4507/6507 Software and Hardware Verification
LTL Model Checking

Andrei Popescu

University of Sheffield

These slides contain material from Denisa Diaconescu, Georg Struth and Traian Florin Șerbănuță

Overview

The LTL model checking problem

Overview

The LTL model checking problem

The model checking algorithm in three steps

Overview

The LTL model checking problem

The model checking algorithm in three steps

Generalized Nondeterministic Büchi Automata (GNBA)

- Translation of LTL formulas to automata
- Product automata
- Emptiness decision problem

Overview

The LTL model checking problem

The model checking algorithm in three steps

Generalized Nondeterministic Büchi Automata (GNBA)

- Translation of LTL formulas to automata
- Product automata
- Emptiness decision problem

We will see

- not only what needs to be done
- but also why it works – and we will give proofs for that

Introduction

Preliminaries: Atom-Set Traces

Recall: For a set of states S, a labeling function $L : S \to \mathcal{P}(Atoms)$, an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ and a formula φ, we know what $\pi \models_L \varphi$ (π satisfies φ w.r.t. labeling L) means.

Preliminaries: Atom-Set Traces

Recall: For a set of states S, a labeling function $L : S \to \mathcal{P}(Atoms)$, an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ and a formula φ, we know what $\pi \models_L \varphi$ (π satisfies φ w.r.t. labeling L) means.

Important observation: This concept does not depend on the exact S and L, but only on the image of L through S.

Preliminaries: Atom-Set Traces

Recall: For a set of states S, a labeling function $L : S \to \mathcal{P}(Atoms)$, an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ and a formula φ, we know what $\pi \models_L \varphi$ (π satisfies φ w.r.t. labeling L) means.

Important observation: This concept does not depend on the exact S and L, but only on the image of L through S. More precisely, given:

- S, L and $\pi = s_0 s_1 s_2 \ldots$ infinite sequence of states in S
- S', L' and $\pi' = s'_0 s'_1 s'_2 \ldots$ infinite sequence of states in S'

then, assuming $L(s_i) = L'(s'_i)$ for all $i \geq 0$, we have

$$\pi \models_L \varphi \text{ iff } \pi' \models_{L'} \varphi$$

Preliminaries: Atom-Set Traces

Recall: For a set of states S, a labeling function $L : S \to \mathcal{P}(Atoms)$, an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ and a formula φ, we know what $\pi \models_L \varphi$ (π satisfies φ w.r.t. labeling L) means.

Important observation: This concept does not depend on the exact S and L, but only on the image of L through S. More precisely, given:

- S, L and $\pi = s_0 s_1 s_2 \ldots$ infinite sequence of states in S
- S', L' and $\pi' = s'_0 s'_1 s'_2 \ldots$ infinite sequence of states in S'

then, assuming $L(s_i) = L'(s'_i)$ for all $i \geq 0$, we have

$$\pi \models_L \varphi \;\text{ iff }\; \pi' \models_{L'} \varphi$$

In other words, it is only the infinite sequence of atom sets $L(s_0)L(s_1)L(s_2)\ldots$ that matters – we call this the atom-set trace of $\pi = s_0 s_1 s_2 \ldots$ through L.

Preliminaries: Atom-Set Traces

Recall: For a set of states S, a labeling function $L : S \to \mathcal{P}(Atoms)$, an infinite sequence of states $\pi = s_0 s_1 s_2 \ldots$ and a formula φ, we know what $\pi \models_L \varphi$ (π satisfies φ w.r.t. labeling L) means.

Important observation: This concept does not depend on the exact S and L, but only on the image of L through S. More precisely, given:

- S, L and $\pi = s_0 s_1 s_2 \ldots$ infinite sequence of states in S
- S', L' and $\pi' = s'_0 s'_1 s'_2 \ldots$ infinite sequence of states in S'

then, assuming $L(s_i) = L'(s'_i)$ for all $i \geq 0$, we have

$$\pi \models_L \varphi \;\text{ iff }\; \pi' \models_{L'} \varphi$$

In other words, it is only the infinite sequence of atom sets $L(s_0)L(s_1)L(s_2)\ldots$ that matters – we call this the atom-set trace of $\pi = s_0 s_1 s_2 \ldots$ through L.

Example: If $L(s_0) = \{a, b\}$, $L(s_1) = \{a\}$ and $L(s_2) = \{b\}$, then the atom-set trace of $s_0 s_2 s_1 s_2 s_0 \ldots$ is $\{a, b\}\,\{b\}\,\{a\}\,\{b\}\,\{a, b\}\ldots$

The LTL Model Checking Problem

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS, $s_0 \in S$, and φ an LTL formula.

The LTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.

The LTL Model Checking Problem

Let $\mathcal{M} = (S, \to, L)$ be an LTS, $s_0 \in S$, and φ an LTL formula.

The LTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.
(Remember that this means: $\pi \models_L \varphi$ for all $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$, where we write $\mathit{Paths}_{s_0}(\mathcal{M})$ for the set of paths of $\mathcal{M}$ that start in s_0.)

The LTL Model Checking Problem

Let $\mathcal{M} = (S, \to, L)$ be an LTS, $s_0 \in S$, and φ an LTL formula.

The LTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.
(Remember that this means: $\pi \models_L \varphi$ for all $\pi \in Paths_{s_0}(\mathcal{M})$, where we write $Paths_{s_0}(\mathcal{M})$ for the set of paths of $\mathcal{M}$ that start in s_0.)

We will see that there is an algorithmic solution to this problem.

The LTL Model Checking Problem

Let $\mathcal{M} = (S, \to, L)$ be an LTS, $s_0 \in S$, and φ an LTL formula.

The LTL model checking problem is to determine whether $\mathcal{M}, s_0 \models \varphi$, i.e., whether $\mathcal{M}$ satisfies φ in state s_0.
(Remember that this means: $\pi \models_L \varphi$ for all $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$, where we write $\mathit{Paths}_{s_0}(\mathcal{M})$ for the set of paths of $\mathcal{M}$ that start in s_0.)

We will see that there is an algorithmic solution to this problem.

In-class exercise. Please discuss the flaws of the following argument: It is obvious that the LTL model checking problem has an algorithmic solution, because both the LTS and the formula are finite objects, so whether the LTS satisfies a formula in a given state should be decidable by simply applying the definition of satisfaction and doing an exhaustive check through the finite set of states.

The Model Checking Algorithm – High Level

The algorithm has three steps:

Step 1. Construct an automaton for the formula $\neg\varphi$.

The Model Checking Algorithm – High Level

The algorithm has three steps:

Step 1. Construct an automaton for the formula $\neg\varphi$.

- We write $\mathcal{A}ut_\psi$ for the automaton of a formula ψ. Thus we construct $\mathcal{A}ut_{\neg\varphi}$.

The Model Checking Algorithm – High Level

The algorithm has three steps:

Step 1. Construct an automaton for the formula $\neg\varphi$.

- We write $\mathcal{A}ut_\psi$ for the automaton of a formula ψ. Thus we construct $\mathcal{A}ut_{\neg\varphi}$.
- The automaton has a notion of acceped word, where a word will be an infinite sequence $A_0 A_1 A_2 \ldots$ of atom sets: for all $i \geq 0$, $A_i \in \mathcal{P}(Atoms)$. The set of its accepted words forms its accepted language.

The Model Checking Algorithm – High Level

The algorithm has three steps:

Step 1. Construct an automaton for the formula $\neg\varphi$.

- We write $\mathcal{A}ut_\psi$ for the automaton of a formula ψ. Thus we construct $\mathcal{A}ut_{\neg\varphi}$.
- The automaton has a notion of acceped word, where a word will be an infinite sequence $A_0 A_1 A_2 \ldots$ of atom sets: for all $i \geq 0$, $A_i \in \mathcal{P}(Atoms)$. The set of its accepted words forms its accepted language.
- It has the property that for any formula ψ, set of states S, labeling function $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π,

  $\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$

  ($\mathcal{A}ut_\psi$ accepts precisely the atom-set traces of sequences that satisfy ψ)

The Model Checking Algorithm – High Level

The algorithm has three steps:

Step 1. Construct an automaton for the formula $\neg\varphi$.

- We write $\mathcal{A}ut_{\psi}$ for the automaton of a formula ψ. Thus we construct $\mathcal{A}ut_{\neg\varphi}$.
- The automaton has a notion of acceped word, where a word will be an infinite sequence $A_0 A_1 A_2 \ldots$ of atom sets: for all $i \geq 0$, $A_i \in \mathcal{P}(Atoms)$. The set of its accepted words forms its accepted language.
- It has the property that for any formula ψ, set of states S, labeling function $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π,

  $\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_{\psi}$

  ($\mathcal{A}ut_{\psi}$ accepts precisely the atom-set traces of sequences that satisfy ψ)
- Thus, $\mathcal{A}ut_{\neg\varphi}$ accepts precisely the atom-set traces of sequences that do not satisfy φ.

The Model Checking Algorithm – High Level

Step 2. Combine the LTS $\mathcal{M} = (S, \to, L)$ and its target state s_0 with the automaton $\mathcal{A}ut_{\neg\varphi}$.

The Model Checking Algorithm – High Level

Step 2. Combine the LTS $\mathcal{M} = (S, \rightarrow, L)$ and its target state s_0 with the automaton $\mathcal{A}ut_{\neg\varphi}$.

This results in a product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ whose accepted words are those coming from the paths π of $\mathcal{M}$ that start in s_0 and do not satisfy φ:

$\pi \in Paths_{s_0}(\mathcal{M})$ and $\pi \not\models_L \varphi$

iff

the atom-set trace of π through L is accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$

The Model Checking Algorithm – High Level

Step 2. Combine the LTS $\mathcal{M} = (S, \rightarrow, L)$ and its target state s_0 with the automaton $\mathcal{A}ut_{\neg\varphi}$.

This results in a product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ whose accepted words are those coming from the paths π of $\mathcal{M}$ that start in s_0 and do not satisfy φ:

$$\pi \in \mathit{Paths}_{s_0}(\mathcal{M}) \text{ and } \pi \not\models_L \varphi$$

$$\text{iff}$$

$$\text{the atom-set trace of } \pi \text{ through } L \text{ is accepted by } (\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$$

Step 3. Check whether the product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ has its accepted language empty.

The Model Checking Algorithm – High Level

Step 2. Combine the LTS $\mathcal{M} = (S, \to, L)$ and its target state s_0 with the automaton $\mathcal{A}ut_{\neg\varphi}$.

This results in a product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ whose accepted words are those coming from the paths π of $\mathcal{M}$ that start in s_0 and do not satisfy φ:

$$\pi \in Paths_{s_0}(\mathcal{M}) \text{ and } \pi \not\models_L \varphi$$

iff

the atom-set trace of π through L is accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$

Step 3. Check whether the product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ has its accepted language empty.

- If the accepted language is empty, it means that $\pi \models_L \varphi$ for all $\pi \in Paths_{s_0}(\mathcal{M})$. So we conclude Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$.

The Model Checking Algorithm – High Level

Step 2. Combine the LTS $\mathcal{M} = (S, \to, L)$ and its target state s_0 with the automaton $\mathcal{A}ut_{\neg\varphi}$.

This results in a product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ whose accepted words are those coming from the paths π of $\mathcal{M}$ that start in s_0 and do <u>not</u> satisfy φ:

$$\pi \in Paths_{s_0}(\mathcal{M}) \text{ and } \pi \not\models_L \varphi$$
$$\text{iff}$$
$$\text{the atom-set trace of } \pi \text{ through } L \text{ is accepted by } (\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$$

Step 3. Check whether the product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ has its accepted language empty.

- If the accepted language is empty, it means that $\pi \models_L \varphi$ for all $\pi \in Paths_{s_0}(\mathcal{M})$. So we conclude Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$.
- If the accepted language is non-empty, we obtain $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$. So we conclude No, it is not the case that $\mathcal{M}, s_0 \models \varphi$.

The Model Checking Algorithm – High Level

Step 2. Combine the LTS $\mathcal{M} = (S, \rightarrow, L)$ and its target state s_0 with the automaton $\mathcal{A}ut_{\neg\varphi}$.

This results in a product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ whose accepted words are those coming from the paths π of $\mathcal{M}$ that start in s_0 and do not satisfy φ:

$$\pi \in Paths_{s_0}(\mathcal{M}) \text{ and } \pi \not\models_L \varphi$$
$$\text{iff}$$
$$\text{the atom-set trace of } \pi \text{ through } L \text{ is accepted by } (\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$$

Step 3. Check whether the product automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ has its accepted language empty.

- If the accepted language is empty, it means that $\pi \models_L \varphi$ for all $\pi \in Paths_{s_0}(\mathcal{M})$. So we conclude Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$.
- If the accepted language is non-empty, we obtain $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$. So we conclude No, it is not the case that $\mathcal{M}, s_0 \models \varphi$.

The Model Checking Algorithm – Remaining Details

Step 1. Construct an automaton A_ψ for any formula ψ such that, for any set of states S, labeling $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π:
$\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$

The Model Checking Algorithm – Remaining Details

Step 1. Construct an automaton A_{ψ} for any formula ψ such that, for any set of states S, labeling $L : S \to \mathcal{P}(\mathit{Atoms})$ and infinite sequences of states π: $\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_{\psi}$

What type of automaton do we construct? How do we construct it? Why does it satisfy the required property?

The Model Checking Algorithm – Remaining Details

Step 1. Construct an automaton A_ψ for any formula ψ such that, for any set of states S, labeling $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π: $\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$

What type of automaton do we construct? How do we construct it? Why does it satisfy the required property?

Step 2. From $\mathcal{M} = (S, \to, L)$ and $\mathcal{A}ut_{\neg\varphi}$, build $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ such that, for all π: $\pi \in Paths_{s_0}(\mathcal{M})$ and $\pi \not\models_L \varphi$ iff the atom-set trace of π through L is accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$.

The Model Checking Algorithm – Remaining Details

Step 1. Construct an automaton A_ψ for any formula ψ such that, for any set of states S, labeling $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π: $\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$

What type of automaton do we construct? How do we construct it? Why does it satisfy the required property?

Step 2. From $\mathcal{M} = (S, \to, L)$ and $\mathcal{A}ut_{\neg\varphi}$, build $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ such that, for all π: $\pi \in Paths_{s_0}(\mathcal{M})$ and $\pi \not\models_L \varphi$ iff the atom-set trace of π through L is accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$.

What is exactly the product automaton? Why does it satisfy the required property?

The Model Checking Algorithm – Remaining Details

Step 1. Construct an automaton A_ψ for any formula ψ such that, for any set of states S, labeling $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π: $\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$

What type of automaton do we construct? How do we construct it? Why does it satisfy the required property?

Step 2. From $\mathcal{M} = (S, \to, L)$ and $\mathcal{A}ut_{\neg\varphi}$, build $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ such that, for all π: $\pi \in Paths_{s_0}(\mathcal{M})$ and $\pi \not\models_L \varphi$ iff the atom-set trace of π through L is accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$.

What is exactly the product automaton? Why does it satisfy the required property?

Step 3. Check whether the language accepted by the automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.
If so, conclude $\mathcal{M}, s \models \varphi$; otherwise conclude $\mathcal{M}, s \not\models \varphi$.

The Model Checking Algorithm – Remaining Details

Step 1. Construct an automaton A_ψ for any formula ψ such that, for any set of states S, labeling $L : S \to \mathcal{P}(Atoms)$ and infinite sequences of states π:
$\pi \models_L \psi$ iff the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$

What type of automaton do we construct? How do we construct it? Why does it satisfy the required property?

Step 2. From $\mathcal{M} = (S, \to, L)$ and $\mathcal{A}ut_{\neg\varphi}$, build $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ such that, for all π: $\pi \in Paths_{s_0}(\mathcal{M})$ and $\pi \not\models_L \varphi$ iff the atom-set trace of π through L is accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$.

What is exactly the product automaton? Why does it satisfy the required property?

Step 3. Check whether the language accepted by the automaton $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.
If so, conclude $\mathcal{M}, s \models \varphi$; otherwise conclude $\mathcal{M}, s \not\models \varphi$.

How is the check done? Why are the conclusions correct?

GNBAs

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for <u>infinite</u> words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for <u>infinite</u> words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\to \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \to$)

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \rightarrow$)
- $\mathcal{F} \subseteq \mathcal{P}(Q)$; the elements of $\mathcal{F}$ are sets of states called accepting sets

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \rightarrow$)
- $\mathcal{F} \subseteq \mathcal{P}(Q)$; the elements of $\mathcal{F}$ are sets of states called accepting sets

A word is an infinite sequence of letters $w = x_0 x_1 x_2 \ldots$ with each $x_i \in \Sigma$.

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\to \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \to$)
- $\mathcal{F} \subseteq \mathcal{P}(Q)$; the elements of $\mathcal{F}$ are sets of states called accepting sets

A word is an infinite sequence of letters $w = x_0 x_1 x_2 \dots$ with each $x_i \in \Sigma$.

Given a word $w = x_0 x_1 x_2 \dots$, a run for w is an infinite sequence of states $q_0 q_1 q_2 \dots$ with $q_0 \in I$ that transit via its letters: $q_i \xrightarrow{x_i} q_{i+1}$ for each $i \geq 0$.

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for infinite words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \rightarrow$)
- $\mathcal{F} \subseteq \mathcal{P}(Q)$; the elements of $\mathcal{F}$ are sets of states called accepting sets

A word is an infinite sequence of letters $w = x_0 x_1 x_2 \dots$ with each $x_i \in \Sigma$.

Given a word $w = x_0 x_1 x_2 \dots$, a run for w is an infinite sequence of states $q_0 q_1 q_2 \dots$ with $q_0 \in I$ that transit via its letters: $q_i \xrightarrow{x_i} q_{i+1}$ for each $i \geq 0$.

A run $q_0 q_1 q_2 \dots$ for w is called accepting if it visits infinitely often each of the accepting sets: for all $F \in \mathcal{F}$, the set $\{i \geq 0 \mid q_i \in F\}$ is infinite.

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for <u>infinite</u> words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\to \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \to$)
- $\mathcal{F} \subseteq \mathcal{P}(Q)$; the elements of $\mathcal{F}$ are sets of states called accepting sets

A word is an infinite sequence of letters $w = x_0 x_1 x_2 \ldots$ with each $x_i \in \Sigma$.

Given a word $w = x_0 x_1 x_2 \ldots$, a run for w is an infinite sequence of states $q_0 q_1 q_2 \ldots$ with $q_0 \in I$ that transit via its letters: $q_i \xrightarrow{x_i} q_{i+1}$ for each $i \geq 0$.

A run $q_0 q_1 q_2 \ldots$ for w is called accepting if it visits infinitely often each of the accepting sets: for all $F \in \mathcal{F}$, the set $\{i \geq 0 \mid q_i \in F\}$ is infinite.

A word w is said to be accepted by $\mathcal{A}ut$ if it has an accepting run in $\mathcal{A}ut$.

Step 1: What Type of Automaton Do We Use?

Think: nondeteministic finite automata, but used for <u>infinite</u> words

A Generalized Nondeterministic Büchi Automaton (GNBA for short) is a 5-tuple $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

- Σ is a finite set of letters, called the alphabet
- Q is a finite set of states
- $I \subseteq Q$ is a set of initial states
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation (write $q \xrightarrow{a} q'$ for $(q, a, q') \in \rightarrow$)
- $\mathcal{F} \subseteq \mathcal{P}(Q)$; the elements of $\mathcal{F}$ are sets of states called accepting sets

A word is an infinite sequence of letters $w = x_0 x_1 x_2 \ldots$ with each $x_i \in \Sigma$.

Given a word $w = x_0 x_1 x_2 \ldots$, a run for w is an infinite sequence of states $q_0 q_1 q_2 \ldots$ with $q_0 \in I$ that transit via its letters: $q_i \xrightarrow{x_i} q_{i+1}$ for each $i \geq 0$.

A run $q_0 q_1 q_2 \ldots$ for w is called accepting if it visits infinitely often each of the accepting sets: for all $F \in \mathcal{F}$, the set $\{i \geq 0 \mid q_i \in F\}$ is infinite.

A word w is said to be accepted by $\mathcal{A}ut$ if it has an accepting run in $\mathcal{A}ut$. The language accepted by $\mathcal{A}ut$ is the set of words accepted by $\mathcal{A}ut$.

GNBA – Example

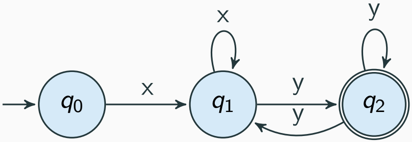
$\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

$$\begin{aligned}
&\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\} \\
&I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\} \\
&\to = \{\,(q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2), \\
&\qquad\quad (q_2, y, q_1),\ (q_2, y, q_2)\,\}
\end{aligned}$$

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

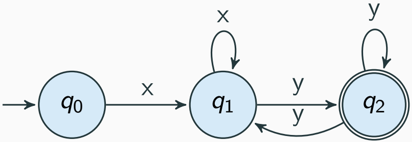
$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\rightarrow = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$



GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\to = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$

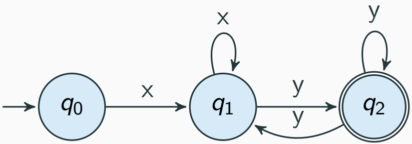


Note: Here, a run is accepting iff q_2 appears in it infinitely often.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\rightarrow = \{\,(q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$



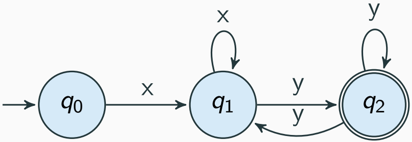
Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\rightarrow = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2) \,\}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

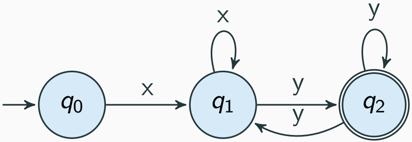
x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\, y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\rightarrow = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

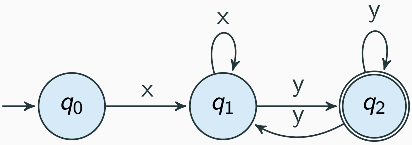
$x\,y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

$x\,y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\to = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\,y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

$x\,y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

y^∞ has no run.

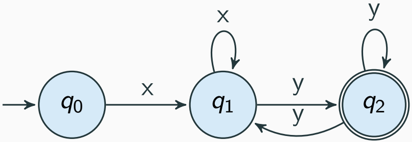
GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

$\Sigma = \{x, y\}$ $\quad Q = \{q_0, q_1, q_2\}$

$I = \{q_0\}$ $\quad \mathcal{F} = \{\{q_2\}\}$

$\to = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),$
$\quad (q_2, y, q_1),\ (q_2, y, q_2)\,\}$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\,y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

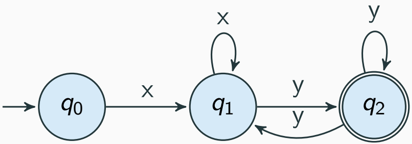
$x\,y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

y^∞ has no run. $\quad x\,y\,x^\infty$ has no run.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\rightarrow = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2) \,\}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\, y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

$x\, y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

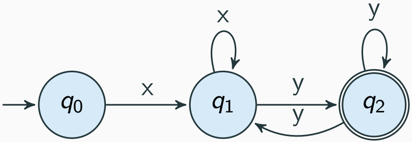
y^∞ has no run. $\quad x\, y\, x^\infty$ has no run.

How about $x\,(x\, y^2)^\infty$?

