Teaching Runtime Verification

Martin Leucker
(minor modifications by Borzoo Bonakdarpour)

Institute for Software Engineering and Programming Languages
University of Lübeck

18. November 2013

Outline

Runtime Verification

Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

Presentation outline

Runtime Verification

Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

Runtime Verification

Definition (Runtime Verification)

Runtime verification is the discipline of computer science that deals with the study, development, and application of those verification techniques that allow checking whether a *run* of a system under scrutiny (SUS) satisfies or violates a given correctness property.

Its distinguishing research effort lies in *synthesizing monitors from high level specifications.*

Runtime Verification

Definition (Runtime Verification)

Runtime verification is the discipline of computer science that deals with the study, development, and application of those verification techniques that allow checking whether a *run* of a system under scrutiny (SUS) satisfies or violates a given correctness property.

Its distinguishing research effort lies in *synthesizing monitors from high level specifications.*

Definition (Monitor)

A monitor is a device that reads a finite trace and yields a certain verdict.

A verdict is typically a truth value from some truth domain.

Taxonomy


runtime verification
trace
finite
finite non-completed
infinite
integration
inline
outline
stage
online
offline
interference
invasive
non-invasive
steering
passive
active
monitoring
event sequence
state sequence
input/ output behavior
application area
information collection
performance evaluation
security
safety checking

Presentation outline

Runtime Verification

Runtime Verification for LTL

- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

Runtime Verification for LTL

Observing executions/runs

Runtime Verification for LTL

Observing executions/runs

Idea

Specify correctness properties in LTL

Runtime Verification for LTL

Observing executions/runs

Idea

Specify correctness properties in LTL

Commercial

Specify correctness properties in Regular LTL

Runtime Verification for LTL

Definition (Syntax of LTL formulae)

Let p be an atomic proposition from a finite set of atomic propositions AP. The set of LTL formulae, denoted with LTL, is inductively defined by the following grammar:

$$\begin{array}{rcl} \varphi & ::= & \mathit{true} \mid p \mid \varphi \vee \varphi \mid \varphi \; \mathit{U} \; \varphi \mid \mathit{X}\varphi \mid \\ & & \mathit{false} \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \; \mathit{R} \; \varphi \mid \bar{\mathit{X}}\varphi \mid \\ & & \neg\varphi \end{array}$$

LTL for the working engineer??

Simple??

"LTL is for theoreticians—but for practitioners?"

Runtime Verification for LTL

Idea

Specify correctness properties in LTL

Definition (Syntax of LTL formulae)

Let p be an atomic proposition from a finite set of atomic propositions AP. The set of LTL formulae, denoted with LTL, is inductively defined by the following grammar:

$$\begin{aligned} \varphi \quad ::= \quad & \mathit{true} \mid p \mid \varphi \vee \varphi \mid \varphi \; U \; \varphi \mid X\varphi \mid \\ & \mathit{false} \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \; R \; \varphi \mid \bar{X}\varphi \mid \\ & \neg\varphi \end{aligned}$$

Monitoring LTL on finite words

(Bad) Idea

just compute semantics. . .

Outline

Runtime Verification

Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

LTL on finite, but not completed words

Application area: Specify properties of finite but expanding word

LTL on finite, but not completed words

Be Impartial!

- go for a final verdict ($\top$ or $\bot$) only if you really know

LTL on finite, but not completed words

Be Impartial!

- go for a final verdict ($\top$ or $\bot$) only if you really know
- be a man: stick to your word

LTL on finite, but not complete words

Impartiality implies multiple values

Every two-valued logic is not impartial.

Monitoring LTL on finite but expanding words

Left-to-right!

