) = \mathcal{L}_\omega(\mathcal{A})$.*

*Proof.* Let $G = (Q, \Sigma, \delta, Q_0, \mathcal{F})$, with $\mathcal{F} = \{F_1, ..., F_k\}$ and

1. If $k = 1$ then $G$ is an NBA
2. If $k \geq 2$ then NBA $\mathcal{A}$ results from $k$ copies of $G$: $\mathcal{A} = (Q', \Sigma, \delta', Q'_0, F')$ where:
   - $Q' = Q \times \{1, ..., k\}$,
   - $Q'_0 = Q_0 \times \{1\} = \{\langle q_0, 1\rangle \mid q_0 \in Q_0\}$, and
   - $F' = F_1 \times \{1\} = \{\langle q_F, 1\rangle \mid q_F \in F_1\}$.

   The transition function $\delta'$ :

$$\delta'(\langle q, i\rangle, A) = \begin{cases} \{\langle q', i\rangle \mid q' \in \delta(q, A)\} & \text{if } q \notin F_i \\ \{\langle q', i+1\rangle \mid q' \in \delta(q, A)\} & \text{otherwise } (i = k \text{ back}) \end{cases}$$

$$size(\mathcal{A}) = O(Size(G) \cdot |\mathcal{F}|)$$

# 7.2 Complexity of LTL to NBA

For each LTL formula $\varphi$, there is an NBA $\mathcal{A}$ s.t. $\mathcal{L}_\omega(\mathcal{A}) = \mathit{Words}(\varphi)$ and $\mathit{size}(\mathcal{A}) \leq 2^{cl(\varphi)} \cdot |\varphi| = 2^{O|\varphi|}$

1. From LTL formula $\varphi$ to GNBA $G$ of size $2^{cl(\varphi)}$
2. From GNBA $G$ to NBA $\mathcal{A}$ of size $\mathit{size}(G) \cdot |\mathcal{F}|$
3. $|\mathcal{F}|$: number of acceptance sets in $G$, $|\mathcal{F}| \leq |\varphi|$

# 7.3 Complexity of LTL Model Checking

**Theorem 24**

*The LTL model-checking problem is PSPACE-complete (PSPACE and PSPACE-hard).*

1. **PTIME** (or briefly P) denotes the class of all decision problems that can be solved by a deterministic polytime algorithm
2. **NP** denotes the class of all decision problems that can be solved by a nondeterministic polytime algorithm.
3. **PSPACE** denotes the class of all decision problems that can be solved by a deterministic polyspace algorithm.
4. Decision problem $P$ is **PSPACE-hard** if all problems in PSPACE are polynomially reducible to $P$.

# Conclusion

1. LTL is a logic for formalizing path-based properties
2. LTL formulae can be transformed algorithmically into nondeterministic Büchi automata (NBA). This transformation can cause an exponential blowup.
3. The LTL model-checking problem can be solved by a nested depth-first search in the product of the given transition system and an NBA for the negated formula.
4. The time complexity of the automata-based model-checking algorithm for LTL is linear in the size of the transition system and exponential in the length of the formula