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\rightarrow = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2) \,\}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\,y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

$x\,y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

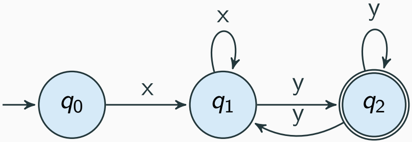
y^∞ has no run. $x\,y\,x^\infty$ has no run.

How about $x\,(x\,y^2)^\infty$? It has an accepting run, namely $q_0 q_1 (q_1 q_2 q_1)^\infty$.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\}$$
$$\to = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\,y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

$x\,y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

y^∞ has no run. $\quad x\,y\,x^\infty$ has no run.

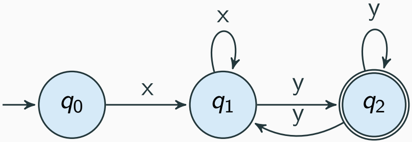
How about $x\,(x\,y^2)^\infty$? It has an accepting run, namely $q_0 q_1 (q_1 q_2 q_1)^\infty$.

$\mathrm{Lang}(\mathcal{A}ut)$ contains $x\,y^\infty$ and $x\,(x\,y^2)^\infty$, but not x^∞, $x\,y^2 x^\infty$, y^∞, $x\,y\,x^\infty$.

GNBA – Example

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\begin{aligned}
&\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\} \\
&I = \{q_0\} \qquad \mathcal{F} = \{\{q_2\}\} \\
&\rightarrow = \{\, (q_0, x, q_1),\ (q_1, x, q_1),\ (q_1, y, q_2), \\
&\qquad (q_2, y, q_1),\ (q_2, y, q_2)\,\}
\end{aligned}$$



Note: Here, a run is accepting iff q_2 appears in it infinitely often.

x^∞ has a run, namely $q_0 q_1^\infty$, but not accepting.

$x\,y^\infty$ has an accepting run, namely $q_0 q_1 q_2^\infty$.

$x\,y^2 x^\infty$ has a run, namely $q_0 q_1 q_2 q_1^\infty$, but not accepting.

y^∞ has no run. $x\,y\,x^\infty$ has no run.

How about $x\,(x\,y^2)^\infty$? It has an accepting run, namely $q_0 q_1 (q_1 q_2 q_1)^\infty$.

Lang($\mathcal{A}ut$) contains $x\,y^\infty$ and $x\,(x\,y^2)^\infty$, but not x^∞, $x\,y^2 x^\infty$, y^∞, $x\,y\,x^\infty$.

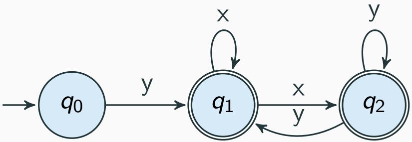
$$\begin{aligned}
\text{Lang}(\mathcal{A}ut) = &\{x^{m_1} y^{n_1} x^{m_2} y^{n_2} \ldots x^{m_p} y^{n_p} x^{m_{p+1}} y^\infty \mid p \geq 0, m_i > 0, n_i > 1\} \cup \\
&\{x^{m_1} y^{n_1} x^{m_2} y^{n_2} \ldots \mid m_i > 0, n_i > 1\}
\end{aligned}$$

Homework Exercise

Consider the following GNBA:

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\begin{array}{ll} \Sigma = \{x, y\} & Q = \{q_0, q_1, q_2\} \\ I = \{q_0\} & \mathcal{F} = \{\{q_1\},\ \{q_2\}\} \\ \multicolumn{2}{l}{\rightarrow = \{\,(q_0, y, q_1),\ (q_1, x, q_1),\ (q_1, x, q_2),} \\ \multicolumn{2}{l}{\qquad (q_2, y, q_1),\ (q_2, y, q_2)\,\}} \end{array}$$



Note: Here, a run is accepting iff <u>both</u> q_1 and q_2 appear in it infinitely often.

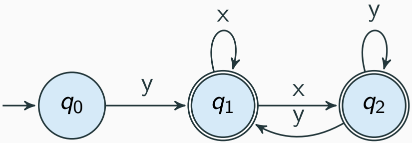
1. Which of the following words have runs, and which have accepting runs:
$y\,x^\infty$, $y\,x\,y^\infty$, x^∞, $(y\,x)^\infty$, $y\,(x^5\,y^3)^\infty$?
2. Can you describe the language accepted by $\mathcal{A}ut$?

Homework Exercise

Same questions as before, but for a slightly different GNBA – the only difference is shown in brown:

$\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where:

$$\Sigma = \{x, y\} \qquad Q = \{q_0, q_1, q_2\}$$
$$I = \{q_0\} \qquad \mathcal{F} = \{\{q_1, q_2\}\}$$
$$\rightarrow = \{\, (q_0, y, q_1),\ (q_1, x, q_1),\ (q_1, x, q_2),\ (q_2, y, q_1),\ (q_2, y, q_2)\,\}$$



Note: Here, a run is accepting iff either q_1 or q_2 appear in it infinitely often.

1. Which of the following words have runs, and which have accepting runs: $y\,x^\infty$, $y\,x\,y^\infty$, x^∞, $(y\,x)^\infty$, $y\,(x^5\,y^3)^\infty$?
2. Can you describe the language accepted by $\mathcal{A}ut$?

Step 1: From LTL Formulas to GNBAs

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_{\psi}$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \rightarrow, \mathcal{F})$.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \rightarrow, \mathcal{F})$.

The alphabet Σ of $\mathcal{A}ut_\psi$ is $\mathcal{P}(Atoms)$, so that words over this alphabet are atom-set traces.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \rightarrow, \mathcal{F})$.

The alphabet Σ of $\mathcal{A}ut_\psi$ is $\mathcal{P}(Atoms)$, so that words over this alphabet are atom-set traces.

We still need to define $Q, I, \rightarrow$ and $\mathcal{F}$.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \rightarrow, \mathcal{F})$.

The alphabet Σ of $\mathcal{A}ut_\psi$ is $\mathcal{P}(Atoms)$, so that words over this alphabet are atom-set traces.

We still need to define $Q, I, \rightarrow$ and $\mathcal{F}$.

> Main idea of the construction: We consider all possible "scenarios" that would make ψ true or false on a presumptive infinite sequence starting in some state, by looking at what can happen with its subformulas.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \rightarrow, \mathcal{F})$.

The alphabet Σ of $\mathcal{A}ut_\psi$ is $\mathcal{P}(Atoms)$, so that words over this alphabet are atom-set traces.

We still need to define $Q, I, \rightarrow$ and $\mathcal{F}$.

> Main idea of the construction: We consider all possible "scenarios" that would make ψ true or false on a presumptive infinite sequence starting in some state, by looking at what can happen with its subformulas.

So, for all subformulas of ψ, we look at all the scenarios of them being true or false in a consistent (i.e., non-contradictory) manner.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \rightarrow, \mathcal{F})$.

The alphabet Σ of $\mathcal{A}ut_\psi$ is $\mathcal{P}(Atoms)$, so that words over this alphabet are atom-set traces.

We still need to define $Q, I, \rightarrow$ and $\mathcal{F}$.

> Main idea of the construction: We consider all possible "scenarios" that would make ψ true or false on a presumptive infinite sequence starting in some state, by looking at what can happen with its subformulas.

So, for all subformulas of ψ, we look at all the scenarios of them being true or false in a consistent (i.e., non-contradictory) manner.

We will often write $\overline{\varphi}$ instead of $\neg\varphi$ for any formula φ.

Step 1: From LTL Formulas to GNBAs

Given an LTL formula ψ, we wish to construct a GNBA $\mathcal{A}ut_\psi$ that accepts precisely the atom-set traces of infinite sequences of states that satisfy ψ.

$\mathcal{A}ut_\psi$ will have the form $(\Sigma, Q, I, \to, \mathcal{F})$.

The alphabet Σ of $\mathcal{A}ut_\psi$ is $\mathcal{P}(Atoms)$, so that words over this alphabet are atom-set traces.

We still need to define $Q, I, \to$ and $\mathcal{F}$.

Main idea of the construction: We consider all possible "scenarios" that would make ψ true or false on a presumptive infinite sequence starting in some state, by looking at what can happen with its subformulas.

So, for all subformulas of ψ, we look at all the scenarios of them being true or false in a consistent (i.e., non-contradictory) manner.

We will often write $\overline{\varphi}$ instead of $\neg\varphi$ for any formula φ.

Important: We will identify (treat as if they are the same) $\overline{\overline{\varphi}}$ with φ – this is OK thanks to the Double Negation property.

Step 1: From LTL Formulas to GNBAs

By the subformulas of ψ, we mean all the formulas that appear as part of ψ.

Step 1: From LTL Formulas to GNBAs

By the subformulas of ψ, we mean all the formulas that appear as part of ψ.

Examples:

An atom a has only one subformula: a itself.

$\Box a$ has two subformulas: a and $\Box a$.

$\Box\Diamond a$ has three subformulas: a, $\Diamond a$ and $\Box\Diamond a$.

$\Box(\overline{\Box a} \vee b)$ has six subformulas a, b, $\Box a$, $\overline{\Box a}$, $\overline{\Box a} \vee b$ and $\Box(\overline{\Box a} \vee b)$.

Step 1: From LTL Formulas to GNBAs

By the subformulas of ψ, we mean all the formulas that appear as part of ψ.

Examples:

An atom a has only one subformula: a itself.

$\Box a$ has two subformulas: a and $\Box a$.

$\Box\Diamond a$ has three subformulas: a, $\Diamond a$ and $\Box\Diamond a$.

$\Box(\overline{\Box a} \vee b)$ has six subformulas a, b, $\Box a$, $\overline{\Box a}$, $\overline{\Box a} \vee b$ and $\Box(\overline{\Box a} \vee b)$.

The above is a "definition by example". Can you define the set of subformulas of a formula rigorously?

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\square a$.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\square a$. For $\square a$ to be true:

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true
- we also demand that $\Box a$ will be true in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true
- we also demand that $\Box a$ will be true in the next state

For $\Box a$ to be false:

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true
- we also demand that $\Box a$ will be true in the next state

For $\Box a$ to be false:

- we either demand a to be false

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\square a$. For $\square a$ to be true:

- we demand a to be true
- we also demand that $\square a$ will be true in the next state

For $\square a$ to be false:

- we either demand a to be false
- or allow a to be true, but demand that $\square a$ will be false in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true
- we also demand that $\Box a$ will be true in the next state

For $\Box a$ to be false:

- we either demand a to be false
- or allow a to be true, but demand that $\Box a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Box a\}$ $\{a, \overline{\Box a}\}$ $\{\overline{a}, \overline{\Box a}\}$

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true
- we also demand that $\Box a$ will be true in the next state

For $\Box a$ to be false:

- we either demand a to be false
- or allow a to be true, but demand that $\Box a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Box a\}$ $\quad$ $\{a, \overline{\Box a}\}$ $\quad$ $\{\overline{a}, \overline{\Box a}\}$
And we'll also have some requirements on moving forward to the next state.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box a$. For $\Box a$ to be true:

- we demand a to be true
- we also demand that $\Box a$ will be true in the next state

For $\Box a$ to be false:

- we either demand a to be false
- or allow a to be true, but demand that $\Box a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Box a\}$ $\{a, \overline{\Box a}\}$ $\{\overline{a}, \overline{\Box a}\}$
And we'll also have some requirements on moving forward to the next state.

Note that $\{\overline{a}, \Box a\}$ is not a possible scenario: It would be self-contradictory!

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\lozenge a$. For $\lozenge a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\lozenge a$ will be true in the next state

For $\lozenge a$ to be false:

- we demand a to be false

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

- we demand a to be false
- we also demand that $\Diamond a$ will be false in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

- we demand a to be false
- we also demand that $\Diamond a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Diamond a\}$ $\{\overline{a}, \Diamond a\}$ $\{\overline{a}, \overline{\Diamond a}\}$

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

- we demand a to be false
- we also demand that $\Diamond a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Diamond a\}$ $\{\overline{a}, \Diamond a\}$ $\{\overline{a}, \overline{\Diamond a}\}$
And again, we'll have some requirements on moving forward to the next state.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

- we demand a to be false
- we also demand that $\Diamond a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Diamond a\}$ $\{\overline{a}, \Diamond a\}$ $\{\overline{a}, \overline{\Diamond a}\}$
And again, we'll have some requirements on moving forward to the next state.

Note that $\{a, \overline{\Diamond a}\}$ is not a possible scenario: It would be self-contradictory.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

- we demand a to be false
- we also demand that $\Diamond a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Diamond a\}$ $\quad$ $\{\overline{a}, \Diamond a\}$ $\quad$ $\{\overline{a}, \overline{\Diamond a}\}$
And again, we'll have some requirements on moving forward to the next state.

Note that $\{a, \overline{\Diamond a}\}$ is not a possible scenario: It would be self-contradictory.

Homework Question: In which way is this similar to the discussion on the previous slide? Hint: $\Diamond$ and $\Box$ are dual to each other.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Diamond a$. For $\Diamond a$ to be true:

- we either demand a to be true
- or allow a to be false, but demand that $\Diamond a$ will be true in the next state

For $\Diamond a$ to be false:

- we demand a to be false
- we also demand that $\Diamond a$ will be false in the next state

Thus, for the current state we have the following three possible scenarios:
$\{a, \Diamond a\}$ $\{\bar{a}, \Diamond a\}$ $\{\bar{a}, \overline{\Diamond a}\}$
And again, we'll have some requirements on moving forward to the next state.

Note that $\{a, \overline{\Diamond a}\}$ is not a possible scenario: It would be self-contradictory.

Homework Question: In which way is this similar to the discussion on the previous slide? Hint: $\Diamond$ and $\Box$ are dual to each other.

In summary: We compute all possible scenarios for the correct state, and also remember some unfinished business for the next state.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box\Diamond a$. For $\Box\Diamond a$ to be true:

- we demand $\Diamond a$ to be true, hence:
  - we either demand a to be true
  - or allow a to be false, but demand that $\Diamond a$ will be true in the next state
- and demand $\Box\Diamond a$ to be true in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box\Diamond a$. For $\Box\Diamond a$ to be true:

- we demand $\Diamond a$ to be true, hence:
  - we either demand a to be true
  - or allow a to be false, but demand that $\Diamond a$ will be true in the next state
- and demand $\Box\Diamond a$ to be true in the next state

For $\Box\Diamond a$ to be false:

- we either demand $\Diamond a$ to be false, which also means that a is false
- or allow $\Diamond a$ to be true (but demand that $\Box\Diamond a$ will be false in the next state), in which case:
  - either a is true
  - or a is false but $\Diamond a$ will be true in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box\Diamond a$. For $\Box\Diamond a$ to be true:

- we demand $\Diamond a$ to be true, hence:
  - we either demand a to be true
  - or allow a to be false, but demand that $\Diamond a$ will be true in the next state
- and demand $\Box\Diamond a$ to be true in the next state

For $\Box\Diamond a$ to be false:

- we either demand $\Diamond a$ to be false, which also means that a is false
- or allow $\Diamond a$ to be true (but demand that $\Box\Diamond a$ will be false in the next state), in which case:
  - either a is true
  - or a is false but $\Diamond a$ will be true in the next state

Thus, for the current state we have five possible scenarios:

$\{a, \Diamond a, \Box\Diamond a\}$ $\{\bar{a}, \Diamond a, \Box\Diamond a\}$ $\{\bar{a}, \overline{\Diamond a}, \overline{\Box\Diamond a}\}$ $\{a, \Diamond a, \overline{\Box\Diamond a}\}$ $\{\bar{a}, \Diamond a, \overline{\Box\Diamond a}\}$

And we'll also have some requirements on moving forward to the next state.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box\Diamond a$. For $\Box\Diamond a$ to be true:

- we demand $\Diamond a$ to be true, hence:
  - we either demand a to be true
  - or allow a to be false, but demand that $\Diamond a$ will be true in the next state
- and demand $\Box\Diamond a$ to be true in the next state

For $\Box\Diamond a$ to be false:

- we either demand $\Diamond a$ to be false, which also means that a is false
- or allow $\Diamond a$ to be true (but demand that $\Box\Diamond a$ will be false in the next state), in which case:
  - either a is true
  - or a is false but $\Diamond a$ will be true in the next state

Thus, for the current state we have five possible scenarios:

$\{a, \Diamond a, \Box\Diamond a\}$ $\{\overline{a}, \Diamond a, \Box\Diamond a\}$ $\{\overline{a}, \overline{\Diamond a}, \overline{\Box\Diamond a}\}$ $\{a, \Diamond a, \overline{\Box\Diamond a}\}$ $\{\overline{a}, \Diamond a, \overline{\Box\Diamond a}\}$

And we'll also have some requirements on moving forward to the next state.

Note that $\{\overline{a}, \overline{\Diamond a}, \Box\Diamond a\}$ is not a possible scenario. Why? Also, can you identify and explain other impossible scenarios?

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box(\overline{\Box a} \vee b)$.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box(\overline{\Box a} \vee b)$. For $\Box(\overline{\Box a} \vee b)$ to be true:

- we demand $\overline{\Box a} \vee b$ to be true, hence:
  1. we either demand $\overline{\Box a}$ to be true, in which case:
     1.1. we either demand a to be false
     1.2. or we allow it to be true, but demand that $\overline{\Box a}$ will be false in the next state
  2. or allow $\overline{\Box a}$ to be false, and demand b to be true
- we also demand that $\Box(\overline{\Box a} \vee b)$ will be true in the next state

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box(\overline{\Box a} \vee b)$. For $\Box(\overline{\Box a} \vee b)$ to be true:

- we demand $\overline{\Box a} \vee b$ to be true, hence:
  1. we either demand $\overline{\Box a}$ to be true, in which case:
     1.1. we either demand a to be false
     1.2. or we allow it to be true, but demand that $\overline{\Box a}$ will be false in the next state
  2. or allow $\overline{\Box a}$ to be false, and demand b to be true
- we also demand that $\Box(\overline{\Box a} \vee b)$ will be true in the next state

And a similar analysis yields all possibilities for $\Box(\overline{\Box a} \vee b)$ to be false.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box(\overline{\Box a} \vee b)$. For $\Box(\overline{\Box a} \vee b)$ to be true:

- we demand $\overline{\Box a} \vee b$ to be true, hence:
  1. we either demand $\overline{\Box a}$ to be true, in which case:
     1.1. we either demand a to be false
     1.2. or we allow it to be true, but demand that $\overline{\Box a}$ will be false in the next state
  2. or allow $\overline{\Box a}$ to be false, and demand b to be true
- we also demand that $\Box(\overline{\Box a} \vee b)$ will be true in the next state

And a similar analysis yields all possibilities for $\Box(\overline{\Box a} \vee b)$ to be false.

Thus, for the current state we have the following possible scenarios:

1.1. $\{\bar{a},\ b,\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$ $\quad$ $\{\bar{a},\ \bar{b},\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$

1.2. $\{a,\ b,\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$ $\quad$ $\{a,\ \bar{b},\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$

2. $\{a,\ b,\ \Box a,\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$

... together with those for $\Box(\overline{\Box a} \vee b)$ to be false (not shown here).

And we'll also have some requirements on moving forward to the next state.

Step 1: From LTL Formulas to GNBAs – Discussion

Take ψ to be $\Box(\overline{\Box a} \vee b)$. For $\Box(\overline{\Box a} \vee b)$ to be true:

- we demand $\overline{\Box a} \vee b$ to be true, hence:
  1. we either demand $\overline{\Box a}$ to be true, in which case:
     1.1. we either demand a to be false
     1.2. or we allow it to be true, but demand that $\overline{\Box a}$ will be false in the next state
  2. or allow $\overline{\Box a}$ to be false, and demand b to be true
- we also demand that $\Box(\overline{\Box a} \vee b)$ will be true in the next state

And a similar analysis yields all possibilities for $\Box(\overline{\Box a} \vee b)$ to be false.

Thus, for the current state we have the following possible scenarios:

1.1. $\{\bar{a},\ b,\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$ $\quad$ $\{\bar{a},\ \bar{b},\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$

1.2. $\{a,\ b,\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$ $\quad$ $\{a,\ \bar{b},\ \overline{\Box a},\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$

2. $\{a,\ b,\ \Box a,\ \overline{\Box a} \vee b,\ \Box(\overline{\Box a} \vee b)\}$

... together with those for $\Box(\overline{\Box a} \vee b)$ to be false (not shown here).

And we'll also have some requirements on moving forward to the next state.

Note. These scenarios are <u>complete</u>, i.e., answer the truth question on all subformulas, and <u>consistent</u>, i.e., they do not have contradictions, e.g., containing both φ and $\overline{\varphi}$, or containing $\Box\varphi$ but not φ.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

We define $Cl(\psi)$, the closure of ψ, to be the set of all subformulas of ψ and their negations.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

We define $Cl(\psi)$, the closure of ψ, to be the set of all subformulas of ψ and their negations. For example:
$Cl\,(a) = \{a,\ \overline{a}\,\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

We define $Cl(\psi)$, the closure of ψ, to be the set of all subformulas of ψ and their negations. For example:
$Cl\,(a) = \{a,\, \overline{a}\,\}$
$Cl\,(\Box a) = \{a,\, \overline{a},\, \Box a,\, \overline{\Box a}\,\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

We define $Cl(\psi)$, the closure of ψ, to be the set of all subformulas of ψ and their negations. For example:

$Cl\,(a) = \{a,\ \overline{a}\,\}$
$Cl\,(\Box a) = \{a,\ \overline{a},\ \Box a,\ \overline{\Box a}\,\}$
$Cl\,(\Box\Diamond a) = \{a,\ \overline{a},\ \Diamond a,\ \overline{\Diamond a},\ \Box\Diamond a,\ \overline{\Box\Diamond a}\,\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

We define $Cl(\psi)$, the closure of ψ, to be the set of all subformulas of ψ and their negations. For example:

$Cl\,(a) = \{a,\, \overline{a}\,\}$

$Cl\,(\Box a) = \{a,\, \overline{a},\, \Box a,\, \overline{\Box a}\,\}$

$Cl\,(\Box\Diamond a) = \{a,\, \overline{a},\, \Diamond a,\, \overline{\Diamond a},\, \Box\Diamond a,\, \overline{\Box\Diamond a}\,\}$

$Cl\,(\Box(\overline{\Box a} \vee b)) =$

$\{a,\, \overline{a},\, b,\, \overline{b},\, \Box a, \overline{\Box a},\, \overline{\Box a} \vee b,\, \overline{\overline{\Box a} \vee b},\, \Box(\overline{\Box a} \vee b),\, \overline{\Box(\overline{\Box a} \vee b)}\,\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

$\Sigma = \mathcal{P}(Atoms)$, so that words over this alphabet will be atom-set traces.