Monitoring LTL on finite but expanding words

Automata-theoretic approach

- Synthesize automaton
- Monitoring = stepping through automaton

Outline

Runtime Verification

Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

Anticipatory Semantics

Consider possible extensions of the non-completed word

Outline

Runtime Verification

Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

LTL for RV [BLS@FSTTCS'06]

Basic idea

- LTL over infinite words is commonly used for specifying correctness properties
- finite words in RV:
  prefixes of infinite, so-far unknown words
- re-use existing semantics

LTL for RV [BLS@FSTTCS'06]

Basic idea

- LTL over infinite words is commonly used for specifying correctness properties
- finite words in RV:
  prefixes of infinite, so-far unknown words
- re-use existing semantics

3-valued semantics for LTL over finite words (LTL_3)

$$[u \models \varphi] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \models \varphi \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \not\models \varphi \\ ? & \text{else} \end{cases}$$

Impartial Anticipation

Impartial

- Stay with $\top$ and $\bot$

Impartial Anticipation

Impartial

- Stay with $\top$ and $\bot$

Anticipatory

- Go for $\top$ or $\bot$
- Consider *XXX false*

$$\epsilon \quad \models \quad \mathit{XXX}\,\mathit{false}$$

Impartial Anticipation

Impartial

- Stay with $\top$ and $\bot$

Anticipatory

- Go for $\top$ or $\bot$
- Consider *XXX false*

$$\begin{array}{lll} \epsilon & \models & \mathit{XXX}\,\mathit{false} \\ a & \models & \mathit{XX}\,\mathit{false} \end{array}$$

Impartial Anticipation

Impartial

- Stay with $\top$ and $\bot$

Anticipatory

- Go for $\top$ or $\bot$
- Consider *XXX false*

$$\begin{array}{lcl} \epsilon & \models & \mathit{XXX}\,\mathit{false} \\ a & \models & \mathit{XX}\,\mathit{false} \\ aa & \models & \mathit{X}\,\mathit{false} \end{array}$$

Impartial Anticipation

Impartial

- Stay with $\top$ and $\bot$

Anticipatory

- Go for $\top$ or $\bot$
- Consider *XXX false*

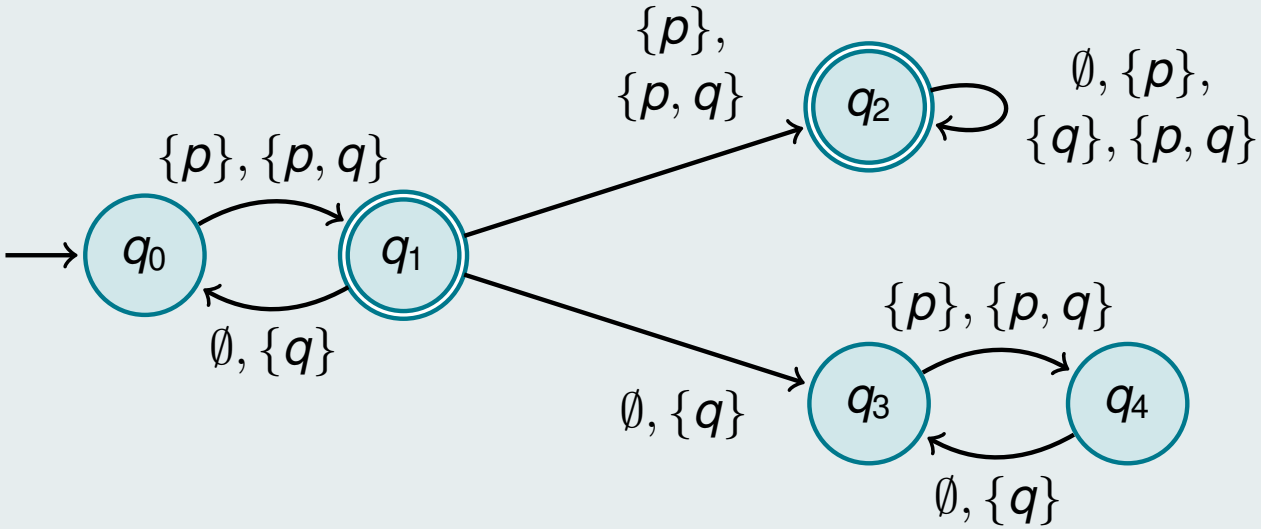
$$\begin{array}{lcl} \epsilon & \models & \mathit{XXX}\,\mathit{false} \\ a & \models & \mathit{XX}\,\mathit{false} \\ aa & \models & \mathit{X}\,\mathit{false} \\ aaa & \models & \mathit{false} \end{array}$$

$$[\epsilon \models \mathit{XXX}\,\mathit{false}] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : \epsilon\sigma \models \mathit{XXX}\,\mathit{false} \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : \epsilon\sigma \not\models \mathit{XXX}\,\mathit{false} \\ ? & \text{else} \end{cases}$$