We define $Cl(\psi)$, the closure of ψ, to be the set of all subformulas of ψ and their negations. For example:

$Cl\,(a) = \{a,\ \overline{a}\,\}$
$Cl\,(\Box a) = \{a,\ \overline{a},\ \Box a,\ \overline{\Box a}\,\}$
$Cl\,(\Box\Diamond a) = \{a,\ \overline{a},\ \Diamond a,\ \overline{\Diamond a},\ \Box\Diamond a,\ \overline{\Box\Diamond a}\,\}$
$Cl\,(\Box(\overline{\Box a} \vee b)) =$
$\{a,\ \overline{a},\ b,\ \overline{b},\ \Box a, \overline{\Box a},\ \overline{\Box a} \vee b,\ \overline{\overline{\Box a} \vee b},\ \Box(\overline{\Box a} \vee b),\ \overline{\Box(\overline{\Box a} \vee b)}\ \}$

$\overline{\overline{\Box a}}$ is not shown in $Cl\,(\Box(\overline{\Box a} \vee b))$, because it is the same as $\Box a$.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary where

- elementary means: complete and propositionally / temporally consistent

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary where

- elementary means: complete and propositionally / temporally consistent
- complete means: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K$ or $\overline{\varphi} \in K$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary where

- elementary means: complete and propositionally / temporally consistent
- complete means: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K$ or $\overline{\varphi} \in K$
- propositionally consistent means: for all $\varphi, \varphi_1, \varphi_2$
  - if $\varphi \in Cl(\psi)$, then $\varphi \in K$ implies $\overline{\varphi} \notin K$
  - if $\varphi_1 \wedge \varphi_2 \in Cl(\psi)$, then $\varphi_1 \wedge \varphi_2 \in K$ iff $\varphi_1 \in K$ and $\varphi_2 \in K$
  - if $\varphi_1 \vee \varphi_2 \in Cl(\psi)$, then $\varphi_1 \vee \varphi_2 \in K$ iff $\varphi_1 \in K$ or $\varphi_2 \in K$
  - if $\varphi_1 \to \varphi_2 \in Cl(\psi)$, then $\varphi_1 \to \varphi_2 \in K$ iff $[\varphi_1 \in K$ implies $\varphi_2 \in K]$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary where

- elementary means: complete and propositionally / temporally consistent
- complete means: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K$ or $\overline{\varphi} \in K$
- propositionally consistent means: for all $\varphi, \varphi_1, \varphi_2$
  - if $\varphi \in Cl(\psi)$, then $\varphi \in K$ implies $\overline{\varphi} \notin K$
  - if $\varphi_1 \wedge \varphi_2 \in Cl(\psi)$, then $\varphi_1 \wedge \varphi_2 \in K$ iff $\varphi_1 \in K$ and $\varphi_2 \in K$
  - if $\varphi_1 \vee \varphi_2 \in Cl(\psi)$, then $\varphi_1 \vee \varphi_2 \in K$ iff $\varphi_1 \in K$ or $\varphi_2 \in K$
  - if $\varphi_1 \to \varphi_2 \in Cl(\psi)$, then $\varphi_1 \to \varphi_2 \in K$ iff [$\varphi_1 \in K$ implies $\varphi_2 \in K$]

What do these conditions remind you of?

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary where

- elementary means: complete and propositionally / temporally consistent
- complete means: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K$ or $\overline{\varphi} \in K$
- propositionally consistent means: for all $\varphi, \varphi_1, \varphi_2$
  - if $\varphi \in Cl(\psi)$, then $\varphi \in K$ implies $\overline{\varphi} \notin K$
  - if $\varphi_1 \wedge \varphi_2 \in Cl(\psi)$, then $\varphi_1 \wedge \varphi_2 \in K$ iff $\varphi_1 \in K$ and $\varphi_2 \in K$
  - if $\varphi_1 \vee \varphi_2 \in Cl(\psi)$, then $\varphi_1 \vee \varphi_2 \in K$ iff $\varphi_1 \in K$ or $\varphi_2 \in K$
  - if $\varphi_1 \rightarrow \varphi_2 \in Cl(\psi)$, then $\varphi_1 \rightarrow \varphi_2 \in K$ iff [$\varphi_1 \in K$ implies $\varphi_2 \in K$]

  What do these conditions remind you of?
- temporally consistent means:
  - if $\Box\varphi \in Cl(\psi)$, then $\Box\varphi \in K$ implies $\varphi \in K$
  - if $\Diamond\varphi \in Cl(\psi)$, then $\varphi \in K$ implies $\Diamond\varphi \in K$
  - if $\varphi_1 \,\mathsf{U}\, \varphi_2 \in Cl(\psi)$, then $\varphi_2 \in K$ implies $\varphi_1 \mathsf{U} \varphi_2 \in K$
  - if $\varphi_1 \,\mathsf{U}\, \varphi_2 \in Cl(\psi)$, then $\varphi_1 \,\mathsf{U}\, \varphi_2 \in K$ and $\varphi_2 \notin K$ implies $\varphi_1 \in K$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: Q is the set of states of $\mathcal{A}ut_\psi$.

We define $Q =$ the set of all sets $K \subseteq Cl(\psi)$ that are elementary where

- elementary means: complete and propositionally / temporally consistent
- complete means: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K$ or $\overline{\varphi} \in K$
- propositionally consistent means: for all $\varphi, \varphi_1, \varphi_2$
  - if $\varphi \in Cl(\psi)$, then $\varphi \in K$ implies $\overline{\varphi} \notin K$
  - if $\varphi_1 \wedge \varphi_2 \in Cl(\psi)$, then $\varphi_1 \wedge \varphi_2 \in K$ iff $\varphi_1 \in K$ and $\varphi_2 \in K$
  - if $\varphi_1 \vee \varphi_2 \in Cl(\psi)$, then $\varphi_1 \vee \varphi_2 \in K$ iff $\varphi_1 \in K$ or $\varphi_2 \in K$
  - if $\varphi_1 \rightarrow \varphi_2 \in Cl(\psi)$, then $\varphi_1 \rightarrow \varphi_2 \in K$ iff $[\varphi_1 \in K$ implies $\varphi_2 \in K]$

  What do these conditions remind you of?
- temporally consistent means:
  - if $\Box\varphi \in Cl(\psi)$, then $\Box\varphi \in K$ implies $\varphi \in K$
  - if $\Diamond\varphi \in Cl(\psi)$, then $\varphi \in K$ implies $\Diamond\varphi \in K$
  - if $\varphi_1 \,\mathsf{U}\, \varphi_2 \in Cl(\psi)$, then $\varphi_2 \in K$ implies $\varphi_1 \mathsf{U} \varphi_2 \in K$
  - if $\varphi_1 \,\mathsf{U}\, \varphi_2 \in Cl(\psi)$, then $\varphi_1 \,\mathsf{U}\, \varphi_2 \in K$ and $\varphi_2 \notin K$ implies $\varphi_1 \in K$

In short: Q consists of all the scenarios for the truth or falsehood of the subformulas of ψ that are complete (do not let anything unsettled) and consistent (do not contain contradictions).

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: I is the set of initial states of $\mathcal{A}ut_\psi$.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: I is the set of initial states of $\mathcal{A}ut_\psi$.

We define $I =$ the set of all sets K in Q that contain ψ.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: I is the set of initial states of $\mathcal{A}ut_\psi$.

We define $I =$ the set of all sets K in Q that contain ψ.

So I incorporates all those scenarios where ψ is true.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: I is the set of initial states of $\mathcal{A}ut_\psi$.

We define $I =$ the set of all sets K in Q that contain ψ.

So I incorporates all those scenarios where ψ is true.

Intuition: The automaton will accept only the atom-set traces of sequences satisfying ψ, which therefore must start in scenarios where ψ is true.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: I is the set of initial states of $\mathcal{A}ut_\psi$.

We define $I =$ the set of all sets K in Q that contain ψ.

So I incorporates all those scenarios where ψ is true.

Intuition: The automaton will accept only the atom-set traces of sequences satisfying ψ, which therefore must start in scenarios where ψ is true.

Example, taking ψ to be $\Box\Diamond a$. Q consists of the following five sets, and I of only those that contain $\Box\Diamond a$ (the two ones shown in blue):

$\{a, \Diamond a, \Box\Diamond a\}$ $\{\bar{a}, \Diamond a, \Box\Diamond a\}$ $\{\bar{a}, \overline{\Diamond a}, \overline{\Box\Diamond a}\}$ $\{a, \Diamond a, \overline{\Box\Diamond a}\}$ $\{\bar{a}, \Diamond a, \overline{\Box\Diamond a}\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: I is the set of initial states of $\mathcal{A}ut_\psi$.

We define $I =$ the set of all sets K in Q that contain ψ.

So I incorporates all those scenarios where ψ is true.

Intuition: The automaton will accept only the atom-set traces of sequences satisfying ψ, which therefore must start in scenarios where ψ is true.

Example, taking ψ to be $\Box\Diamond a$. Q consists of the following five sets, and I of only those that contain $\Box\Diamond a$ (the two ones shown in blue):

$\{a, \Diamond a, \Box\Diamond a\}$ $\{\bar{a}, \Diamond a, \Box\Diamond a\}$ $\{\bar{a}, \overline{\Diamond a}, \overline{\Box\Diamond a}\}$ $\{a, \Diamond a, \overline{\Box\Diamond a}\}$ $\{\bar{a}, \Diamond a, \overline{\Box\Diamond a}\}$

In-class exercise: Please check this example against the definitions of Q and I.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

1. $A = K \cap Atoms$ (i.e., A consists of all the atoms of K)

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

1. $A = K \cap Atoms$ (i.e., A consists of all the atoms of K)
   Intuition: A is the set of atoms that are true in the scenario represented by K (hence can make a valid labeling at that state)

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

1. $A = K \cap Atoms$ (i.e., A consists of all the atoms of K)
   Intuition: A is the set of atoms that are true in the scenario represented by K (hence can make a valid labeling at that state)
2. For all $\varphi, \varphi_1, \varphi_2$

- If $\bigcirc\varphi \in Cl(\psi)$, then $\bigcirc\varphi \in K$ iff $\varphi \in K'$
- If $\Box\varphi \in Cl(\psi)$, then $\Box\varphi \in K$ iff $\varphi \in K$ and $\Box\varphi \in K'$
- If $\Diamond\varphi \in Cl(\psi)$, then $\Diamond\varphi \in K$ iff $\varphi \in K$ or $\Diamond\varphi \in K'$
- If $\varphi_1 \mathrm{U} \varphi_2 \in Cl(\psi)$, then $\varphi_1 \mathrm{U} \varphi_2 \in K$ iff $\varphi_2 \in K$ or $[\varphi_1 \in K$ and $\varphi_1 \mathrm{U} \varphi_2 \in K']$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

1. $A = K \cap Atoms$ (i.e., A consists of all the atoms of K)
   Intuition: A is the set of atoms that are true in the scenario represented by K (hence can make a valid labeling at that state)
2. For all $\varphi, \varphi_1, \varphi_2$

- If $\bigcirc\varphi \in Cl(\psi)$, then $\bigcirc\varphi \in K$ iff $\varphi \in K'$
- If $\Box\varphi \in Cl(\psi)$, then $\Box\varphi \in K$ iff $\varphi \in K$ and $\Box\varphi \in K'$
- If $\Diamond\varphi \in Cl(\psi)$, then $\Diamond\varphi \in K$ iff $\varphi \in K$ or $\Diamond\varphi \in K'$
- If $\varphi_1 \mathrm{U} \varphi_2 \in Cl(\psi)$, then $\varphi_1 \mathrm{U} \varphi_2 \in K$ iff $\varphi_2 \in K$ or $[\varphi_1 \in K$ and $\varphi_1 \mathrm{U} \varphi_2 \in K']$
  Intuition: K is the "now" scenario, and K' is the "next" scenario. The conditions state that these two are mutually consistent w.r.t. the temporal connectives,

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\rightarrow \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

1. $A = K \cap Atoms$ (i.e., A consists of all the atoms of K)
   Intuition: A is the set of atoms that are true in the scenario represented by K (hence can make a valid labeling at that state)
2. For all $\varphi, \varphi_1, \varphi_2$

- If $\bigcirc\varphi \in Cl(\psi)$, then $\bigcirc\varphi \in K$ iff $\varphi \in K'$
- If $\Box\varphi \in Cl(\psi)$, then $\Box\varphi \in K$ iff $\varphi \in K$ and $\Box\varphi \in K'$
- If $\Diamond\varphi \in Cl(\psi)$, then $\Diamond\varphi \in K$ iff $\varphi \in K$ or $\Diamond\varphi \in K'$
- If $\varphi_1 \mathrm{U} \varphi_2 \in Cl(\psi)$, then $\varphi_1 \mathrm{U} \varphi_2 \in K$ iff $\varphi_2 \in K$ or $[\varphi_1 \in K$ and $\varphi_1 \mathrm{U} \varphi_2 \in K']$
  Intuition: K is the "now" scenario, and K' is the "next" scenario. The conditions state that these two are mutually consistent w.r.t. the temporal connectives, e.g.:
  - $\bigcirc\varphi$ true "now" means that φ will be true "next"

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: $\to \subseteq Q \times \Sigma \times Q$ is $\mathcal{A}ut_\psi$'s transition relation. Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

Given $A \in \Sigma$ and $K, K' \in Q$, we define $K \xrightarrow{A} K'$ to mean the following:

1. $A = K \cap Atoms$ (i.e., A consists of all the atoms of K)
   Intuition: A is the set of atoms that are true in the scenario represented by K (hence can make a valid labeling at that state)
2. For all $\varphi, \varphi_1, \varphi_2$

- If $\bigcirc\varphi \in Cl(\psi)$, then $\bigcirc\varphi \in K$ iff $\varphi \in K'$
- If $\Box\varphi \in Cl(\psi)$, then $\Box\varphi \in K$ iff $\varphi \in K$ and $\Box\varphi \in K'$
- If $\Diamond\varphi \in Cl(\psi)$, then $\Diamond\varphi \in K$ iff $\varphi \in K$ or $\Diamond\varphi \in K'$
- If $\varphi_1 \mathsf{U} \varphi_2 \in Cl(\psi)$, then $\varphi_1 \mathsf{U} \varphi_2 \in K$ iff $\varphi_2 \in K$ or $[\varphi_1 \in K$ and $\varphi_1 \mathsf{U} \varphi_2 \in K']$
  Intuition: K is the "now" scenario, and K' is the "next" scenario. The conditions state that these two are mutually consistent w.r.t. the temporal connectives, e.g.:
  - $\bigcirc\varphi$ true "now" means that φ will be true "next"
  - $\Diamond\varphi$ true "now" means that either φ is true "now" or $\Diamond\varphi$ will be postponed to "next" (part of the "unfinished business")

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(\mathit{Atoms})$ and $Q \subseteq \mathcal{P}(\mathit{Formulas})$.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(\mathit{Atoms})$ and $Q \subseteq \mathcal{P}(\mathit{Formulas})$.

We define $\mathcal{F} = \{\mathit{Fulfill}(\Box\varphi) \mid \Box\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\Diamond\varphi) \mid \Diamond\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \in \mathit{Cl}(\psi)\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(\mathit{Atoms})$ and $Q \subseteq \mathcal{P}(\mathit{Formulas})$.

We define $\mathcal{F} = \{\mathit{Fulfill}(\Box\varphi) \mid \Box\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\Diamond\varphi) \mid \Diamond\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\varphi_1 \mathsf{U}\, \varphi_2) \mid \varphi_1 \mathsf{U}\, \varphi_2 \in \mathit{Cl}(\psi)\}$ where:

- $\mathit{Fulfill}(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(\mathit{Atoms})$ and $Q \subseteq \mathcal{P}(\mathit{Formulas})$.

We define $\mathcal{F} = \{\mathit{Fulfill}(\Box\varphi) \mid \Box\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\Diamond\varphi) \mid \Diamond\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \in \mathit{Cl}(\psi)\}$ where:

- $\mathit{Fulfill}(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $\mathit{Fulfill}(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(\mathit{Atoms})$ and $Q \subseteq \mathcal{P}(\mathit{Formulas})$.

We define $\mathcal{F} = \{\mathit{Fulfill}(\Box\varphi) \mid \Box\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\Diamond\varphi) \mid \Diamond\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \in \mathit{Cl}(\psi)\}$ where:

- $\mathit{Fulfill}(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $\mathit{Fulfill}(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $\mathit{Fulfill}(\varphi_1 \,\mathsf{U}\, \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

We define $\mathcal{F} = \{Fulfill(\Box\varphi) \mid \Box\varphi \in Cl(\psi)\} \cup \{Fulfill(\Diamond\varphi) \mid \Diamond\varphi \in Cl(\psi)\} \cup \{Fulfill(\varphi_1 \mathsf{U} \varphi_2) \mid \varphi_1 \mathsf{U} \varphi_2 \in Cl(\psi)\}$ where:

- $Fulfill(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $Fulfill(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $Fulfill(\varphi_1 \mathsf{U} \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Intuition – the long-term fulfillment of the "unfinished business" aspect:

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

We define $\mathcal{F} = \{Fulfill(\Box\varphi) \mid \Box\varphi \in Cl(\psi)\} \cup \{Fulfill(\Diamond\varphi) \mid \Diamond\varphi \in Cl(\psi)\} \cup \{Fulfill(\varphi_1 \mathsf{U} \varphi_2) \mid \varphi_1 \mathsf{U} \varphi_2 \in Cl(\psi)\}$ where:

- $Fulfill(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $Fulfill(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $Fulfill(\varphi_1 \mathsf{U} \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Intuition – the long-term fulfillment of the “unfinished business” aspect:

- Say $\Diamond\varphi$ is part of some “now” scenario $K \in Q$ (namely, $\Diamond\varphi \in K$).

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

We define $\mathcal{F} = \{Fulfill(\Box\varphi) \mid \Box\varphi \in Cl(\psi)\} \cup \{Fulfill(\Diamond\varphi) \mid \Diamond\varphi \in Cl(\psi)\} \cup \{Fulfill(\varphi_1 \mathsf{U} \varphi_2) \mid \varphi_1 \mathsf{U} \varphi_2 \in Cl(\psi)\}$ where:

- $Fulfill(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $Fulfill(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $Fulfill(\varphi_1 \mathsf{U} \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Intuition – the long-term fulfillment of the "unfinished business" aspect:

- Say $\Diamond\varphi$ is part of some "now" scenario $K \in Q$ (namely, $\Diamond\varphi \in K$).
- Then, by temporal consistency, either φ is part of the "now" scenario ($\varphi \in K$) or $\Diamond\varphi$ is postponed to the "next" scenario ($\Diamond\varphi \in K'$).

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

We define $\mathcal{F} = \{Fulfill(\Box\varphi) \mid \Box\varphi \in Cl(\psi)\} \cup \{Fulfill(\Diamond\varphi) \mid \Diamond\varphi \in Cl(\psi)\} \cup \{Fulfill(\varphi_1 \, \mathsf{U} \, \varphi_2) \mid \varphi_1 \, \mathsf{U} \, \varphi_2 \in Cl(\psi)\}$ where:

- $Fulfill(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $Fulfill(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $Fulfill(\varphi_1 \, \mathsf{U} \, \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Intuition – the long-term fulfillment of the "unfinished business" aspect:

- Say $\Diamond\varphi$ is part of some "now" scenario $K \in Q$ (namely, $\Diamond\varphi \in K$).
- Then, by temporal consistency, either φ is part of the "now" scenario ($\varphi \in K$) or $\Diamond\varphi$ is postponed to the "next" scenario ($\Diamond\varphi \in K'$).
- But for how long can it be postponed? For a while, but not infinitely.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

We define $\mathcal{F} = \{Fulfill(\Box\varphi) \mid \Box\varphi \in Cl(\psi)\} \cup \{Fulfill(\Diamond\varphi) \mid \Diamond\varphi \in Cl(\psi)\} \cup \{Fulfill(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \in Cl(\psi)\}$ where:

- $Fulfill(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $Fulfill(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $Fulfill(\varphi_1 \,\mathsf{U}\, \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Intuition – the long-term fulfillment of the "unfinished business" aspect:

- Say $\Diamond\varphi$ is part of some "now" scenario $K \in Q$ (namely, $\Diamond\varphi \in K$).
- Then, by temporal consistency, either φ is part of the "now" scenario ($\varphi \in K$) or $\Diamond\varphi$ is postponed to the "next" scenario ($\Diamond\varphi \in K'$).
- But for how long can it be postponed? For a while, but not infinitely.
- So φ should be fulfilled in a "now" from the future $K^{'''\cdots'}$ after a finite chain of transitions $K \xrightarrow{A_1} K' \xrightarrow{A_2} K'' \rightarrow \ldots \rightarrow K^{''\cdots'}$.

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(\mathit{Atoms})$ and $Q \subseteq \mathcal{P}(\mathit{Formulas})$.

We define $\mathcal{F} = \{\mathit{Fulfill}(\Box\varphi) \mid \Box\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\Diamond\varphi) \mid \Diamond\varphi \in \mathit{Cl}(\psi)\} \cup \{\mathit{Fulfill}(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \in \mathit{Cl}(\psi)\}$ where:

- $\mathit{Fulfill}(\Box\varphi) = \{K \in Q \mid \varphi \in K \text{ implies } \Box\varphi \in K\}$
- $\mathit{Fulfill}(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K \text{ implies } \varphi \in K\}$
- $\mathit{Fulfill}(\varphi_1 \,\mathsf{U}\, \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K \text{ implies } \varphi_2 \in K\}$

Intuition – the long-term fulfillment of the "unfinished business" aspect:

- Say $\Diamond\varphi$ is part of some "now" scenario $K \in Q$ (namely, $\Diamond\varphi \in K$).
- Then, by temporal consistency, either φ is part of the "now" scenario ($\varphi \in K$) or $\Diamond\varphi$ is postponed to the "next" scenario ($\Diamond\varphi \in K'$).
- But for how long can it be postponed? For a while, but not infinitely.
- So φ should be fulfilled in a "now" from the future $K'''^{\cdots\prime}$ after a finite chain of transitions $K \xrightarrow{A_1} K' \xrightarrow{A_2} K'' \rightarrow \ldots \rightarrow K''^{\cdots\prime}$.
- Thus, infinitely often, if $\Diamond\varphi$ is in the "now" then φ is also in the "now".

Step 1: From LTL Formulas to GNBAs – Definition

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ Recall: $\mathcal{F} \subseteq \mathcal{P}(Q)$ is $\mathcal{A}ut_\psi$'s set of accepting sets.
Also, recall that $\Sigma = \mathcal{P}(Atoms)$ and $Q \subseteq \mathcal{P}(Formulas)$.

We define $\mathcal{F} = \{Fulfill(\Box\varphi) \mid \Box\varphi \in Cl(\psi)\} \cup \{Fulfill(\Diamond\varphi) \mid \Diamond\varphi \in Cl(\psi)\} \cup \{Fulfill(\varphi_1 \,\mathsf{U}\, \varphi_2) \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \in Cl(\psi)\}$ where:

- $Fulfill(\Box\varphi) = \{K \in Q \mid \varphi \in K$ implies $\Box\varphi \in K\}$
- $Fulfill(\Diamond\varphi) = \{K \in Q \mid \Diamond\varphi \in K$ implies $\varphi \in K\}$
- $Fulfill(\varphi_1 \,\mathsf{U}\, \varphi_2) = \{K \in Q \mid \varphi_1 \mathsf{U} \varphi_2 \in K$ implies $\varphi_2 \in K\}$

Intuition – the long-term fulfillment of the "unfinished business" aspect:

- Say $\Diamond\varphi$ is part of some "now" scenario $K \in Q$ (namely, $\Diamond\varphi \in K$).
- Then, by temporal consistency, either φ is part of the "now" scenario ($\varphi \in K$) or $\Diamond\varphi$ is postponed to the "next" scenario ($\Diamond\varphi \in K'$).
- But for how long can it be postponed? For a while, but not infinitely.
- So φ should be fulfilled in a "now" from the future $K'''^{\cdots'}$ after a finite chain of transitions $K \xrightarrow{A_1} K' \xrightarrow{A_2} K'' \rightarrow \ldots \rightarrow K''^{\cdots'}$.
- Thus, infinitely often, if $\Diamond\varphi$ is in the "now" then φ is also in the "now".

And similarly for $\Box$ and U – they have their own long-term fulfillment goals.

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a,\ \overline{a},\ \Diamond a,\ \overline{\Diamond a}\,\}$

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a,\, \overline{a},\, \Diamond a,\, \overline{\Diamond a}\,\}$

$\mathcal{A}ut_{\psi} = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{\, \emptyset,\, \{a\}\,\}$

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a,\, \overline{a},\, \Diamond a,\, \overline{\Diamond a}\,\}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{\, \emptyset,\, \{a\}\,\}$

Q consists of all elementary subsets of $Cl(\psi)$
Remember: Elementary = propositionally and temporally consistent, and complete
Thus: $Q = \{\ \{a,\, \Diamond a\},\ \{\overline{a},\, \Diamond a\},\ \{\overline{a},\, \overline{\Diamond a}\}\ \}$

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a,\, \bar{a},\, \Diamond a,\, \overline{\Diamond a}\,\}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{\, \emptyset,\, \{a\}\,\}$

Q consists of all elementary subsets of $Cl(\psi)$

Remember: Elementary = propositionally and temporally consistent, and complete

Thus: $Q = \{\ \{a,\, \Diamond a\},\ \{\bar{a},\, \Diamond a\},\ \{\bar{a},\, \overline{\Diamond a}\}\ \}$

I consists of all sets from Q that contain ψ

Thus: $I\ = \{\ \{a,\, \Diamond a\},\ \{\bar{a},\, \Diamond a\}\ \}$

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a,\, \overline{a},\, \Diamond a,\, \overline{\Diamond a}\,\}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{\, \emptyset,\, \{a\}\,\}$

Q consists of all elementary subsets of $Cl(\psi)$
Remember: Elementary = propositionally and temporally consistent, and complete
Thus: $Q = \{\ \{a,\, \Diamond a\},\ \{\overline{a},\, \Diamond a\},\ \{\overline{a},\, \overline{\Diamond a}\}\ \}$

I consists of all sets from Q that contain ψ
Thus: $I\ = \{\ \{a,\, \Diamond a\},\ \{\overline{a},\, \Diamond a\}\ \}$

$\mathcal{F}$ contains one set, *Fulfill*$(\Diamond a)$.

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a, \bar{a}, \Diamond a, \overline{\Diamond a}\,\}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{\, \emptyset, \{a\}\,\}$

Q consists of all elementary subsets of $Cl(\psi)$
Remember: Elementary = propositionally and temporally consistent, and complete
Thus: $Q = \{\ \{a, \Diamond a\}, \ \{\bar{a}, \Diamond a\}, \ \{\bar{a}, \overline{\Diamond a}\}\ \}$

I consists of all sets from Q that contain ψ
Thus: $I \ = \{\ \{a, \Diamond a\}, \ \{\bar{a}, \Diamond a\}\ \}$

$\mathcal{F}$ contains one set, $\mathit{Fulfill}(\Diamond a)$.
$\mathit{Fulfill}(\Diamond a)$ contains those sets K from Q with the following property:
$\Diamond a \in K$ implies $a \in K$.

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{ a, \overline{a}, \Diamond a, \overline{\Diamond a} \}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{ \emptyset, \{a\} \}$

Q consists of all elementary subsets of $Cl(\psi)$
Remember: Elementary = propositionally and temporally consistent, and complete
Thus: $Q = \{ \{a, \Diamond a\}, \{\overline{a}, \Diamond a\}, \{\overline{a}, \overline{\Diamond a}\} \}$

I consists of all sets from Q that contain ψ
Thus: $I = \{ \{a, \Diamond a\}, \{\overline{a}, \Diamond a\} \}$

$\mathcal{F}$ contains one set, $Fulfill(\Diamond a)$.
$Fulfill(\Diamond a)$ contains those sets K from Q with the following property:
$\Diamond a \in K$ implies $a \in K$.
This is true about all sets from Q, except for the one that contains $\Diamond a$ but not a,
hence $Fulfill(\Diamond a) = \{ \{a, \Diamond a\}, \{\overline{a}, \overline{\Diamond a}\} \}$

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{\, a, \bar{a}, \Diamond a, \overline{\Diamond a}\,\}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{\, \emptyset, \{a\}\,\}$

Q consists of all elementary subsets of $Cl(\psi)$
Remember: Elementary = propositionally and temporally consistent, and complete
Thus: $Q = \{\ \{a, \Diamond a\},\ \{\bar{a}, \Diamond a\},\ \{\bar{a}, \overline{\Diamond a}\}\ \}$

I consists of all sets from Q that contain ψ
Thus: $I \ = \{\ \{a, \Diamond a\},\ \{\bar{a}, \Diamond a\}\ \}$

$\mathcal{F}$ contains one set, $\mathit{Fulfill}(\Diamond a)$.
$\mathit{Fulfill}(\Diamond a)$ contains those sets K from Q with the following property:
$\Diamond a \in K$ implies $a \in K$.
This is true about all sets from Q, except for the one that contains $\Diamond a$ but not a,
hence $\mathit{Fulfill}(\Diamond a) = \{\ \{a, \Diamond a\},\ \{\bar{a}, \overline{\Diamond a}\}\ \}$
Thus: $\mathcal{F} = \{\, \mathit{Fulfill}(\Diamond a)\,\} = \{\ \{\ \{a, \Diamond a\},\ \{\bar{a}, \overline{\Diamond a}\}\ \}\ \}$

Running Example

Let ψ be $\Diamond a$. Note that $Cl(\psi) = \{ a, \bar{a}, \Diamond a, \overline{\Diamond a} \}$

$\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ consists of the following:

$\Sigma = \mathcal{P}(\{a\}) = \{ \emptyset, \{a\} \}$

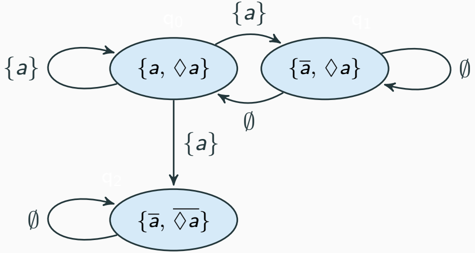
Q consists of all elementary subsets of $Cl(\psi)$
Remember: Elementary = propositionally and temporally consistent, and complete
Thus: $Q = \{ \{a, \Diamond a\}, \{\bar{a}, \Diamond a\}, \{\bar{a}, \overline{\Diamond a}\} \}$

I consists of all sets from Q that contain ψ
Thus: $I = \{ \{a, \Diamond a\}, \{\bar{a}, \Diamond a\} \}$

$\mathcal{F}$ contains one set, $Fulfill(\Diamond a)$.
$Fulfill(\Diamond a)$ contains those sets K from Q with the following property:
$\Diamond a \in K$ implies $a \in K$.
This is true about all sets from Q, except for the one that contains $\Diamond a$ but not a,
hence $Fulfill(\Diamond a) = \{ \{a, \Diamond a\}, \{\bar{a}, \overline{\Diamond a}\} \}$
Thus: $\mathcal{F} = \{ Fulfill(\Diamond a) \} = \{ \{ \{a, \Diamond a\}, \{\bar{a}, \overline{\Diamond a}\} \} \}$

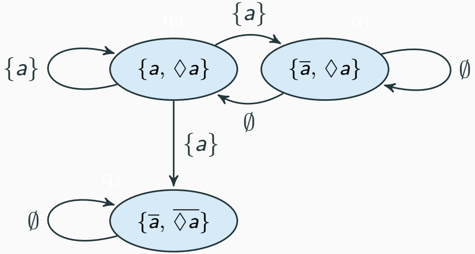
Note: Σ is a set of sets of atoms; Q, I and $Fulfill(\Diamond a)$ are sets of sets of formulas; $\mathcal{F}$ is a set of sets of sets of formulas.

The transition relation $\rightarrow$ is shown below:



Running Example

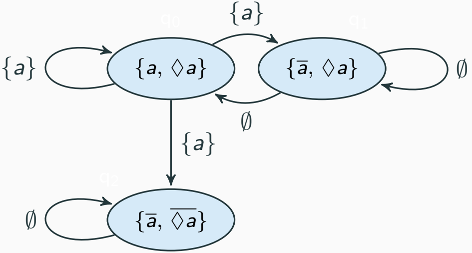
The transition relation $\to$ is shown below:



Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

Running Example

The transition relation $\rightarrow$ is shown below:

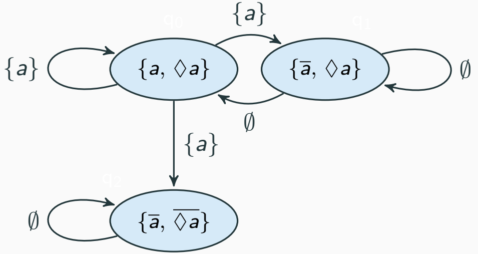


Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

1. A consists of all the atoms in K.

Running Example

The transition relation $\to$ is shown below:

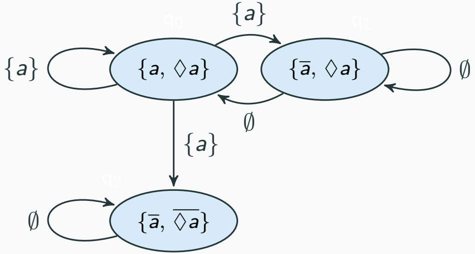


Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

1. A consists of all the atoms in K. This means that showing A is redundant, since it is determined by the source K of the transition:
    - All transitions coming out of $\{a, \Diamond a\}$ have $A = \{a\}$

Running Example

The transition relation $\rightarrow$ is shown below:



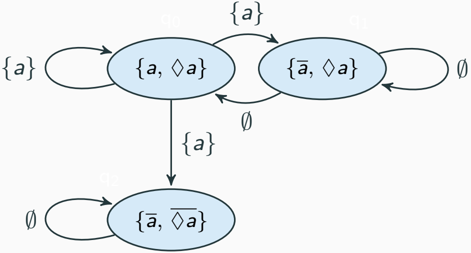
Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

1. A consists of all the atoms in K. This means that showing A is redundant, since it is determined by the source K of the transition:

- All transitions coming out of $\{a, \Diamond a\}$ have $A = \{a\}$
- All transitions coming out of $\{\overline{a}, \Diamond a\}$ have $A = \emptyset$

Running Example

The transition relation $\rightarrow$ is shown below:



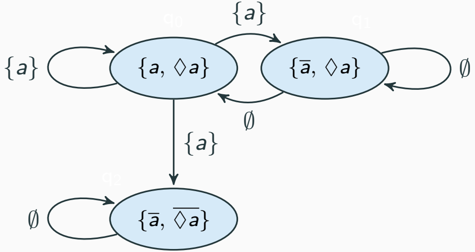
Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

1. A consists of all the atoms in K. This means that showing A is redundant, since it is determined by the source K of the transition:

- All transitions coming out of $\{a, \Diamond a\}$ have $A = \{a\}$
- All transitions coming out of $\{\overline{a}, \Diamond a\}$ have $A = \emptyset$
- All transitions coming out of $\{\overline{a}, \overline{\Diamond a}\}$ have $A = \emptyset$

Running Example

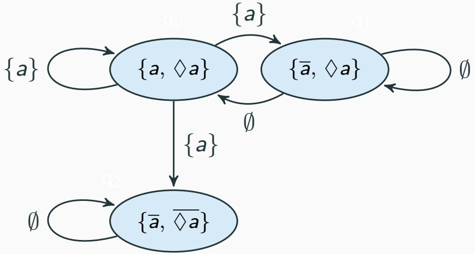
The transition relation $\rightarrow$ is shown below:



Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

Running Example

The transition relation $\rightarrow$ is shown below:

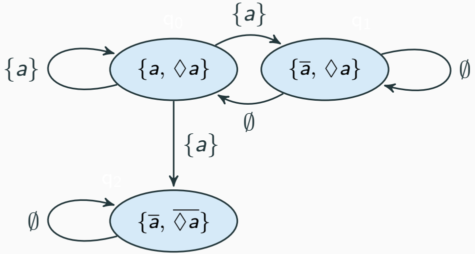


Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

2. $\Diamond a \in K$ iff $a \in K$ or $\Diamond a \in K'$.

Running Example

The transition relation $\rightarrow$ is shown below:

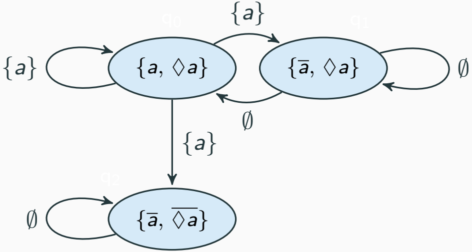


Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

2. $\Diamond a \in K$ iff $a \in K$ or $\Diamond a \in K'$. So, for example:
   - There is a transition between $\{a, \Diamond a\}$ and itself, since this condition holds for $K = K' = \{a, \Diamond a\}$

Running Example

The transition relation $\to$ is shown below:

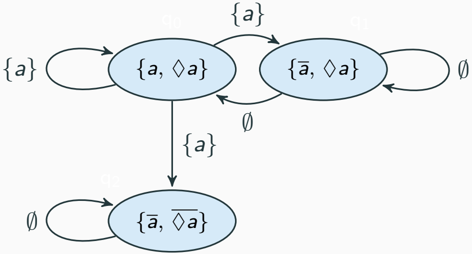


Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

2. $\Diamond a \in K$ iff $a \in K$ or $\Diamond a \in K'$. So, for example:
   - There is a transition between $\{a, \Diamond a\}$ and itself, since this condition holds for $K = K' = \{a, \Diamond a\}$
   - There is a transition between $\{a, \Diamond a\}$ and $\{\overline{a}, \Diamond a\}$, since this condition holds for $K = \{a, \Diamond a\}$ and $K' = \{\overline{a}, \Diamond a\}$

Running Example

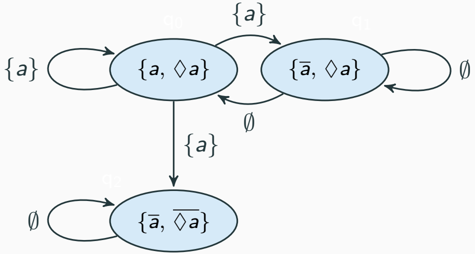
The transition relation $\rightarrow$ is shown below:



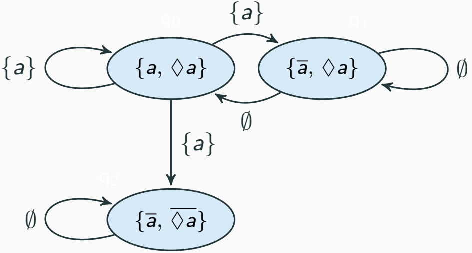
Note that for any $K, K' \in Q$ and $A \in \Sigma$, the conditions defining $K \xrightarrow{A} K'$ are:

2. $\Diamond a \in K$ iff $a \in K$ or $\Diamond a \in K'$. So, for example:
   - There is a transition between $\{a, \Diamond a\}$ and itself, since this condition holds for $K = K' = \{a, \Diamond a\}$
   - There is a transition between $\{a, \Diamond a\}$ and $\{\bar{a}, \Diamond a\}$, since this condition holds for $K = \{a, \Diamond a\}$ and $K' = \{\bar{a}, \Diamond a\}$
   - There is no transition between $\{\bar{a}, \overline{\Diamond a}\}$ and $\{\bar{a}, \Diamond a\}$, since this condition fails for $K = \{\bar{a}, \overline{\Diamond a}\}$ and $K' = \{\bar{a}, \Diamond a\}$ – indeed, $\Diamond a \in K'$ but $\Diamond a \notin K$

Running Example



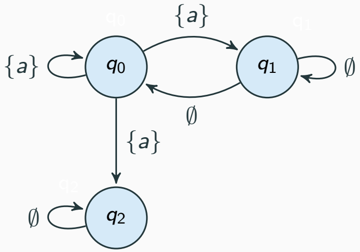
Running Example



$I \;=\{\ \{a, \Diamond a\},\ \{\overline{a}, \Diamond a\}\ \}$

$\mathcal{F} = \{\mathit{Fulfill}(\Diamond a)\} = \{\ \{\ \{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\}\ \}\ \}$

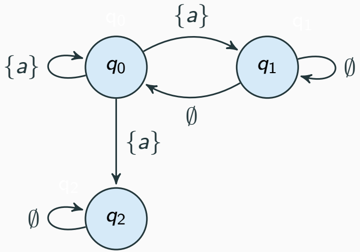
Running Example



$I = \{ q_0, q_1 \}$

$\mathcal{F} = \{Fulfill(\Diamond a)\} = \{ \{ q_0, q_2 \} \}$

Running Example

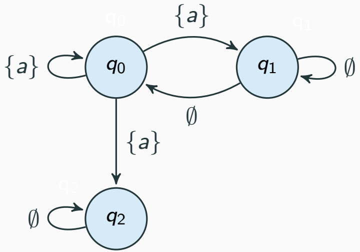


$I \; = \{\; q_0, \; q_1 \;\}$

$\mathcal{F} = \{\mathit{Fulfill}(\Diamond a)\} = \{\; \{\; q_0, \; q_2 \;\} \;\}$

Accepted language of $\mathcal{A}ut_{\Diamond a}$?

Running Example



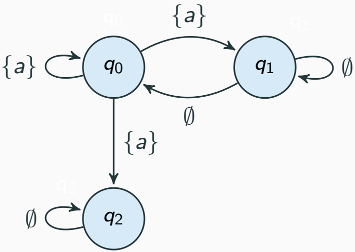
$I = \{ q_0,\ q_1 \}$

$\mathcal{F} = \{\mathit{Fulfill}(\Diamond a)\} = \{ \{ q_0,\ q_2 \} \}$

Accepted language of $\mathcal{A}ut_{\Diamond a}$?

All words of the form $A_0 A_1 A_2 \ldots$ (with each $A_i \subseteq \{a\}$) such that there exists $j \geq 0$ with $A_i = \{a\}$.

Running Example



$I \; = \{ \; q_0, \;\; q_1 \; \}$

$\mathcal{F} = \{\mathit{Fulfill}(\Diamond a)\} = \{ \; \{ \; q_0, \;\; q_2 \; \} \; \}$

Accepted language of $\mathcal{A}ut_{\Diamond a}$?

All words of the form $A_0 A_1 A_2 \ldots$ (with each $A_i \subseteq \{a\}$) such that there exists $j \geq 0$ with $A_i = \{a\}$.

... and this is exactly the property we need from the atom-set trace of a sequence π satisfying $\Diamond a$.

Step 1: From LTL Formulas to GNBAs – Correctness

Next, we will prove the following:

Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$

$$\pi \models_L \psi \;\text{ iff }\; \mathcal{A}ut_\psi \text{ accepts the atom-set trace of } \pi \text{ through } L.$$

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$.
We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it.
We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run,

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$.
We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it.
We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$.
We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it.
We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$ – thanks to the properties of satisfaction, incl. the expansion laws

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$.
We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it.
We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$ – thanks to the properties of satisfaction, incl. the expansion laws

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.

We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$ – thanks to the properties of satisfaction, incl. the expansion laws

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$ – also thanks to the properties of satisfaction.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$.
We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it.
We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$ – thanks to the properties of satisfaction, incl. the expansion laws

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$ – also thanks to the properties of satisfaction. For example:
Given $\Diamond\varphi \in Cl(\psi)$, we must check that $\Diamond\varphi \in K_i$, i.e., $\pi^i \models_L \Diamond\varphi$, implies $\varphi \in K_i$, i.e., $\pi^i \models_L \varphi$, for infinitely many i's.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$ – thanks to the properties of satisfaction, incl. the expansion laws

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$ – also thanks to the properties of satisfaction. For example:
Given $\Diamond\varphi \in Cl(\psi)$, we must check that $\Diamond\varphi \in K_i$, i.e., $\pi^i \models_L \Diamond\varphi$, implies $\varphi \in K_i$, i.e., $\pi^i \models_L \varphi$, for infinitely many i's.
This is true because $\pi^i \models_L \Diamond\varphi$ implies that there exists $j \geq i$ such that $\pi^j \models_L \Diamond\varphi$ and $\pi^j \models_L \varphi$.

Step 1: From LTL Formulas to GNBAs – Correctness

Left-to-Right Implication of Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \to \mathcal{P}(Atom)$:

If $\pi \models_L \psi$ then $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

Proof idea. Assume $\pi \models_L \psi$. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. We must show that $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$, i.e., it has an accepting run for it. We take the run to be $K_0 K_1 K_2 \ldots$ where $K_i = \{\varphi \in Cl(\psi) \mid \pi^i \models_L \varphi\}$.
We can check that:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

- $K_i \in Q$, i.e., K_i is elementary – thanks to the properties of satisfaction
- $K_0 \in I$, i.e., $\psi \in K_0$ – immediate, since $\pi \models_L \psi$.
- $K_i \xrightarrow{A_i} K_{i+1}$ – thanks to the properties of satisfaction, incl. the expansion laws

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$ – also thanks to the properties of satisfaction. For example:
Given $\Diamond\varphi \in Cl(\psi)$, we must check that $\Diamond\varphi \in K_i$, i.e., $\pi^i \models_L \Diamond\varphi$, implies $\varphi \in K_i$, i.e., $\pi^i \models_L \varphi$, for infinitely many i's.
This is true because $\pi^i \models_L \Diamond\varphi$ implies that there exists $j \geq i$ such that $\pi^j \models_L \Diamond\varphi$ and $\pi^j \models_L \varphi$. TYU: Prove this last statement.

Step 1: From LTL Formulas to GNBAs – Correctness

Right-to-Left Implication of Correctness Thm. for Step 1. For any S, π and L:

If $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L, then $\pi \models_L \psi$.

Step 1: From LTL Formulas to GNBAs – Correctness

Right-to-Left Implication of Correctness Thm. for Step 1. For any S, π and L:

If $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L, then $\pi \models_L \psi$.

Proof idea. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assuming $\mathcal{A}ut_\psi$ has an accepting run $K_0 K_1 K_2 \ldots$ for $A_0 A_1 A_2 \ldots$, we must show that $\pi \models_L \psi$.

Step 1: From LTL Formulas to GNBAs – Correctness

Right-to-Left Implication of Correctness Thm. for Step 1. For any S, π and L:

If $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L, then $\pi \models_L \psi$.

Proof idea. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assuming $\mathcal{A}ut_\psi$ has an accepting run $K_0 K_1 K_2 \ldots$ for $A_0 A_1 A_2 \ldots$, we must show that $\pi \models_L \psi$.

We can show something more general. Remember that being an accepting run means the following:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

(1.1) $K_i \in Q$, i.e., K_i is elementary; (1.2) $K_0 \in I$ (i.e., $\psi \in K_0$); (1.3) $K_i \xrightarrow{A_i} K_{i+1}$.

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$.

Step 1: From LTL Formulas to GNBAs – Correctness

Right-to-Left Implication of Correctness Thm. for Step 1. For any S, π and L:

If $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L, then $\pi \models_L \psi$.

Proof idea. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assuming $\mathcal{A}ut_\psi$ has an accepting run $K_0 K_1 K_2 \ldots$ for $A_0 A_1 A_2 \ldots$, we must show that $\pi \models_L \psi$.

We can show something more general. Remember that being an accepting run means the following:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

(1.1) $K_i \in Q$, i.e., K_i is elementary; (1.2) $K_0 \in I$ (i.e., $\psi \in K_0$); (1.3) $K_i \xrightarrow{A_i} K_{i+1}$.

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$.