Monitor construction – Idea I

$$[u \models \varphi] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \models \varphi \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \not\models \varphi \\ ? & \text{else} \end{cases}$$

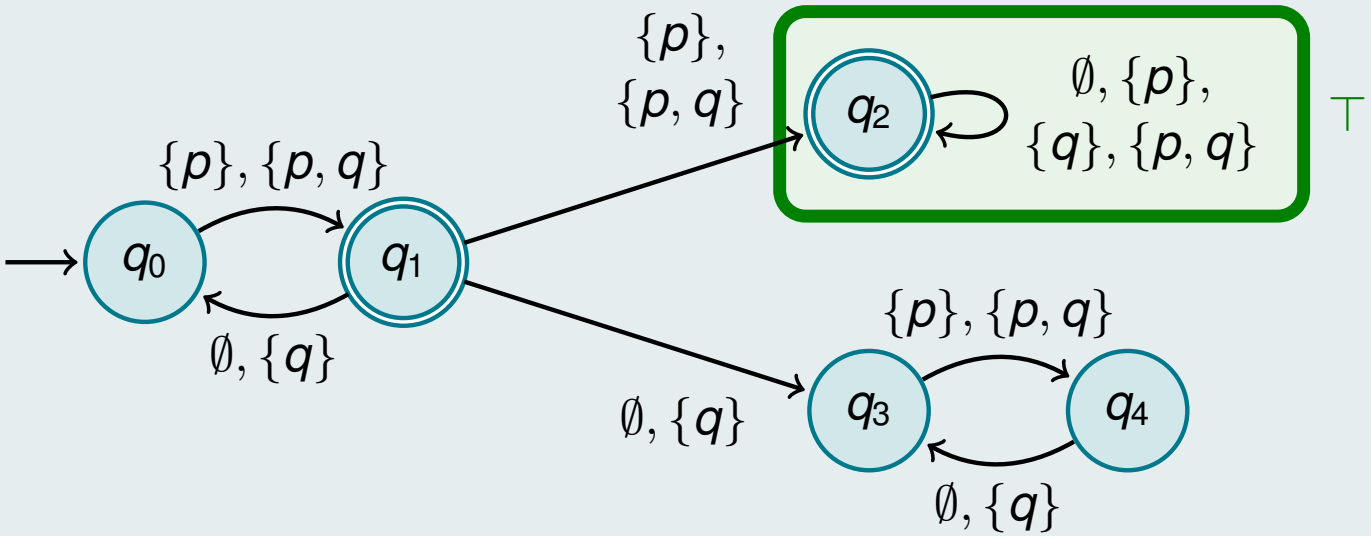
For example consider $AP = \{p, q\}$ and $\Sigma = 2^{AP}$:



Monitor construction – Idea I

$$[u \models \varphi] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \models \varphi \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \not\models \varphi \\ ? & \text{else} \end{cases}$$