Our generalization involves:

- replacing our fixed formula ψ with an arbitrary $\varphi \in K_0$
- renouncing the hypothesis (1.2) (of starting in an initial state)
- strengthening "$\varphi \in K_0$ implies $\pi \models_L \varphi$" to an "iff" statement, namely: (*) for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$

Step 1: From LTL Formulas to GNBAs – Correctness

Right-to-Left Implication of Correctness Thm. for Step 1. For any S, π and L:

If $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L, then $\pi \models_L \psi$.

Proof idea. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assuming $\mathcal{A}ut_\psi$ has an accepting run $K_0 K_1 K_2 \ldots$ for $A_0 A_1 A_2 \ldots$, we must show that $\pi \models_L \psi$.

We can show something more general. Remember that being an accepting run means the following:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

(1.1) $K_i \in Q$, i.e., K_i is elementary; (1.2) $K_0 \in I$ (i.e., $\psi \in K_0$); (1.3) $K_i \xrightarrow{A_i} K_{i+1}$.

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$.

Our generalization involves:

- replacing our fixed formula ψ with an arbitrary $\varphi \in K_0$
- renouncing the hypothesis (1.2) (of starting in an initial state)
- strengthening "$\varphi \in K_0$ implies $\pi \models_L \varphi$" to an "iff" statement, namely:
  (*) for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$

So we prove (1.1), (1.3) and (2) imply (*).

Step 1: From LTL Formulas to GNBAs – Correctness

Right-to-Left Implication of Correctness Thm. for Step 1. For any S, π and L:

If $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L, then $\pi \models_L \psi$.

Proof idea. Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assuming $\mathcal{A}ut_\psi$ has an accepting run $K_0 K_1 K_2 \ldots$ for $A_0 A_1 A_2 \ldots$, we must show that $\pi \models_L \psi$.

We can show something more general. Remember that being an accepting run means the following:

(1) $K_0 K_1 K_2 \ldots$ is a run, meaning:

(1.1) $K_i \in Q$, i.e., K_i is elementary; (1.2) $K_0 \in I$ (i.e., $\psi \in K_0$); (1.3) $K_i \xrightarrow{A_i} K_{i+1}$.

(2) $K_0 K_1 K_2 \ldots$ is accepting, meaning that it visits infinitely often the sets in $\mathcal{F}$.

Our generalization involves:

- replacing our fixed formula ψ with an arbitrary $\varphi \in K_0$
- renouncing the hypothesis (1.2) (of starting in an initial state)
- strengthening "$\varphi \in K_0$ implies $\pi \models_L \varphi$" to an "iff" statement, namely:
  (*) for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$

So we prove (1.1), (1.3) and (2) imply (*). $\boxed{\text{TYU: Why is this more general?}}$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ is an atom p. We have a chain of equivalent statements:

$$p \in K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \dots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \dots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ is an atom p. We have a chain of equivalent statements:

$$p \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$p \in A_0 = L(s_0)$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ is an atom p. We have a chain of equivalent statements:

$$p \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$p \in A_0 = L(s_0)$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L p$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ is an atom p. We have a chain of equivalent statements:

$$p \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$p \in A_0 = L(s_0)$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L p$$

This was an easy case.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\varphi_1 \wedge \varphi_2$. We have a chain of equivalent statements:

$$\varphi_1 \wedge \varphi_2 \in K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\varphi_1 \wedge \varphi_2$. We have a chain of equivalent statements:

$$\varphi_1 \wedge \varphi_2 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent)

$$\varphi_1 \in K_0 \text{ and } \varphi_2 \in K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\varphi_1 \wedge \varphi_2$. We have a chain of equivalent statements:

$$\varphi_1 \wedge \varphi_2 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent)

$$\varphi_1 \in K_0 \text{ and } \varphi_2 \in K_0$$

iff (by the induction hypothesis)

$$\pi \models_L \varphi_1 \text{ and } \pi \models_L \varphi_2$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\varphi_1 \wedge \varphi_2$. We have a chain of equivalent statements:

$$\varphi_1 \wedge \varphi_2 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent)

$$\varphi_1 \in K_0 \text{ and } \varphi_2 \in K_0$$

iff (by the induction hypothesis) TYU: OK to apply the induction hypothesis?

$$\pi \models_L \varphi_1 \text{ and } \pi \models_L \varphi_2$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \dots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \dots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\varphi_1 \wedge \varphi_2$. We have a chain of equivalent statements:

$$\varphi_1 \wedge \varphi_2 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent)

$$\varphi_1 \in K_0 \text{ and } \varphi_2 \in K_0$$

iff (by the induction hypothesis) TYU: OK to apply the induction hypothesis?

$$\pi \models_L \varphi_1 \text{ and } \pi \models_L \varphi_2$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \varphi_1 \wedge \varphi_2$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\varphi_1 \wedge \varphi_2$. We have a chain of equivalent statements:

$$\varphi_1 \wedge \varphi_2 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent)

$$\varphi_1 \in K_0 \text{ and } \varphi_2 \in K_0$$

iff (by the induction hypothesis) TYU: OK to apply the induction hypothesis?

$$\pi \models_L \varphi_1 \text{ and } \pi \models_L \varphi_2$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \varphi_1 \wedge \varphi_2$$

This case is entirely routine; and the same is true for all propositional connectives.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\neg\varphi_1$. We have a chain of equivalent statements:

$$\neg\varphi_1 \in K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\neg\varphi_1$. We have a chain of equivalent statements:

$$\neg\varphi_1 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent and complete)

$$\varphi_1 \notin K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\neg\varphi_1$. We have a chain of equivalent statements:

$$\neg\varphi_1 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent and complete)

$$\varphi_1 \notin K_0$$

iff (by the induction hypothesis)

$$\pi \not\models_L \varphi_1$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \dots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \dots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\neg\varphi_1$. We have a chain of equivalent statements:

$$\neg\varphi_1 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent and complete)

$$\varphi_1 \notin K_0$$

iff (by the induction hypothesis)

$$\pi \not\models_L \varphi_1$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \neg\varphi_1$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\neg\varphi_1$. We have a chain of equivalent statements:

$$\neg\varphi_1 \in K_0$$

iff (since K_0 is elementary, in particular propositionally consistent and complete)

$$\varphi_1 \notin K_0$$

iff (by the induction hypothesis)

$$\pi \not\models_L \varphi_1$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \neg\varphi_1$$

This case is also entirely routine; but only because the statement to be proved is strong enough! An "implies" instead of "iff" would not work.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc\varphi_1$.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \dots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \dots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc \varphi_1$. Let the atom-set trace $K'_0 K'_1 K'_2 \dots$ be defined as $K'_i = K_{i+1}$ for all $i \geq 0$. In other words, $K'_0 K'_1 K'_2 \dots$ is $K_1 K_2 K_3 \dots$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc\varphi_1$. Let the atom-set trace $K'_0 K'_1 K'_2 \ldots$ be defined as $K'_i = K_{i+1}$ for all $i \geq 0$. In other words, $K'_0 K'_1 K'_2 \ldots$ is $K_1 K_2 K_3 \ldots$ We have:

$$\bigcirc\varphi_1 \in K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc\varphi_1$. Let the atom-set trace $K_0' K_1' K_2' \ldots$ be defined as $K_i' = K_{i+1}$ for all $i \geq 0$. In other words, $K_0' K_1' K_2' \ldots$ is $K_1 K_2 K_3 \ldots$ We have:

$$\bigcirc\varphi_1 \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$\varphi_1 \in K_1, \text{ i.e., } \varphi_1 \in K_0'$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \dots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \dots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc\varphi_1$. Let the atom-set trace $K'_0 K'_1 K'_2 \dots$ be defined as $K'_i = K_{i+1}$ for all $i \geq 0$. In other words, $K'_0 K'_1 K'_2 \dots$ is $K_1 K_2 K_3 \dots$ We have:

$$\bigcirc\varphi_1 \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$\varphi_1 \in K_1, \text{ i.e., } \varphi_1 \in K'_0$$

iff (by the induction hypothesis applied to $K'_0 K'_1 K'_2 \dots$ and π^1)

$$\pi^1 \models_L \varphi_1$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc\varphi_1$. Let the atom-set trace $K'_0 K'_1 K'_2 \ldots$ be defined as $K'_i = K_{i+1}$ for all $i \geq 0$. In other words, $K'_0 K'_1 K'_2 \ldots$ is $K_1 K_2 K_3 \ldots$ We have:

$$\bigcirc\varphi_1 \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$\varphi_1 \in K_1, \text{ i.e., } \varphi_1 \in K'_0$$

iff (by the induction hypothesis applied to $K'_0 K'_1 K'_2 \ldots$ and π^1)

$$\pi^1 \models_L \varphi_1$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \bigcirc\varphi_1$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\bigcirc \varphi_1$. Let the atom-set trace $K'_0 K'_1 K'_2 \ldots$ be defined as $K'_i = K_{i+1}$ for all $i \geq 0$. In other words, $K'_0 K'_1 K'_2 \ldots$ is $K_1 K_2 K_3 \ldots$ We have:

$$\bigcirc \varphi_1 \in K_0$$

iff (by the definition of $\mathcal{A}ut_\psi$'s transition relation $\rightarrow$)

$$\varphi_1 \in K_1, \text{ i.e., } \varphi_1 \in K'_0$$

iff (by the induction hypothesis applied to $K'_0 K'_1 K'_2 \ldots$ and π^1)

$$\pi^1 \models_L \varphi_1$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \bigcirc \varphi_1$$

This case required applying the induction hypothesis not to $K_0 K_1 K_2 \ldots$ and π, but to their shifted versions $K_1 K_2 K_3 \ldots$ and π^1.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond \varphi_1$.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond \varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond\varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$ We have:

$$\Diamond\varphi_1 \in K_0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond\varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$ We have:

$$\Diamond\varphi_1 \in K_0$$

iff (by a lemma)

$$\varphi_1 \in K_j, \text{ i.e., } \varphi_1 \in K_0^j \text{ for some } j \geq 0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond\varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$ We have:

$$\Diamond\varphi_1 \in K_0$$

iff (by a lemma)

$$\varphi_1 \in K_j, \text{ i.e., } \varphi_1 \in K_0^j \text{ for some } j \geq 0$$

iff (by the induction hypothesis applied to $K_0^j K_1^j K_2^j \ldots$ and π^j)

$$\pi^j \models_L \varphi_1 \text{ for some } j \geq 0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond\varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$ We have:

$\Diamond\varphi_1 \in K_0$

iff (by a lemma)

$\varphi_1 \in K_j$, i.e., $\varphi_1 \in K_0^j$ for some $j \geq 0$

iff (by the induction hypothesis applied to $K_0^j K_1^j K_2^j \ldots$ and π^j)

$\pi^j \models_L \varphi_1$ for some $j \geq 0$

iff (by the definition of the satisfaction relation)

$\pi \models_L \Diamond\varphi_1$

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond\varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$ We have:

$$\Diamond\varphi_1 \in K_0$$

iff (by a lemma)

$$\varphi_1 \in K_j,\text{ i.e., } \varphi_1 \in K_0^j \text{ for some } j \geq 0$$

iff (by the induction hypothesis applied to $K_0^j K_1^j K_2^j \ldots$ and π^j)

$$\pi^j \models_L \varphi_1 \text{ for some } j \geq 0$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \Diamond\varphi_1$$

This case required applying the induction hypothesis not to $K_0 K_1 K_2 \ldots$ and π, but to their j-shifted versions $K_j K_{j+1} K_{j+2} \ldots$ and π^j.

Step 1: From LTL Formulas to GNBAs – Correctness

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ. Some representative cases:

Assume φ has the form $\Diamond\varphi_1$. Let, for each $j \geq 0$, the atom-set trace $K_0^j K_1^j K_2^j \ldots$ be defined as $K_i^j = K_{i+j}$ for all $i \geq 0$. I.e., $K_0^j K_1^j K_2^j \ldots$ is $K_j K_{j+1} K_{j+2} \ldots$ We have:

$$\Diamond\varphi_1 \in K_0$$

iff (by a lemma)

$$\varphi_1 \in K_j, \text{ i.e., } \varphi_1 \in K_0^j \text{ for some } j \geq 0$$

iff (by the induction hypothesis applied to $K_0^j K_1^j K_2^j \ldots$ and π^j)

$$\pi^j \models_L \varphi_1 \text{ for some } j \geq 0$$

iff (by the definition of the satisfaction relation)

$$\pi \models_L \Diamond\varphi_1$$

This case required applying the induction hypothesis not to $K_0 K_1 K_2 \ldots$ and π, but to their j-shifted versions $K_j K_{j+1} K_{j+2} \ldots$ and π^j. CYA: Anything missing in this proof?

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) $[\varphi \notin K_0$ and $\Diamond\varphi \in K_1]$

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) $[\varphi \notin K_0$ and $\Diamond\varphi \in K_1]$

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the "unfinished business" situation.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) [$\varphi \notin K_0$ and $\Diamond\varphi \in K_1$]

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the "unfinished business" situation.
- If case (2) holds, then either (1) $\varphi \in K_1$ or (2) [$\varphi \notin K_1$ and $\Diamond\varphi \in K_2$]

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) [$\varphi \notin K_0$ and $\Diamond\varphi \in K_1$]

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the "unfinished business" situation.

- If case (2) holds, then either (1) $\varphi \in K_1$ or (2) [$\varphi \notin K_1$ and $\Diamond\varphi \in K_2$]
- If case (2) holds again, then either (1) $\varphi \in K_2$ or (2) [$\varphi \notin K_2$ and $\Diamond\varphi \in K_3$]

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) $[\varphi \notin K_0$ and $\Diamond\varphi \in K_1]$

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the "unfinished business" situation.
- If case (2) holds, then either (1) $\varphi \in K_1$ or (2) $[\varphi \notin K_1$ and $\Diamond\varphi \in K_2]$
- If case (2) holds again, then either (1) $\varphi \in K_2$ or (2) $[\varphi \notin K_2$ and $\Diamond\varphi \in K_3]$
- And so on.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) $[\varphi \notin K_0$ and $\Diamond\varphi \in K_1]$

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the "unfinished business" situation.

- If case (2) holds, then either (1) $\varphi \in K_1$ or (2) $[\varphi \notin K_1$ and $\Diamond\varphi \in K_2]$
- If case (2) holds again, then either (1) $\varphi \in K_2$ or (2) $[\varphi \notin K_2$ and $\Diamond\varphi \in K_3]$
- And so on.

Moreover, $K_0 K_1 K_2 \ldots$ is an accepting run in $\mathcal{A}ut_\psi$, which means that infinitely often for $j \geq 0$, $\Diamond\varphi \in K_j$ implies $\varphi \in K_j$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \rightarrow \ldots$, from the definition of $\rightarrow$ we have that:
- Either (1) $\varphi \in K_0$ or (2) $[\varphi \notin K_0$ and $\Diamond\varphi \in K_1]$

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the "unfinished business" situation.

- If case (2) holds, then either (1) $\varphi \in K_1$ or (2) $[\varphi \notin K_1$ and $\Diamond\varphi \in K_2]$
- If case (2) holds again, then either (1) $\varphi \in K_2$ or (2) $[\varphi \notin K_2$ and $\Diamond\varphi \in K_3]$
- And so on.

Moreover, $K_0 K_1 K_2 \ldots$ is an accepting run in $\mathcal{A}ut_\psi$, which means that infinitely often for $j \geq 0$, $\Diamond\varphi \in K_j$ implies $\varphi \in K_j$.
So case (2) cannot hold infinitely, meaning that case (1) will hold at some point j.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the left-to-right direction, assume $\Diamond\varphi \in K_0$.
Since $K_0 \xrightarrow{A_0} K_1 \xrightarrow{A_1} K_2 \to \ldots$, from the definition of $\to$ we have that:
- Either (1) $\varphi \in K_0$ or (2) $[\varphi \notin K_0$ and $\Diamond\varphi \in K_1]$

Case (1) means fulfilling the eventuality, whereas case (2) means postponing it to next time – remember the “unfinished business” situation.
- If case (2) holds, then either (1) $\varphi \in K_1$ or (2) $[\varphi \notin K_1$ and $\Diamond\varphi \in K_2]$
- If case (2) holds again, then either (1) $\varphi \in K_2$ or (2) $[\varphi \notin K_2$ and $\Diamond\varphi \in K_3]$
- And so on.

Moreover, $K_0 K_1 K_2 \ldots$ is an accepting run in $\mathcal{A}ut_\psi$, which means that infinitely often for $j \geq 0$, $\Diamond\varphi \in K_j$ implies $\varphi \in K_j$.
So case (2) cannot hold infinitely, meaning that case (1) will hold at some point j.
We thus obtain j such that $\varphi \in K_j$, as desired.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.
We have two cases:
Case 1: $j = 0$. Then we are done, since $\Diamond\varphi \in K_0$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.
We have two cases:
Case 1: $j = 0$. Then we are done, since $\Diamond\varphi \in K_0$.
Case 2: $j > 0$. Let $j' = j - 1$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.
We have two cases:
Case 1: $j = 0$. Then we are done, since $\Diamond\varphi \in K_0$.
Case 2: $j > 0$. Let $j' = j - 1$.
Since $\Diamond\varphi \in K_j$ and $K_{j'} \xrightarrow{A_{j'}} K_j$, from the definition of $\rightarrow$ we have that $\Diamond\varphi \in K_{j'}$.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.
We have two cases:
Case 1: $j = 0$. Then we are done, since $\Diamond\varphi \in K_0$.
Case 2: $j > 0$. Let $j' = j - 1$.
Since $\Diamond\varphi \in K_j$ and $K_{j'} \xrightarrow{A_{j'}} K_j$, from the definition of $\rightarrow$ we have that $\Diamond\varphi \in K_{j'}$.
And we continue the same reasoning for j' instead of j.

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.
We have two cases:
Case 1: $j = 0$. Then we are done, since $\Diamond\varphi \in K_0$.
Case 2: $j > 0$. Let $j' = j - 1$.
Since $\Diamond\varphi \in K_j$ and $K_{j'} \xrightarrow{A_{j'}} K_j$, from the definition of $\rightarrow$ we have that $\Diamond\varphi \in K_{j'}$.
And we continue the same reasoning for j' instead of j.

At some point, case 1 must hold, since j keeps decreasing. (Strictly speaking, this is an induction on j.)

Step 1: From LTL Formulas to GNBAs – Correctness

Lemma: For all φ such that $\Diamond\varphi \in Cl(\psi)$, we have

$$\Diamond\varphi \in K_0 \quad \text{iff} \quad \varphi \in K_j \text{ for some } j \geq 0$$

Proof idea.

For the right–to-left direction, assume $\varphi \in K_j$ for some $j \geq 0$.
Since K_j is elementary, in particular temporally consistent, we also have $\Diamond\varphi \in K_j$.
We have two cases:
Case 1: $j = 0$. Then we are done, since $\Diamond\varphi \in K_0$.
Case 2: $j > 0$. Let $j' = j - 1$.
Since $\Diamond\varphi \in K_j$ and $K_{j'} \xrightarrow{A_{j'}} K_j$, from the definition of $\rightarrow$ we have that $\Diamond\varphi \in K_{j'}$.
And we continue the same reasoning for j' instead of j.

At some point, case 1 must hold, since j keeps decreasing. (Strictly speaking, this is an induction on j.)
So we obtain $\Diamond\varphi \in K_0$, as desired.

Homework Exercise

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ.

Homework Exercise

Assume $\pi = s_0 s_1 s_2 \ldots$ and let $A_i = L(s_i)$ for all $i \geq 0$. Assume:

(1.1) K_i is elementary; (1.3) $K_i \xrightarrow{A_i} K_{i+1}$;

(2) $K_0 K_1 K_2 \ldots$ visits infinitely often the sets in $\mathcal{F}$.

We must show: for all $\varphi \in Cl(\psi)$, we have $\varphi \in K_0$ iff $\pi \models_L \varphi$.

The proof goes by induction on the structure of φ.

Do the proofs for the remaining cases:

Assume φ has the form $\varphi_1 \vee \varphi_2$. ... Routine

Assume φ has the form $\varphi_1 \rightarrow \varphi_2$. ... Routine

Assume φ has the form $\Box\varphi_1$. ... Interesting. You will need a lemma like for $\Diamond$.

Assume φ has the form $\varphi_1 \,\mathsf{U}\, \varphi_2$. ... Interesting. You will need a lemma like for $\Diamond$.

Summary and Outlook

For any formula ψ, we defined the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$.

We proved the following:

Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \rightarrow \mathcal{P}(Atom)$

$$\pi \models_L \psi \text{ iff } \mathcal{A}ut_\psi \text{ accepts the atom-set trace of } \pi \text{ through } L.$$

Summary and Outlook

For any formula ψ, we defined the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$.

We proved the following:

Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \rightarrow \mathcal{P}(Atom)$

$\pi \models_L \psi$ iff $\mathcal{A}ut_\psi$ accepts the atom-set trace of π through L.

We can say that automaton $\mathcal{A}ut_\psi$ mimics, or simulates, or encodes, the semantic behavior of ψ.

Summary and Outlook

For any formula ψ, we defined the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$.

We proved the following:

Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \rightarrow \mathcal{P}(Atom)$

$$\pi \models_L \psi \text{ iff } \mathcal{A}ut_\psi \text{ accepts the atom-set trace of } \pi \text{ through } L.$$

We can say that automaton $\mathcal{A}ut_\psi$ mimics, or simulates, or encodes, the semantic behavior of ψ.

Next, we look into how to encode satisfaction of a formula by an LTS (in a state) using GNBAs – this is Step 2.

Summary and Outlook

For any formula ψ, we defined the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$.

We proved the following:

Correctness Theorem for Step 1. For any set of states S, infinite sequence of states π and labeling functions $L : S \rightarrow \mathcal{P}(Atom)$

$$\pi \models_L \psi \text{ iff } \mathcal{A}ut_\psi \text{ accepts the atom-set trace of } \pi \text{ through } L.$$

We can say that automaton $\mathcal{A}ut_\psi$ mimics, or simulates, or encodes, the semantic behavior of ψ.

Next, we look into how to encode satisfaction of a formula by an LTS (in a state) using GNBAs – this is Step 2.

Finally, we will look into how to algorithmically decide satisfaction, once encoded – this is Step 3.

Homework Exercise

Describe the automaton $\mathcal{A}ut_\psi$ in the following cases:

- $Atoms = \{a\}$ and $\psi = \Box a$.
- $Atoms = \{a, b\}$ and $\psi = a \mathsf{U} b$
- $Atoms = \{a, b\}$ and $\psi = \Diamond(a \wedge b)$

Step 2: Product GNBA

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_{\psi} = (\Sigma, Q, I, \rightarrow, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. Remember that $\to \subseteq S \times S$ and $L : S \to \mathcal{P}(Atoms)$.

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. Remember that $\to \subseteq S \times S$ and $L : S \to \mathcal{P}(Atoms)$.

Note: We write $\to$ for both the transition relation $\to \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\to \subseteq S \times S$ of $\mathcal{M}$.

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. Remember that $\rightarrow \subseteq S \times S$ and $L : S \rightarrow \mathcal{P}(Atoms)$.

Note: We write $\rightarrow$ for both the transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\rightarrow \subseteq S \times S$ of $\mathcal{M}$.

We define the product of $(\mathcal{M}, s_0)$ and $\mathcal{A}ut_\psi$ to be the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ whose components are as follows:

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. Remember that $\rightarrow \subseteq S \times S$ and $L : S \rightarrow \mathcal{P}(Atoms)$.

Note: We write $\rightarrow$ for both the transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\rightarrow \subseteq S \times S$ of $\mathcal{M}$.

We define the product of $(\mathcal{M}, s_0)$ and $\mathcal{A}ut_\psi$ to be the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ whose components are as follows:

$Q_\times = \{(s, K) \mid s \in S,\ K \in Q \text{ and } L(s) = Atoms \cap K\}$

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. Remember that $\to \subseteq S \times S$ and $L : S \to \mathcal{P}(Atoms)$.

Note: We write $\to$ for both the transition relation $\to \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\to \subseteq S \times S$ of $\mathcal{M}$.

We define the product of $(\mathcal{M}, s_0)$ and $\mathcal{A}ut_\psi$ to be the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$ whose components are as follows:

$Q_\times = \{(s, K) \mid s \in S,\ K \in Q \text{ and } L(s) = Atoms \cap K\}$

$I_\times = \{(s_0, K) \mid (s_0, K) \in Q_\times \text{ and } K \in I\}$

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. Remember that $\rightarrow \subseteq S \times S$ and $L : S \rightarrow \mathcal{P}(Atoms)$.

Note: We write $\rightarrow$ for both the transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\rightarrow \subseteq S \times S$ of $\mathcal{M}$.

We define the product of $(\mathcal{M}, s_0)$ and $\mathcal{A}ut_\psi$ to be the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ whose components are as follows:

$Q_\times = \{(s, K) \mid s \in S,\ K \in Q \text{ and } L(s) = Atoms \cap K\}$

$I_\times = \{(s_0, K) \mid (s_0, K) \in Q_\times \text{ and } K \in I\}$

$(s, K) \xrightarrow{A}_\times (s', K')$ iff $s \rightarrow s'$ and $K \xrightarrow{A} K'$

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \rightarrow, L)$ be an LTS and $s_0 \in S$. Remember that $\rightarrow \subseteq S \times S$ and $L : S \rightarrow \mathcal{P}(Atoms)$.

Note: We write $\rightarrow$ for both the transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\rightarrow \subseteq S \times S$ of $\mathcal{M}$.

We define the product of $(\mathcal{M}, s_0)$ and $\mathcal{A}ut_\psi$ to be the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ whose components are as follows:

$Q_\times = \{(s, K) \mid s \in S,\ K \in Q \text{ and } L(s) = Atoms \cap K\}$

$I_\times = \{(s_0, K) \mid (s_0, K) \in Q_\times \text{ and } K \in I\}$

$(s, K) \xrightarrow{A}_\times (s', K')$ iff $s \rightarrow s'$ and $K \xrightarrow{A} K'$ (Note: $L(s) = Atoms \cap K = A$)

Step 2: Product GNBA – Definition

So we have $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$, where $\Sigma = \mathcal{P}(Atoms)$.

Side note: The next construction works not only for $\mathcal{A}ut_\psi$, but for any GNBA whose alphabet is $\mathcal{P}(Atoms)$.

Let $\mathcal{M} = (S, \to, L)$ be an LTS and $s_0 \in S$. Remember that $\to \subseteq S \times S$ and $L : S \to \mathcal{P}(Atoms)$.

Note: We write $\to$ for both the transition relation $\to \subseteq Q \times \Sigma \times Q$ of $\mathcal{A}ut_\psi$ and the transition relation $\to \subseteq S \times S$ of $\mathcal{M}$.

We define the product of $(\mathcal{M}, s_0)$ and $\mathcal{A}ut_\psi$ to be the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$ whose components are as follows:

$Q_\times = \{(s, K) \mid s \in S,\ K \in Q \text{ and } L(s) = Atoms \cap K\}$

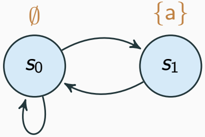
$I_\times = \{(s_0, K) \mid (s_0, K) \in Q_\times \text{ and } K \in I\}$

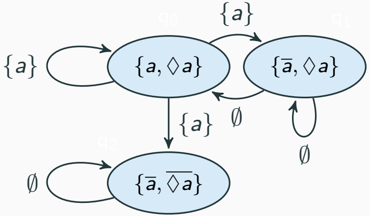
$(s, K) \xrightarrow{A}_\times (s', K')$ iff $s \to s'$ and $K \xrightarrow{A} K'$ (Note: $L(s) = Atoms \cap K = A$)

$\mathcal{F}_\times = \{\ \{(s, K) \mid (s, K) \in Q \text{ and } K \in F\} \mid F \in \mathcal{F}\ \}$

Running Example (Continued)

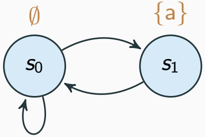
Consider the LTS $\mathcal{M} = (S, \rightarrow, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ has set of states Q and transition relation $\rightarrow$ shown in the picture on the right. Also,
$I = \{\, \{a, \Diamond a\},\ \{\overline{a}, \Diamond a\}\,\}$ and $\mathcal{F} = \{\ \{\, \{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\}\,\}\ \}$.

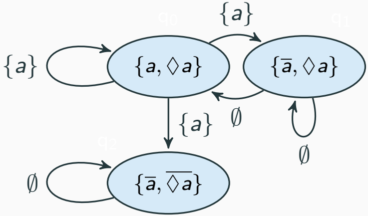




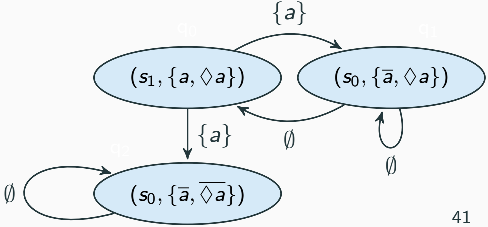
Running Example (Continued)

Consider the LTS $\mathcal{M} = (S, \to, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ has set of states Q and transition relation $\to$ shown in the picture on the right. Also,
$I = \{\, \{a, \Diamond a\},\ \{\overline{a}, \Diamond a\} \,\}$ and $\mathcal{F} = \{\ \{\, \{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\} \,\}\ \}$.



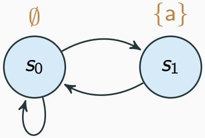


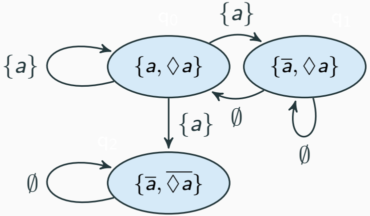
The product GNBA
$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$
has $Q_\times$ and $\to_\times$ shown on the right



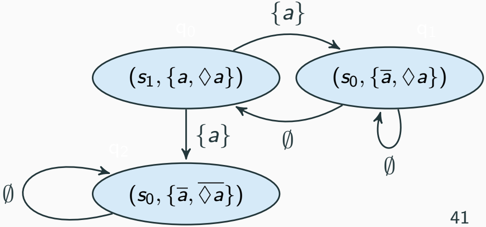
Running Example (Continued)

Consider the LTS $\mathcal{M} = (S, \rightarrow, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ has set of states Q and transition relation $\rightarrow$ shown in the picture on the right. Also,
$I = \{\,\{a, \Diamond a\},\ \{\overline{a}, \Diamond a\}\,\}$ and $\mathcal{F} = \{\ \{\,\{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\}\,\}\ \}$.



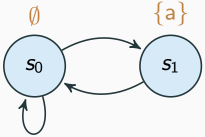


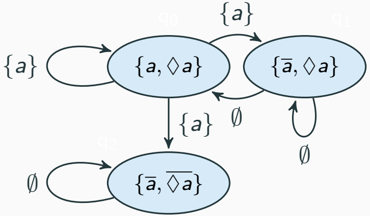
The product GNBA
$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$
has $Q_\times$ and $\rightarrow_\times$ shown on the right,
and has $I_\times = \{\,(s_0, \{\overline{a}, \Diamond a\})\,\}$ and
$\mathcal{F}_\times = \{\ \{\,(s_1, \{a, \Diamond a\}),\ (s_0, \{\overline{a}, \overline{\Diamond a}\})\,\}\ \}$



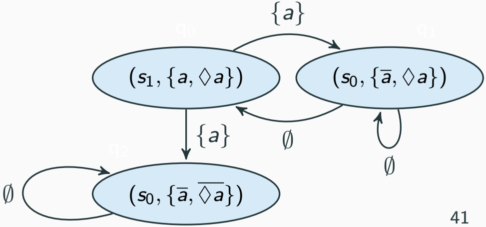
Running Example (Continued)

Consider the LTS $\mathcal{M} = (S, \to, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ has set of states Q and transition relation $\to$ shown in the picture on the right. Also,
$I = \{\, \{a, \Diamond a\},\ \{\overline{a}, \Diamond a\} \,\}$ and $\mathcal{F} = \{\ \{\, \{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\} \,\}\ \}$.



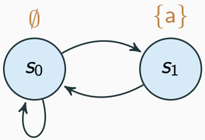


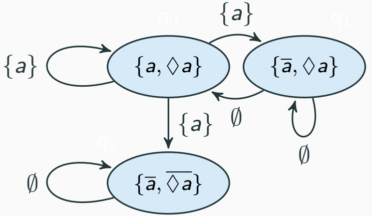
The product GNBA
$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$
has $Q_\times$ and $\to_\times$ shown on the right,
and has $I_\times = \{\, (s_0, \{\overline{a}, \Diamond a\}) \,\}$ and
$\mathcal{F}_\times = \{\ \{\, (s_1, \{a, \Diamond a\}),\ (s_0, \{\overline{a}, \overline{\Diamond a}\}) \,\}\ \}$
E.g., $Q_\times$ contains $(s_1, \{a, \Diamond a\})$
since $L(s_1) = \{a\} = \{a, \Diamond a\} \cap Atoms$



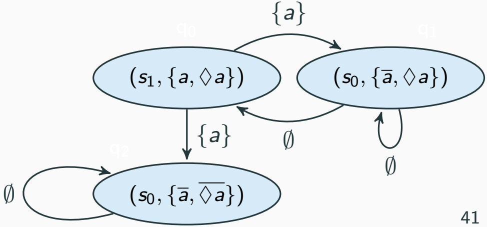
Running Example (Continued)

Consider the LTS $\mathcal{M} = (S, \to, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ has set of states Q and transition relation $\to$ shown in the picture on the right. Also,
$I = \{\,\{a, \Diamond a\},\ \{\overline{a}, \Diamond a\}\,\}$ and $\mathcal{F} = \{\ \{\,\{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\}\,\}\ \}$.



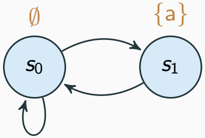


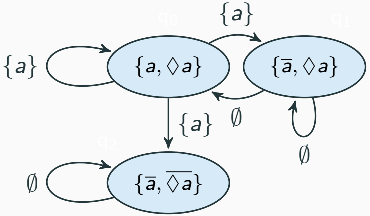
The product GNBA
$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$
has $Q_\times$ and $\to_\times$ shown on the right,
and has $I_\times = \{\,(s_0, \{\overline{a}, \Diamond a\})\,\}$ and
$\mathcal{F}_\times = \{\ \{\,(s_1, \{a, \Diamond a\}),\ (s_0, \{\overline{a}, \overline{\Diamond a}\})\,\}\ \}$
E.g., $Q_\times$ does not contain $(s_1, \{\overline{a}, \Diamond a\})$
since $L(s_1) \neq \emptyset = \{\overline{a}, \Diamond a\} \cap Atoms$



Running Example (Continued)

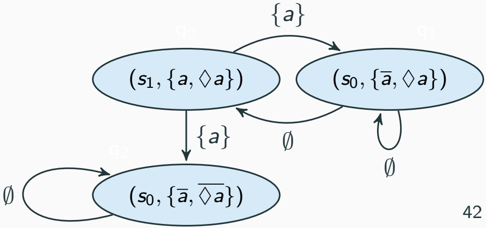
Consider the LTS $\mathcal{M} = (S, \rightarrow, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ has set of states Q and transition relation $\rightarrow$ shown in the picture on the right. Also,
$I = \{\, \{a, \Diamond a\},\ \{\overline{a}, \Diamond a\}\, \}$ and $\mathcal{F} = \{\ \{\, \{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\}\, \}\ \}$.





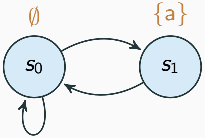
The product GNBA
$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$
has $Q_\times$ and $\rightarrow_\times$ shown to the left,
and has $I_\times = \{\, (s_0, \{\overline{a}, \Diamond a\})\, \}$ and
$\mathcal{F}_\times = \{\ \{\, (s_1, \{a, \Diamond a\}),\ (s_0, \{\overline{a}, \overline{\Diamond a}\})\, \}\ \}$

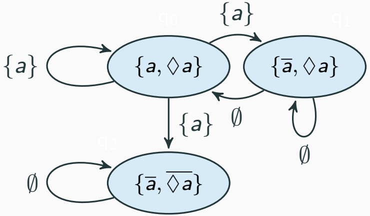
E.g., $(s_1, \{a, \Diamond a\}) \xrightarrow{\{a\}}_\times (s_0, \{\overline{a}, \Diamond a\})$
since $s_1 \rightarrow s_0$ and $\{a, \Diamond a\} \xrightarrow{\{a\}} \{\overline{a}, \Diamond a\}$



Running Example (Continued)

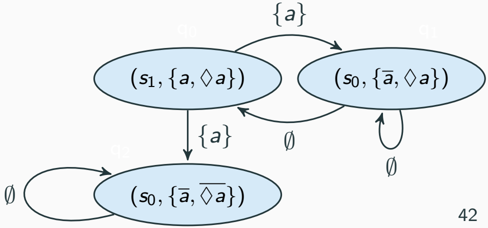
Consider the LTS $\mathcal{M} = (S, \rightarrow, L)$ shown in the picture on the left.
Remember that, taking ψ to be $\Diamond a$, the GNBA $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ has set of states Q and transition relation $\rightarrow$ shown in the picture on the right. Also,
$I = \{\, \{a, \Diamond a\},\ \{\bar{a}, \Diamond a\} \,\}$ and $\mathcal{F} = \{\ \{\, \{a, \Diamond a\},\ \{\bar{a}, \overline{\Diamond a}\} \,\}\ \}$.





The product GNBA
$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$
has $Q_\times$ and $\rightarrow_\times$ shown to the left,
and has $I_\times = \{\, (s_0, \{\bar{a}, \Diamond a\}) \,\}$ and
$\mathcal{F}_\times = \{\ \{\, (s_1, \{a, \Diamond a\}),\ (s_0, \{\bar{a}, \overline{\Diamond a}\}) \,\}\ \}$

E.g., not $(s_1, \{a, \Diamond a\}) \xrightarrow{\{a\}}_\times (s_1, \{a, \Diamond a\})$
since $s_1 \rightarrow s_1$ does not hold



Step 2: Product GNBA – Correctness

Context: $\mathcal{M} = (S, \rightarrow, L)$ is an LTS, $s_0 \in S$, and $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ is the GNBA of an LTL formula ψ.

Step 2: Product GNBA – Correctness

Context: $\mathcal{M} = (S, \rightarrow, L)$ is an LTS, $s_0 \in S$, and $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ is the GNBA of an LTL formula ψ.

We have defined the product GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$.

Step 2: Product GNBA – Correctness

Context: $\mathcal{M} = (S, \rightarrow, L)$ is an LTS, $s_0 \in S$, and $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ is the GNBA of an LTL formula ψ.

We have defined the product GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$.

Correctness Theorem for Step 2. Let $A_0 A_1 A_2 \ldots$ be an infinite sequence of atom sets. Then

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$
iff
there exists $\pi \in Paths_{s_0}(\mathcal{M})$ such that $A_0 A_1 A_2 \ldots$ is the atom-set trace of π through L and $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$

Step 2: Product GNBA – Correctness

Context: $\mathcal{M} = (S, \to, L)$ is an LTS, $s_0 \in S$, and $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ is the GNBA of an LTL formula ψ.

We have defined the product GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$.

Correctness Theorem for Step 2. Let $A_0 A_1 A_2 \ldots$ be an infinite sequence of atom sets. Then

$$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi \text{ accepts } A_0 A_1 A_2 \ldots$$
$$\text{iff}$$
$$\text{there exists } \pi \in Paths_{s_0}(\mathcal{M}) \text{ such that } A_0 A_1 A_2 \ldots \text{ is the}$$
$$\text{atom-set trace of } \pi \text{ through } L \text{ and } \mathcal{A}ut_\psi \text{ accepts } A_0 A_1 A_2 \ldots$$

This has a routine proof, applying the definition of the product automaton.

Step 2: Product GNBA – Correctness

Proof. We have the following chain of equivalent statements:

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$

Step 2: Product GNBA – Correctness

Proof. We have the following chain of equivalent statements:

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0A_1A_2\ldots$

iff

There exists in $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ an accepting run $(s_0, K_0)(s_1, K_1)(s_2, K_2)\ldots$ for $A_0A_1A_2\ldots$

Step 2: Product GNBA – Correctness

Proof. We have the following chain of equivalent statements:

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$

iff

There exists in $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ an accepting run $(s_0, K_0)(s_1, K_1)(s_2, K_2)\ldots$ for $A_0 A_1 A_2 \ldots$

iff (by the definition of accepting runs and of $Q_\times$, $I_\times$ and $\rightarrow_\times$)

There exist $s_0 s_1 s_2 \ldots$ and $K_0 K_1 K_2 \ldots$ such that: $K_0 \in I$,
for all $i \geq 0$: $A_i = L(s_i)$, $s_i \rightarrow s_{i+1}$ and $K_i \xrightarrow{A_i} K_{i+1}$
and for all $G \in \mathcal{F}_\times$, we have $(s_i, K_i) \in G$ for infinitely many $i \geq 0$

Step 2: Product GNBA – Correctness

Proof. We have the following chain of equivalent statements:

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$

iff

There exists in $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ an accepting run $(s_0, K_0)(s_1, K_1)(s_2, K_2) \ldots$ for $A_0 A_1 A_2 \ldots$

iff (by the definition of accepting runs and of $Q_\times$, $I_\times$ and $\rightarrow_\times$)

There exist $s_0 s_1 s_2 \ldots$ and $K_0 K_1 K_2 \ldots$ such that: $K_0 \in I$,
for all $i \geq 0$: $A_i = L(s_i)$, $s_i \rightarrow s_{i+1}$ and $K_i \xrightarrow{A_i} K_{i+1}$
and for all $G \in \mathcal{F}_\times$, we have $(s_i, K_i) \in G$ for infinitely many $i \geq 0$

iff (by the definition of $\mathcal{F}_\times$)

There exist $s_0 s_1 s_2 \ldots$ and $K_0 K_1 K_2 \ldots$ such that: $K_0 \in I$,
for all $i \geq 0$: $A_i = L(s_i)$, $s_i \rightarrow s_{i+1}$ and $K_i \xrightarrow{A_i} K_{i+1}$
and for all $F \in \mathcal{F}$, we have $K_i \in F$ for infinitely many $i \geq 0$

Step 2: Product GNBA – Correctness

Proof. We have the following chain of equivalent statements:

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$

iff

There exists in $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ an accepting run $(s_0, K_0)(s_1, K_1)(s_2, K_2)\ldots$ for $A_0 A_1 A_2 \ldots$

iff (by the definition of accepting runs and of $Q_\times$, $I_\times$ and $\rightarrow_\times$)

There exist $s_0 s_1 s_2 \ldots$ and $K_0 K_1 K_2 \ldots$ such that: $K_0 \in I$,
for all $i \geq 0$: $A_i = L(s_i)$, $s_i \rightarrow s_{i+1}$ and $K_i \xrightarrow{A_i} K_{i+1}$
and for all $G \in \mathcal{F}_\times$, we have $(s_i, K_i) \in G$ for infinitely many $i \geq 0$

iff (by the definition of $\mathcal{F}_\times$)

There exist $s_0 s_1 s_2 \ldots$ and $K_0 K_1 K_2 \ldots$ such that: $K_0 \in I$,
for all $i \geq 0$: $A_i = L(s_i)$, $s_i \rightarrow s_{i+1}$ and $K_i \xrightarrow{A_i} K_{i+1}$
and for all $F \in \mathcal{F}$, we have $K_i \in F$ for infinitely many $i \geq 0$

iff (by the definition of accepting runs, of paths and of "atom-set trace of")

There exist $\pi = s_0 s_1 s_2 \ldots \in Paths_{s_0}(\mathcal{M})$ and $K_0 K_1 K_2 \ldots$ such that:
$A_0 A_1 A_2 \ldots$ is the atom-set trace of π through L and
$K_0 K_1 K_2 \ldots$ is an accepting run (in $\mathcal{A}ut_\psi$) for $A_0 A_1 A_2 \ldots$

Step 2: Product GNBA – Correctness

...

iff (by logic)

There exists $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that:

$A_0 A_1 A_2 \ldots$ is the atom-set trace of π through L and

there exists an accepting run (in $\mathcal{A}ut_{\psi}$) $K_0 K_1 K_2 \ldots$ for $A_0 A_1 A_2 \ldots$

Step 2: Product GNBA – Correctness

...

iff (by logic)

There exists $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that:

$A_0A_1A_2\ldots$ is the atom-set trace of π through L and

there exists an accepting run (in $\mathcal{A}ut_\psi$) $K_0K_1K_2\ldots$ for $A_0A_1A_2\ldots$

iff

There exist $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that:

$A_0A_1A_2\ldots$ is the atom-set trace of π through L and

$\mathcal{A}ut_\psi$ accepts $A_0A_1A_2\ldots$

Step 2: Product GNBA – Correctness

Context: $\mathcal{M} = (S, \to, L)$ is an LTS, $s_0 \in S$, and $\mathcal{A}ut_\psi = (\Sigma, Q, I, \to, \mathcal{F})$ is the GNBA of an LTL formula ψ.

We have defined the product GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \to_\times, \mathcal{F}_\times)$.

Correctness Theorem for Step 2. Let $A_0A_1A_2\ldots$ be an infinite sequence of atom sets. Then

$$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi \text{ accepts } A_0A_1A_2\ldots$$

iff

there exists $\pi \in Paths_{s_0}(\mathcal{M})$ such that $A_0A_1A_2\ldots$ is the atom-set trace of π through L and $\mathcal{A}ut_\psi$ accepts $A_0A_1A_2\ldots$

Step 2: Product GNBA – Correctness

Context: $\mathcal{M} = (S, \rightarrow, L)$ is an LTS, $s_0 \in S$, and $\mathcal{A}ut_\psi = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ is the GNBA of an LTL formula ψ.

We have defined the product GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$.

Correctness Theorem for Step 2. Let $A_0 A_1 A_2 \ldots$ be an infinite sequence of atom sets. Then

$(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$
iff
there exists $\pi \in Paths_{s_0}(\mathcal{M})$ such that $A_0 A_1 A_2 \ldots$ is the atom-set trace of π through L and $\mathcal{A}ut_\psi$ accepts $A_0 A_1 A_2 \ldots$

Corollary.

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_\psi$ is empty
iff
there exists no $\pi \in Paths_{s_0}(\mathcal{M})$ such that the atom-set trace of π through L is accepted by $\mathcal{A}ut_\psi$.

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \rightarrow, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \to, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Proof. We have the following chain of equivalent statements:

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \to, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Proof. We have the following chain of equivalent statements:

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty
iff (by the corollary of the Correctness Theorem for Step 2)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\mathcal{A}ut_{\neg\varphi}$ accepts its atom-set trace through L

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \rightarrow, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Proof. We have the following chain of equivalent statements:

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty
  iff (by the corollary of the Correctness Theorem for Step 2)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\mathcal{A}ut_{\neg\varphi}$ accepts its atom-set trace through L
  iff (by the Correctness Theorem for Step 1)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\pi \models_L \neg\varphi$

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \rightarrow, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Proof. We have the following chain of equivalent statements:

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty
  iff (by the corollary of the Correctness Theorem for Step 2)
There is no $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that $\mathcal{A}ut_{\neg\varphi}$ accepts its atom-set trace through L
  iff (by the Correctness Theorem for Step 1)
There is no $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$ such that $\pi \models_L \neg\varphi$
  iff (by logic)
For all $\pi \in \mathit{Paths}_{s_0}(\mathcal{M})$, we have $\pi \not\models_L \neg\varphi$

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \rightarrow, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Proof. We have the following chain of equivalent statements:

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty
 iff (by the corollary of the Correctness Theorem for Step 2)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\mathcal{A}ut_{\neg\varphi}$ accepts its atom-set trace through L
 iff (by the Correctness Theorem for Step 1)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\pi \models_L \neg\varphi$
 iff (by logic)
For all $\pi \in Paths_{s_0}(\mathcal{M})$, we have $\pi \not\models_L \neg\varphi$
 iff (by the semantics of $\neg$)
For all $\pi \in Paths_{s_0}(\mathcal{M})$, we have $\pi \models_L \varphi$

Overall Correctness Theorem

The product between an LTS with a state and the GNBA of the negation of a formula encodes the satisfaction relation in the following sense:

Overall Correctness Theorem. For any LTS $\mathcal{M} = (S, \rightarrow, L)$, state $s_0 \in S$ and formula φ: $\mathcal{M}, s_0 \models \varphi$ iff the language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Proof. We have the following chain of equivalent statements:

The language accepted by $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty
iff (by the corollary of the Correctness Theorem for Step 2)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\mathcal{A}ut_{\neg\varphi}$ accepts its atom-set trace through L
iff (by the Correctness Theorem for Step 1)
There is no $\pi \in Paths_{s_0}(\mathcal{M})$ such that $\pi \models_L \neg\varphi$
iff (by logic)
For all $\pi \in Paths_{s_0}(\mathcal{M})$, we have $\pi \not\models_L \neg\varphi$
iff (by the semantics of $\neg$)
For all $\pi \in Paths_{s_0}(\mathcal{M})$, we have $\pi \models_L \varphi$
iff (by the definition of satisfaction by LTSs)
$\mathcal{M}, s_0 \models \varphi$.

Step 3: Deciding Emptiness for GNBAs

Step 3: Deciding Emptiness for GNBAs

It is easy to see that the definitions of:

- The GNBA $\mathcal{A}ut_\psi$ (given any formula ψ) and
- The GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ (LTS $\mathcal{M}$, state s_0 and formula φ)

are computable – you can write programs (in your favorite PL) that compute them.

Step 3: Deciding Emptiness for GNBAs

It is easy to see that the definitions of:

- The GNBA $\mathcal{A}ut_\psi$ (given any formula ψ) and
- The GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ (LTS $\mathcal{M}$, state s_0 and formula φ)

are computable – you can write programs (in your favorite PL) that compute them.

Hence, the Overall Correctness Theorem reduces the model checking problem for LTL, namely determining whether $\mathcal{M}, s_0 \models \varphi$, to the problem of determining whether the language accepted by the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Step 3: Deciding Emptiness for GNBAs

It is easy to see that the definitions of:

- The GNBA $\mathcal{A}ut_{\psi}$ (given any formula ψ) and
- The GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ (LTS $\mathcal{M}$, state s_0 and formula φ)

are computable – you can write programs (in your favorite PL) that compute them.

Hence, the Overall Correctness Theorem reduces the model checking problem for LTL, namely determining whether $\mathcal{M}, s_0 \models \varphi$, to the problem of determining whether the language accepted by the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Our last piece in the puzzle:

Decidablity Theorem. Emptiness for GNBA is decidable,

Step 3: Deciding Emptiness for GNBAs

It is easy to see that the definitions of:

- The GNBA $\mathcal{A}ut_\psi$ (given any formula ψ) and
- The GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ (LTS $\mathcal{M}$, state s_0 and formula φ)

are computable – you can write programs (in your favorite PL) that compute them.

Hence, the Overall Correctness Theorem reduces the model checking problem for LTL, namely determining whether $\mathcal{M}, s_0 \models \varphi$, to the problem of determining whether the language accepted by the GNBA $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$ is empty.

Our last piece in the puzzle:

Decidablity Theorem. Emptiness for GNBA is decidable, meaning: There is a program that takes as input a GNBA $\mathcal{A}ut$, always terminates, and returns

- 'Yes', if $\text{Lang}(\mathcal{A}ut) = \emptyset$
- 'No', if $\text{Lang}(\mathcal{A}ut) \neq \emptyset$

Step 3: Deciding Emptiness for GNBAs

The Decidability Theorem will be proved with the help of a lemma.

Step 3: Deciding Emptiness for GNBAs

The Decidability Theorem will be proved with the help of a lemma.

For any GNBA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$, we define its graph $Gr(\mathcal{A}ut) = (Q, \rightarrow)$ to be the following directed graph:

- The nodes of $Gr(\mathcal{A}ut)$ are the states Q
- Given $q_1, q_2 \in Q$, there is an edge between q_1 and q_2, written $q_1 \rightarrow q_2$, iff there exists a transition $q_1 \xrightarrow{x} q_2$ for some $x \in \Sigma$.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \to)$ be a directed graph, with nodes Q and edges $\to \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \to q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $q_{m+1} \dots q_{m+n}$ is a cycle.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $q_{m+1} \dots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \ldots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \ldots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \ldots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $q_{m+1} \ldots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

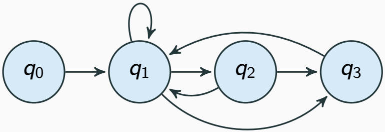
A (finite) path is a finite sequence $q_1 \ldots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \ldots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \ldots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $q_{m+1} \ldots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \ldots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \ldots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \ldots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $q_{m+1} \ldots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Example:

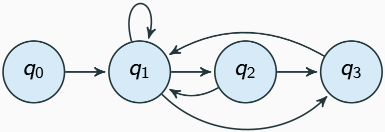


Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $q_{m+1} \dots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Example:



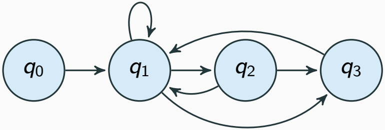
$q_1 q_1$, $q_1 q_2 q_1$ and $q_2 q_3 q_1 q_2$ are cycles.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \to)$ be a directed graph, with nodes Q and edges $\to \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \to q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $q_{m+1} \dots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Example:



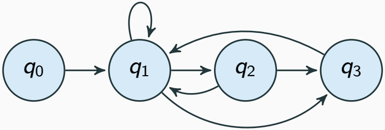
$q_1 q_1$, $q_1 q_2 q_1$ and $q_2 q_3 q_1 q_2$ are cycles.
$q_0 q_1 q_1$, $q_0 q_1 q_2 q_1$ and $q_0 q_1 q_2 q_3 q_1 q_2$ are lassos.

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $q_{m+1} \dots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Example:



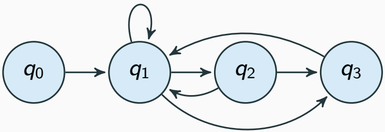
$q_1 q_1$, $q_1 q_2 q_1$ and $q_2 q_3 q_1 q_2$ are cycles.
$q_0 q_1 q_1$, $q_0 q_1 q_2 q_1$ and $q_0 q_1 q_2 q_3 q_1 q_2$ are lassos.
SCCs:
$\{q_0\}$ is maximal and trivial

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \to)$ be a directed graph, with nodes Q and edges $\to \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \ldots q_n$ where $q_i \to q_{i+1}$ for all $i \in \{1, \ldots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \ldots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $q_{m+1} \ldots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Example:



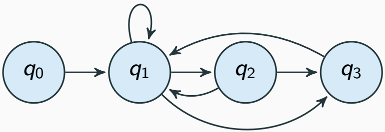
$q_1 q_1$, $q_1 q_2 q_1$ and $q_2 q_3 q_1 q_2$ are cycles.
$q_0 q_1 q_1$, $q_0 q_1 q_2 q_1$ and $q_0 q_1 q_2 q_3 q_1 q_2$ are lassos.
SCCs:
$\{q_0\}$ is maximal and trivial
$\{q_1\}$ and $\{q_1, q_2\}$ are non-maximal and non-trivial

Parenthesis: Some Graph Concepts Recalled

Let $G = (Q, \rightarrow)$ be a directed graph, with nodes Q and edges $\rightarrow \subseteq Q \times Q$.

A (finite) path is a finite sequence $q_1 \dots q_n$ where $q_i \rightarrow q_{i+1}$ for all $i \in \{1, \dots, n-1\}$. q' is accessible from q if there is a path from q to q'.
A cycle is a path $q_1 \dots q_n$ of length ≥ 2 that has the first and last nodes equal: $q_1 = q_n$.
A lasso is a path ending in a cycle – i.e., a path of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $q_{m+1} \dots q_{m+n}$ is a cycle.
A strongly connected component (SCC) is a set of nodes $C \subseteq Q$ such that, between any two elements of C, there exists a path consisting of elements of C only.
An SCC C is called maximal if there exists no other SCC C' such that $C \subset C'$. It is called non-trivial if there exists at least one edge between its nodes.

Example:



$q_1 q_1$, $q_1 q_2 q_1$ and $q_2 q_3 q_1 q_2$ are cycles.
$q_0 q_1 q_1$, $q_0 q_1 q_2 q_1$ and $q_0 q_1 q_2 q_3 q_1 q_2$ are lassos.
SCCs:
$\{q_0\}$ is maximal and trivial
$\{q_1\}$ and $\{q_1, q_2\}$ are non-maximal and non-trivial
$\{q_1, q_2, q_3\}$ is maximal and non-trivial

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

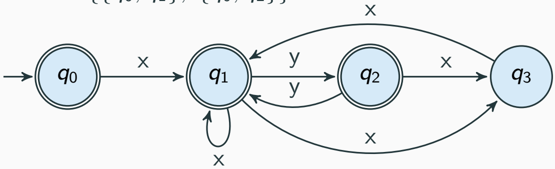
I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \dots, n\}$ with $q_{m+i} \in F$.

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \ldots, n\}$ with $q_{m+i} \in F$.

Example: Consider the GNFA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where Σ, Q, I and $\rightarrow$ are like in the picture, and $\mathcal{F} = \{\{q_0, q_1\}, \{q_0, q_2\}\}$.

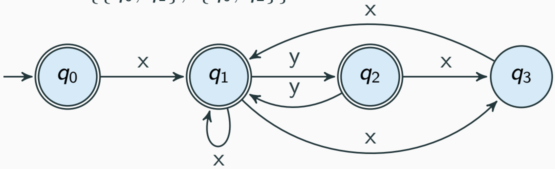


Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \ldots, n\}$ with $q_{m+i} \in F$.

Example: Consider the GNFA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where Σ, Q, I and $\rightarrow$ are like in the picture, and $\mathcal{F} = \{\{q_0, q_1\}, \{q_0, q_2\}\}$.



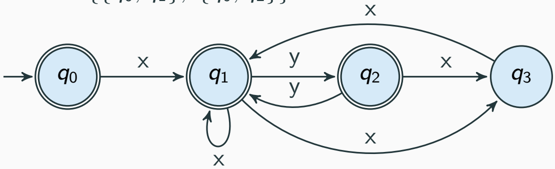
$q_0 q_1 q_2 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \ldots, n\}$ with $q_{m+i} \in F$.

Example: Consider the GNFA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where Σ, Q, I and $\rightarrow$ are like in the picture, and $\mathcal{F} = \{\{q_0, q_1\},\ \{q_0, q_2\}\}$.



$q_0 q_1 q_2 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

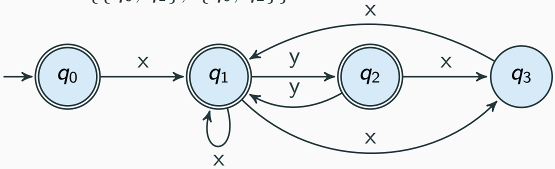
$q_0 q_1 q_2 q_3 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_3 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \ldots, n\}$ with $q_{m+i} \in F$.

Example: Consider the GNFA $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where Σ, Q, I and $\to$ are like in the picture, and $\mathcal{F} = \{\{q_0, q_1\}, \{q_0, q_2\}\}$.



$q_0 q_1 q_2 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

$q_0 q_1 q_2 q_3 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_3 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

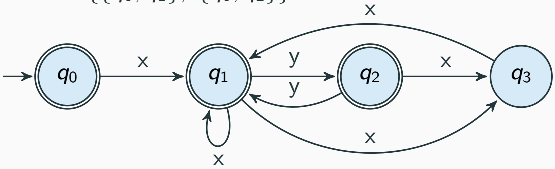
$q_0 q_1 q_3 q_1$ is not an accepting lasso since its cycle $q_1 q_3 q_1$ has no state from $\{q_0, q_2\}$.

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \dots, n\}$ with $q_{m+i} \in F$.

Example: Consider the GNFA $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where Σ, Q, I and $\to$ are like in the picture, and $\mathcal{F} = \{\{q_0, q_1\}, \{q_0, q_2\}\}$.



$q_0 q_1 q_2 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

$q_0 q_1 q_2 q_3 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_3 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

$q_0 q_1 q_3 q_1$ is not an accepting lasso since its cycle $q_1 q_3 q_1$ has no state from $\{q_0, q_2\}$.

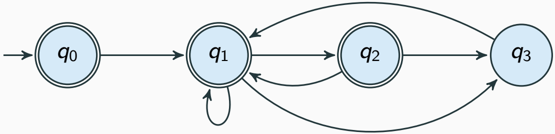
Note: When discussing accepting lassos for $\mathcal{A}ut$, labels on transitions do not matter.

Step 3: Deciding Emptiness for GNBAs

An accepting lasso for $\mathcal{A}ut$ is a lasso in $Gr(\mathcal{A}ut)$ starting in an initial state of $\mathcal{A}ut$ and containing states from all the accepting sets of $\mathcal{A}ut$ on its ending cycle.

I.e., it is a path in $Gr(\mathcal{A}ut)$ of the form $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ where $m \geq 0$, $n > 0$, $q_0 \in I$, $q_{m+n} = q_{m+1}$ and for all $F \in \mathcal{F}$, there exists $i \in \{1, \dots, n\}$ with $q_{m+i} \in F$.

Example: Consider the GNFA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where Σ, Q, I and $\rightarrow$ are like in the picture, and $\mathcal{F} = \{\{q_0, q_1\}, \{q_0, q_2\}\}$.



$q_0 q_1 q_2 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

$q_0 q_1 q_2 q_3 q_1$ is an accepting lasso because it starts in the initial state q_0, and its cycle $q_1 q_2 q_3 q_1$ contains a state from each accepting set: $q_1 \in \{q_0, q_1\}$ and $q_2 \in \{q_0, q_2\}$.

$q_0 q_1 q_3 q_1$ is not an accepting lasso since its cycle $q_1 q_3 q_1$ has no state from $\{q_0, q_2\}$.

Note: When discussing accepting lassos for $\mathcal{A}ut$, labels on transitions do not matter.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:

(1) $Lang(\mathcal{A}ut) \neq \emptyset$.

(2) There exists an accepting lasso for $\mathcal{A}ut$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:

(1) $Lang(\mathcal{A}ut) \neq \emptyset$.

(2) There exists an accepting lasso for $\mathcal{A}ut$.

(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:

- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For one direction, let $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For one direction, let $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

By the definition of an accepting lasso and of $Gr(\mathcal{A}ut)$, we have a finite word $x_0 \dots x_m x_{m+1} \dots x_{m+n-1}$ such that $q_i \xrightarrow{x_i} q_{i+1}$ for all $i \in \{0, \dots, m+n-1\}$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For one direction, let $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

By the definition of an accepting lasso and of $Gr(\mathcal{A}ut)$, we have a finite word $x_0 \dots x_m x_{m+1} \dots x_{m+n-1}$ such that $q_i \xrightarrow{x_i} q_{i+1}$ for all $i \in \{0, \dots, m+n-1\}$.

Then $q_0 \dots q_m (q_{m+1} \dots q_{m+n-1})^\infty$ is an accepting run in $\mathcal{A}ut$ for the (infinite) word $x_0 \dots x_m (x_{m+1} \dots x_{m+n-1})^\infty$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For one direction, let $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

By the definition of an accepting lasso and of $Gr(\mathcal{A}ut)$, we have a finite word $x_0 \dots x_m x_{m+1} \dots x_{m+n-1}$ such that $q_i \xrightarrow{x_i} q_{i+1}$ for all $i \in \{0, \dots, m+n-1\}$.

Then $q_0 \dots q_m (q_{m+1} \dots q_{m+n-1})^\infty$ is an accepting run in $\mathcal{A}ut$ for the (infinite) word $x_0 \dots x_m (x_{m+1} \dots x_{m+n-1})^\infty$.

So $x_0 \dots x_m (x_{m+1} \dots x_{m+n-1})^\infty \in \text{Lang}(\mathcal{A}ut)$, hence $\text{Lang}(\mathcal{A}ut) \neq \emptyset$, as desired.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For the opposite direction, let $x_0 x_1 x_2 \ldots \in \mathrm{Lang}(\mathcal{A}ut)$, and let $q_0 q_1 q_2 \ldots$ be an accepting run for it.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For the opposite direction, let $x_0 x_1 x_2 \ldots \in \text{Lang}(\mathcal{A}ut)$, and let $q_0 q_1 q_2 \ldots$ be an accepting run for it.

Since Q is finite, there exists $q \in Q$ that occurs in $q_0 q_1 q_2 \ldots$ infinitely often, and let i such that $q_i = q$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For the opposite direction, let $x_0 x_1 x_2 \ldots \in \mathrm{Lang}(\mathcal{A}ut)$, and let $q_0 q_1 q_2 \ldots$ be an accepting run for it.

Since Q is finite, there exists $q \in Q$ that occurs in $q_0 q_1 q_2 \ldots$ infinitely often, and let i such that $q_i = q$.

Since $q_0 q_1 q_2 \ldots$ is accepting, there exists $j > i$ such that $q_i q_{i+1} \ldots q_j$ contains states from each accepting set, i.e., for all $F \in \mathcal{F}$, there exists $l \in \{i, \ldots, j\}$ with $q_l \in F$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For the opposite direction, let $x_0 x_1 x_2 \ldots \in \mathrm{Lang}(\mathcal{A}ut)$, and let $q_0 q_1 q_2 \ldots$ be an accepting run for it.

Since Q is finite, there exists $q \in Q$ that occurs in $q_0 q_1 q_2 \ldots$ infinitely often, and let i such that $q_i = q$.

Since $q_0 q_1 q_2 \ldots$ is accepting, there exists $j > i$ such that $q_i q_{i+1} \ldots q_j$ contains states from each accepting set, i.e., for all $F \in \mathcal{F}$, there exists $l \in \{i, \ldots, j\}$ with $q_l \in F$.

Let $k > j$ be the index of the next occurrence of q in $q_0 q_1 q_2 \ldots$ after index j. So we have $q_i = q_k = q$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. First we show "(1) iff (2)".

For the opposite direction, let $x_0 x_1 x_2 \ldots \in \mathrm{Lang}(\mathcal{A}ut)$, and let $q_0 q_1 q_2 \ldots$ be an accepting run for it.

Since Q is finite, there exists $q \in Q$ that occurs in $q_0 q_1 q_2 \ldots$ infinitely often, and let i such that $q_i = q$.

Since $q_0 q_1 q_2 \ldots$ is accepting, there exists $j > i$ such that $q_i q_{i+1} \ldots q_j$ contains states from each accepting set, i.e., for all $F \in \mathcal{F}$, there exists $l \in \{i, \ldots, j\}$ with $q_l \in F$.

Let $k > j$ be the index of the next occurrence of q in $q_0 q_1 q_2 \ldots$ after index j. So we have $q_i = q_k = q$.

Then $q_0 \ldots q_i q_{i+1} \ldots q_k$ is an accepting lasso for $\mathcal{A}ut$, as desired.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For one direction, let $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For one direction, let $q_0 \ldots q_m q_{m+1} \ldots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

Since $q_{m+1}, \ldots, q_{m+n}$ form a cycle, they are all part of a nontrivial SCC, hence of a maximal nontrivial SCC C.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For one direction, let $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

Since $q_{m+1}, \dots, q_{m+n}$ form a cycle, they are all part of a nontrivial SCC, hence of a maximal nontrivial SCC C.

C contains states from each accepting set because $\{q_{m+1}, \dots, q_{m+n}\} \subseteq C$ does.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For one direction, let $q_0 \dots q_m q_{m+1} \dots q_{m+n}$ be an accepting lasso for $\mathcal{A}ut$.

Since $q_{m+1}, \dots, q_{m+n}$ form a cycle, they are all part of a nontrivial SCC, hence of a maximal nontrivial SCC C.

C contains states from each accepting set because $\{q_{m+1}, \dots, q_{m+n}\} \subseteq C$ does.

Finally, $q_0 \dots q_{m+1}$ is a path from a state in I to a state in C.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For the other direction, let C be an SCC with the properties mentioned at (3).

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For the other direction, let C be an SCC with the properties mentioned at (3).

Let $q_0 \dots q_1$ be a path from an initial state $q_0 \in I$ to some $q_1 \in C$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For the other direction, let C be an SCC with the properties mentioned at (3).

Let $q_0 \dots q_1$ be a path from an initial state $q_0 \in I$ to some $q_1 \in C$.

Let $q_1 q_2 \dots q_n$ be a cycle that contains all elements of C (possibly repeated) – such a cycle exists because C is an SCC.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(2) There exists an accepting lasso for $\mathcal{A}ut$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C intersects every accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Proof idea. Finally, we show "(2) iff (3)".

For the other direction, let C be an SCC with the properties mentioned at (3).

Let $q_0 \dots q_1$ be a path from an initial state $q_0 \in I$ to some $q_1 \in C$.

Let $q_1 q_2 \dots q_n$ be a cycle that contains all elements of C (possibly repeated) – such a cycle exists because C is an SCC.

Then $q_0 \dots q_1 q_2 \dots q_n$ is an accepting lasso for $\mathcal{A}ut$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:

(1) $Lang(\mathcal{A}ut) \neq \emptyset$.

(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:

- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Input: A GNBA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where $\mathcal{F} = \{F_1, \ldots, F_n\}$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Input: A GNBA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where $\mathcal{F} = \{F_1, \ldots, F_n\}$.
Let $G = Gr(\mathcal{A}ut)$.

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Input: A GNBA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where $\mathcal{F} = \{F_1, \ldots, F_n\}$.
Let $G = Gr(\mathcal{A}ut)$.
Compute G's maximal non-trivial SCCs $\{C_1, \ldots, C_m\}$ (Tarjan's DFS algorithm)

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Input: A GNBA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where $\mathcal{F} = \{F_1, \ldots, F_n\}$.
Let $G = Gr(\mathcal{A}ut)$.
Compute G's maximal non-trivial SCCs $\{C_1, \ldots, C_m\}$ (Tarjan's DFS algorithm)
For each $i \in \{1, \ldots, m\}$
    If C_i is accessible from a state in I and
        for each $j \in \{1, \ldots, n\}$, $C_i \cap F_j \neq \emptyset$

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Input: A GNBA $\mathcal{A}ut = (\Sigma, Q, I, \rightarrow, \mathcal{F})$ where $\mathcal{F} = \{F_1, \ldots, F_n\}$.
Let $G = Gr(\mathcal{A}ut)$.
Compute G's maximal non-trivial SCCs $\{C_1, \ldots, C_m\}$ (Tarjan's DFS algorithm)
For each $i \in \{1, \ldots, m\}$
    If C_i is accessible from a state in I and
        for each $j \in \{1, \ldots, n\}$, $C_i \cap F_j \neq \emptyset$
    then output "No, the accepted language is not empty."

Step 3: Deciding Emptiness for GNBAs

Lemma. Let $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ be a GNBA. Then the following are equivalent:
(1) $Lang(\mathcal{A}ut) \neq \emptyset$.
(3) There exists a maximal non-trivial SCC C of $Gr(\mathcal{A}ut)$ such that:
- some state in C is accessible from some state in I;
- C contains states from each accepting set, i.e., $C \cap F \neq \emptyset$ for all $F \in \mathcal{F}$.

Decidablity Theorem. Emptiness for GNBA is decidable.

Proof. By the above "(1) iff (3)" part of the lemma, the following algorithm decides GNBA emptiness.

Input: A GNBA $\mathcal{A}ut = (\Sigma, Q, I, \to, \mathcal{F})$ where $\mathcal{F} = \{F_1, \ldots, F_n\}$.
Let $G = Gr(\mathcal{A}ut)$.
Compute G's maximal non-trivial SCCs $\{C_1, \ldots, C_m\}$ (Tarjan's DFS algorithm)
For each $i \in \{1, \ldots, m\}$
    If C_i is accessible from a state in I and
        for each $j \in \{1, \ldots, n\}$, $C_i \cap F_j \neq \emptyset$
    then output "No, the accepted language is not empty."
Output "Yes, the accepted language is empty."

Summary of the LTL Model Checking Algorithm

Input: An LTS $\mathcal{M} = (S, \to, L)$, a state $s_0 \in S$, and an LTL formula φ.

Step 1: Compute the GNBA $\mathcal{A}ut = \mathcal{A}ut_{\neg\varphi}$.

Step 2: Compute the GNBA $\mathcal{A}ut' = (\mathcal{M}, s_0) \times \mathcal{A}ut$.

Step 3: Check whether Lang$(\mathcal{A}ut') = \emptyset$.

- If True, then output "Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$."
- If False, then output "No, it is not the case that $\mathcal{M}, s_0 \models \varphi$."

Running Example (Completed)

Let φ be $\neg \Diamond a$.

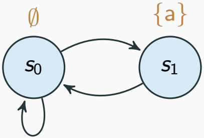
Running Example (Completed)

Let φ be $\neg \Diamond a$. Then $\neg \varphi = \overline{\overline{\Diamond \ a}} = \Diamond \ a$. (Remember we identify $\overline{\overline{\varphi}}$ with φ.)

Running Example (Completed)

Let φ be $\neg \Diamond a$. Then $\neg \varphi = \overline{\overline{\Diamond \ a}} = \Diamond \ a$. (Remember we identify $\overline{\overline{\varphi}}$ with φ.)

$$\mathcal{M} = (S, \rightarrow, L)$$

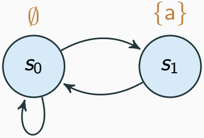


PROBLEM INSTANCE: $\boxed{\text{Does } \mathcal{M}, s_0 \models \varphi?}$

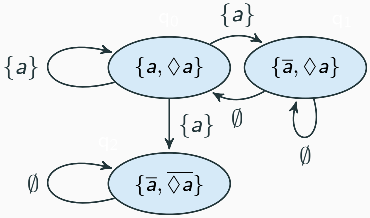
Running Example (Completed)

Let φ be $\neg \Diamond a$. Then $\neg \varphi = \overline{\overline{\Diamond\, a}} = \Diamond\, a$. (Remember we identify $\overline{\overline{\varphi}}$ with φ.)

$$\mathcal{M} = (S, \rightarrow, L)$$



STEP 1: $\mathcal{A}ut_{\neg\varphi} = (\Sigma, Q, I, \rightarrow, \mathcal{F})$

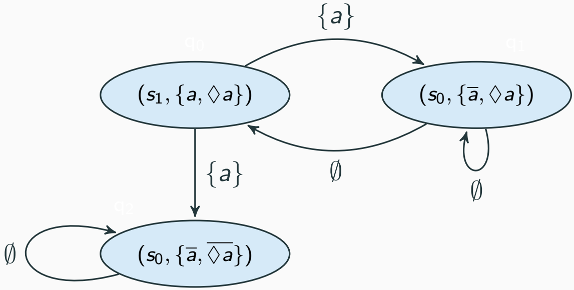


PROBLEM INSTANCE: $\boxed{\text{Does } \mathcal{M}, s_0 \models \varphi?}$

$$I = \{\, \{a, \Diamond a\},\ \{\overline{a}, \Diamond a\} \,\}$$
$$\mathcal{F} = \{\, \{\, \{a, \Diamond a\},\ \{\overline{a}, \overline{\Diamond a}\} \,\} \,\}$$

Running Example (Completed)

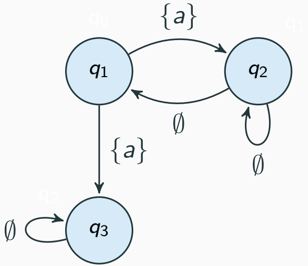
STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$



$$I_\times = \{\, (s_0, \{\overline{a}, \Diamond a\}) \,\}$$
$$\mathcal{F}_\times = \{\, \{\, (s_1, \{a, \Diamond a\}),\ (s_0, \{\overline{a}, \overline{\Diamond a}\}) \,\} \,\}$$

Running Example (Completed)

STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$

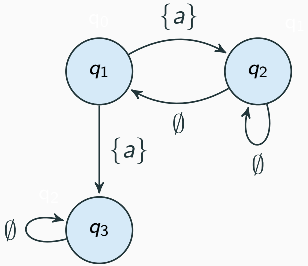


$I_\times = \{ q_2 \}$

$\mathcal{F}_\times = \{ \{ q_1, q_3 \} \}$

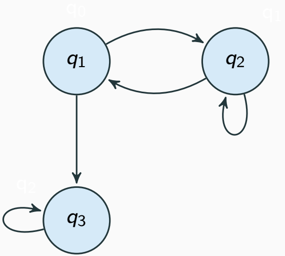
Running Example (Completed)

STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$



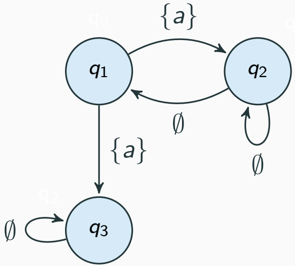
$$I_\times = \{\, q_2 \,\}$$
$$\mathcal{F}_\times = \{\, \{\, q_1,\ q_3 \,\} \,\}$$

STEP 3: $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



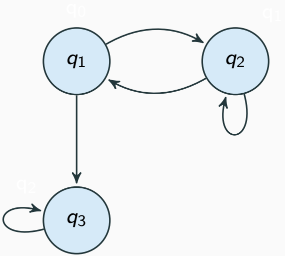
Running Example (Completed)

STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$



$I_\times = \{ q_2 \}$
$\mathcal{F}_\times = \{ \{ q_1, \ q_3 \} \}$

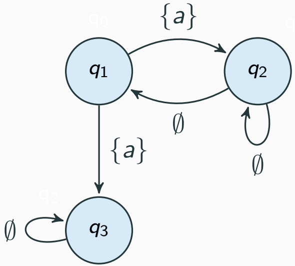
STEP 3: $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



Two maximal non-trivial SCCs: $\{q_1, q_2\}$ and $\{q_3\}$.

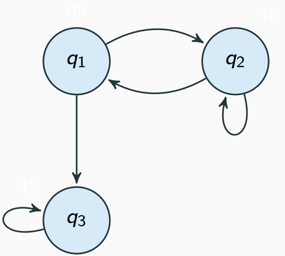
Running Example (Completed)

STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$



$I_\times = \{\, q_2 \,\}$
$\mathcal{F}_\times = \{\, \{\, q_1,\ q_3 \,\} \,\}$

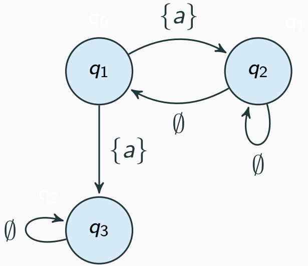
STEP 3: $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



Two maximal non-trivial SCCs: $\{q_1, q_2\}$ and $\{q_3\}$.
$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.
$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

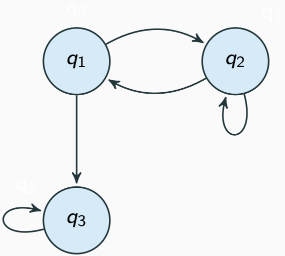
Running Example (Completed)

STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$



$I_\times = \{ q_2 \}$
$\mathcal{F}_\times = \{ \{ q_1, \ q_3 \} \}$

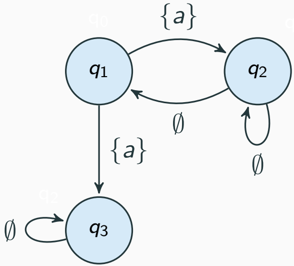
STEP 3: $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



Two maximal non-trivial SCCs: $\{q_1, q_2\}$ and $\{q_3\}$.
$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.
$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.
Hence $\text{Lang}((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$.

Running Example (Completed)

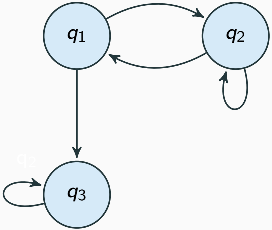
STEP 2: $(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}$



$I_\times = \{ q_2 \}$

$\mathcal{F}_\times = \{ \{ q_1, \; q_3 \} \}$

STEP 3: $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



Two maximal non-trivial SCCs: $\{q_1, q_2\}$ and $\{q_3\}$.

$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

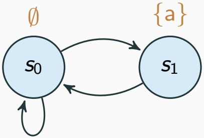
Hence Lang$((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$.

We conclude: No, it is not the case that $\mathcal{M}, s_0 \models \varphi$.

Running Example (Completed)

Let φ be $\neg \Diamond a$. Then $\neg \varphi = \overline{\overline{\Diamond\ a}} = \Diamond\ a$. (Remember we identify $\overline{\overline{\varphi}}$ with φ.)

$$\mathcal{M} = (S, \rightarrow, L)$$



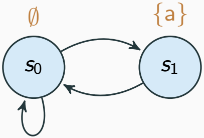
PROBLEM INSTANCE: $\boxed{\text{Does } \mathcal{M}, s_0 \models \varphi?}$

We conclude: No, it is not the case that $\mathcal{M}, s_0 \models \varphi$.

Running Example (Completed)

Let φ be $\neg \Diamond a$. Then $\neg \varphi = \overline{\overline{\Diamond\ a}} = \Diamond\ a$. (Remember we identify $\overline{\overline{\varphi}}$ with φ.)

$$\mathcal{M} = (S, \rightarrow, L)$$



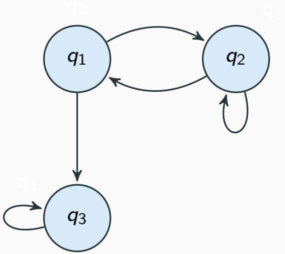
PROBLEM INSTANCE: $\boxed{\text{Does } \mathcal{M}, s_0 \models \varphi?}$

We conclude: No, it is not the case that $\mathcal{M}, s_0 \models \varphi$.

For example, $(s_0 s_1)^\infty \models \Diamond\ a$, hence $(s_0 s_1)^\infty \not\models_L \neg \Diamond\ a$, i.e., $(s_0 s_1)^\infty \not\models_L \varphi$.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\quad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$

$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.

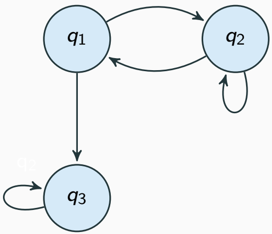
$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

Hence Lang$((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. We conclude: $\mathcal{M}, s_0 \not\models \varphi$.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\qquad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$

$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.

$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

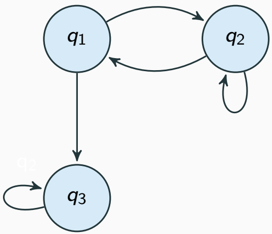
$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

Hence Lang$((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. $\boxed{\text{We conclude: } \mathcal{M}, s_0 \not\models \varphi.}$

Build a lasso: Start with a path from an initial state to our SCC: here, just q_2.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\qquad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$

$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.

$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

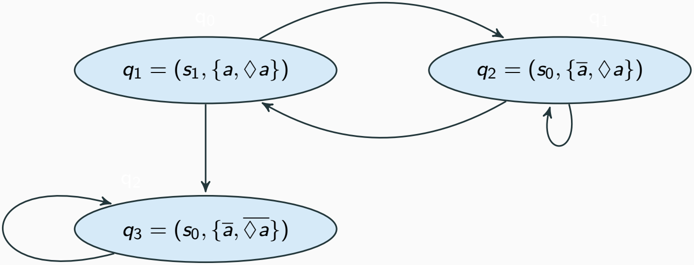
$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

Hence $\text{Lang}((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. $\boxed{\text{We conclude: } \mathcal{M}, s_0 \not\models \varphi.}$

Build a lasso: Start with a path from an initial state to our SCC: here, just q_2.
Continue with a cycle that covers the entire SCC: $q_2 q_1 q_2$.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\qquad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$

$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.

$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

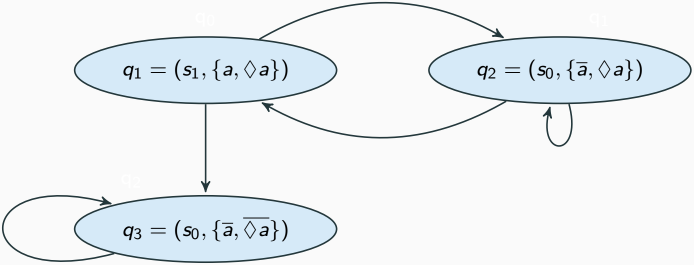
$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

Hence $\mathrm{Lang}((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. $\boxed{\text{We conclude: } \mathcal{M}, s_0 \not\models \varphi.}$

Build a lasso: Start with a path from an initial state to our SCC: here, just q_2.
Continue with a cycle that covers the entire SCC: $q_2 q_1 q_2$.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\qquad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$
$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.
$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.
$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

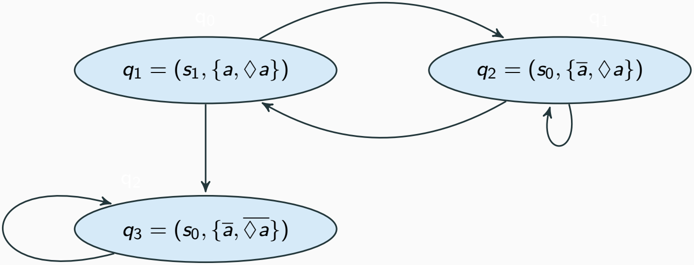
Hence Lang$((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. $\boxed{\text{We conclude: } \mathcal{M}, s_0 \not\models \varphi.}$

Build a lasso: Start with a path from an initial state to our SCC: here, just q_2.
Continue with a cycle that covers the entire SCC: $q_2 q_1 q_2$.

Take the LTS state component of the product states: $s_0 s_1 s_0$.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\quad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$

$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.

$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

Hence $\mathrm{Lang}((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. $\boxed{\text{We conclude: } \mathcal{M}, s_0 \not\models \varphi.}$

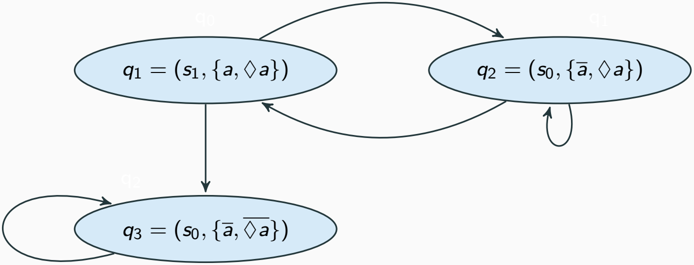
Build a lasso: Start with a path from an initial state to our SCC: here, just q_2.
Continue with a cycle that covers the entire SCC: $q_2 q_1 q_2$.

Take the LTS state component of the product states: $s_0 s_1 s_0$.

This gives us a counterexample path: $(s_0 s_1)^\infty$.

Counterexample Path

$(\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi} = (\Sigma, Q_\times, I_\times, \rightarrow_\times, \mathcal{F}_\times)$ $\quad$ $Gr((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi})$



$I_\times = \{q_2\}$

$\mathcal{F}_\times = \{\ \{q_1, q_3\}\ \}$

Found $\{q_1, q_2\}$ maximal non-trivial SCC.

$\{q_1, q_2\}$ is accessible from $q_2 \in I_\times$.

$\{q_1, q_2\}$ intersects the only accepting set, $\{q_1, q_3\}$.

Hence Lang$((\mathcal{M}, s_0) \times \mathcal{A}ut_{\neg\varphi}) \neq \emptyset$. $\boxed{\text{We conclude: } \mathcal{M}, s_0 \not\models \varphi.}$

Build a lasso: Start with a path from an initial state to our SCC: here, just q_2.
Continue with a cycle that covers the entire SCC: $q_2 q_1 q_2$.

Take the LTS state component of the product states: $s_0 s_1 s_0$.

This gives us a counterexample path: $(s_0 s_1)^\infty$. Indeed, $(s_0 s_1)^\infty \not\models_L \varphi$.

Complexity

Complexity of the LTL Model Checking Algorithm

Input: An LTS $\mathcal{M} = (S, \to, L)$, a state $s_0 \in S$, and an LTL formula φ.

Step 1: Compute the GNBA $\mathcal{A}ut = \mathcal{A}ut_{\neg\varphi}$.

Step 2: Compute the GNBA $\mathcal{A}ut' = (\mathcal{M}, s_0) \times \mathcal{A}ut$.

Step 3: Check whether $\text{Lang}(\mathcal{A}ut') = \emptyset$.

- If True, then output "Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$."
- If False, then output "No, it is not the case that $\mathcal{M}, s_0 \models \varphi$."

Complexity of the LTL Model Checking Algorithm

Input: An LTS $\mathcal{M} = (S, \to, L)$, a state $s_0 \in S$, and an LTL formula φ.

Step 1: Compute the GNBA $\mathcal{A}ut = \mathcal{A}ut_{\neg\varphi}$.
Can be done in $2^{O(|\varphi|)}$ time and space.

Step 2: Compute the GNBA $\mathcal{A}ut' = (\mathcal{M}, s_0) \times \mathcal{A}ut$.

Step 3: Check whether $\mathrm{Lang}(\mathcal{A}ut') = \emptyset$.

- If True, then output "Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$."
- If False, then output "No, it is not the case that $\mathcal{M}, s_0 \models \varphi$."

Complexity of the LTL Model Checking Algorithm

Input: An LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s_0 \in S$, and an LTL formula φ.

Step 1: Compute the GNBA $\mathcal{A}ut = \mathcal{A}ut_{\neg\varphi}$.
Can be done in $2^{O(|\varphi|)}$ time and space.

Step 2: Compute the GNBA $\mathcal{A}ut' = (\mathcal{M}, s_0) \times \mathcal{A}ut$.
Can be done in $O(|\mathcal{M}| \times |\mathcal{A}ut|)$ time and space.

Step 3: Check whether $\mathrm{Lang}(\mathcal{A}ut') = \emptyset$.

- If True, then output "Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$."
- If False, then output "No, it is not the case that $\mathcal{M}, s_0 \models \varphi$."

Complexity of the LTL Model Checking Algorithm

Input: An LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s_0 \in S$, and an LTL formula φ.

Step 1: Compute the GNBA $\mathcal{A}ut = \mathcal{A}ut_{\neg\varphi}$.
Can be done in $2^{O(|\varphi|)}$ time and space.

Step 2: Compute the GNBA $\mathcal{A}ut' = (\mathcal{M}, s_0) \times \mathcal{A}ut$.
Can be done in $O(|\mathcal{M}| \times |\mathcal{A}ut|)$ time and space.

Step 3: Check whether $\mathrm{Lang}(\mathcal{A}ut') = \emptyset$.

- If True, then output "Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$."
- If False, then output "No, it is not the case that $\mathcal{M}, s_0 \models \varphi$."

Can be done in $O(|\mathcal{A}ut'|)$ time and space.

Complexity of the LTL Model Checking Algorithm

Input: An LTS $\mathcal{M} = (S, \rightarrow, L)$, a state $s_0 \in S$, and an LTL formula φ.

Step 1: Compute the GNBA $\mathcal{A}ut = \mathcal{A}ut_{\neg\varphi}$.
Can be done in $2^{O(|\varphi|)}$ time and space.

Step 2: Compute the GNBA $\mathcal{A}ut' = (\mathcal{M}, s_0) \times \mathcal{A}ut$.
Can be done in $O(|\mathcal{M}| \times |\mathcal{A}ut|)$ time and space.

Step 3: Check whether $\text{Lang}(\mathcal{A}ut') = \emptyset$.

- If True, then output "Yes, it is the case that $\mathcal{M}, s_0 \models \varphi$."
- If False, then output "No, it is not the case that $\mathcal{M}, s_0 \models \varphi$."

Can be done in $O(|\mathcal{A}ut'|)$ time and space.

Overall complexity: $O(|\mathcal{M}| \times 2^{O(|\varphi|)})$ time and space.

Summary

Summary of the Discussed Concepts

The model checking problem for LTL

GNBA = Generalized Nondeterministic Büchi Automata

Language accepted by a GNBA

Translation of LTL formulas to GNBAs

Construction of product GNBAs

Deciding the emptiness for (the language accpted by) GNBAs

The three steps of the LTL model checking algorithm

Time and space complexity

Possible Group Presentation Topic

In groups of three, implement the LTL model checking algorithm, where each member of the group takes care of one of the three steps.

Possible Group Presentation Topic

In groups of three, implement the LTL model checking algorithm, where each member of the group takes care of one of the three steps.

Some coordination is of course necessary, but the three steps can be coupled quite loosely if you agree on their input and output formats.

Possible Group Presentation Topic

In groups of three, implement the LTL model checking algorithm, where each member of the group takes care of one of the three steps.

Some coordination is of course necessary, but the three steps can be coupled quite loosely if you agree on their input and output formats.

Feel free to discuss on the COM4507/6507 forum your choice of programming language, libraries, data structures, etc.

Possible Group Presentation Topic

In groups of three, implement the LTL model checking algorithm, where each member of the group takes care of one of the three steps.

Some coordination is of course necessary, but the three steps can be coupled quite loosely if you agree on their input and output formats.

Feel free to discuss on the COM4507/6507 forum your choice of programming language, libraries, data structures, etc.

Note: This task would also be a good preparation for the exam!

Further Reading

Section 5.2 of Baier & Katoen's "Principles of Model Checking" (MIT Press 2008)

Further Reading

Section 5.2 of Baier & Katoen's "Principles of Model Checking" (MIT Press 2008)

Moshe Vardi. An automata-theoretic approach to linear temporal logic. 1996.

Moshe Vardi. Automata-Theoretic Model Checking Revisited. 2007.