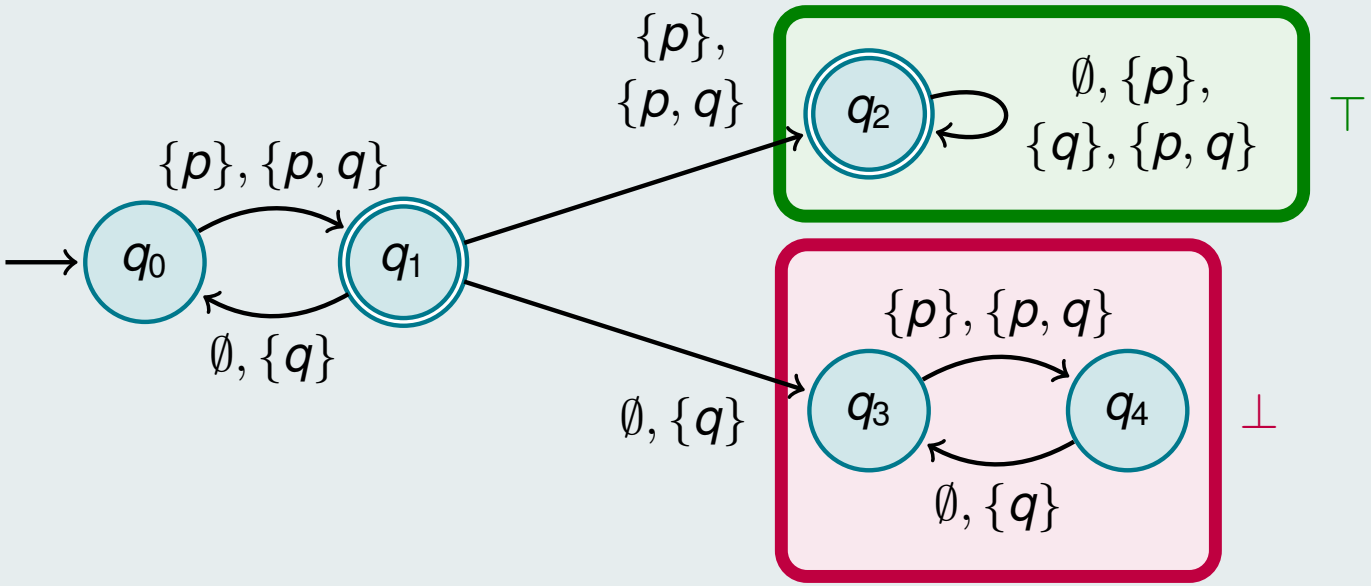
For example consider $AP = \{p, q\}$ and $\Sigma = 2^{AP}$:



Monitor construction – Idea I

$$[u \models \varphi] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \models \varphi \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \not\models \varphi \\ ? & \text{else} \end{cases}$$

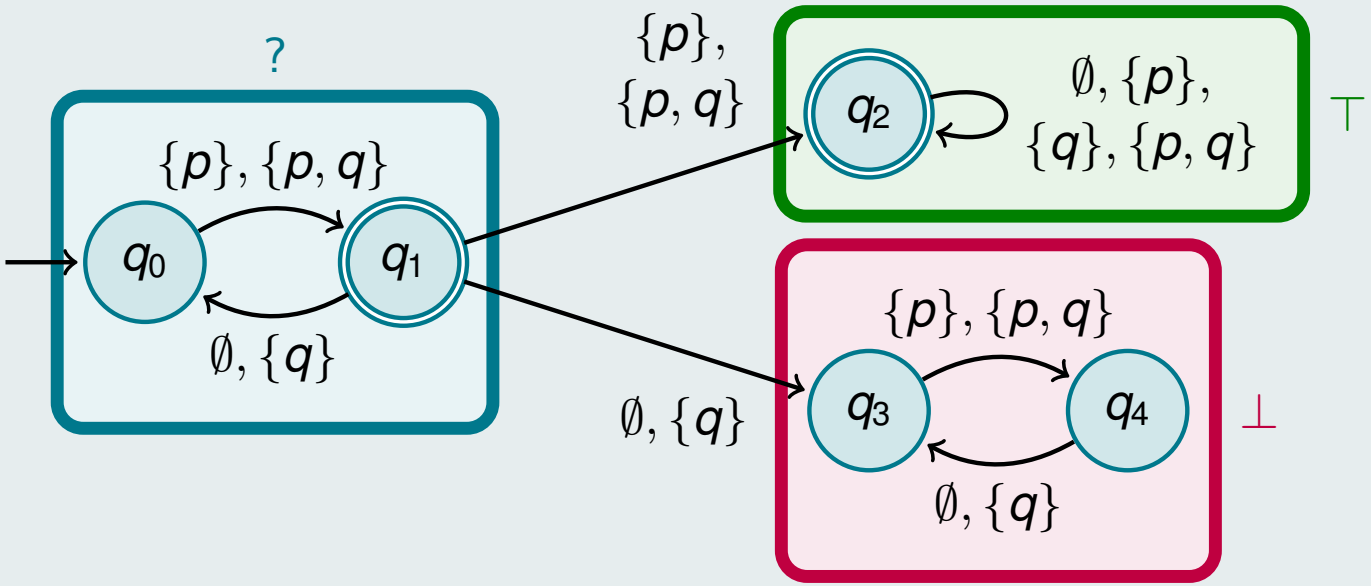
For example consider $AP = \{p, q\}$ and $\Sigma = 2^{AP}$:



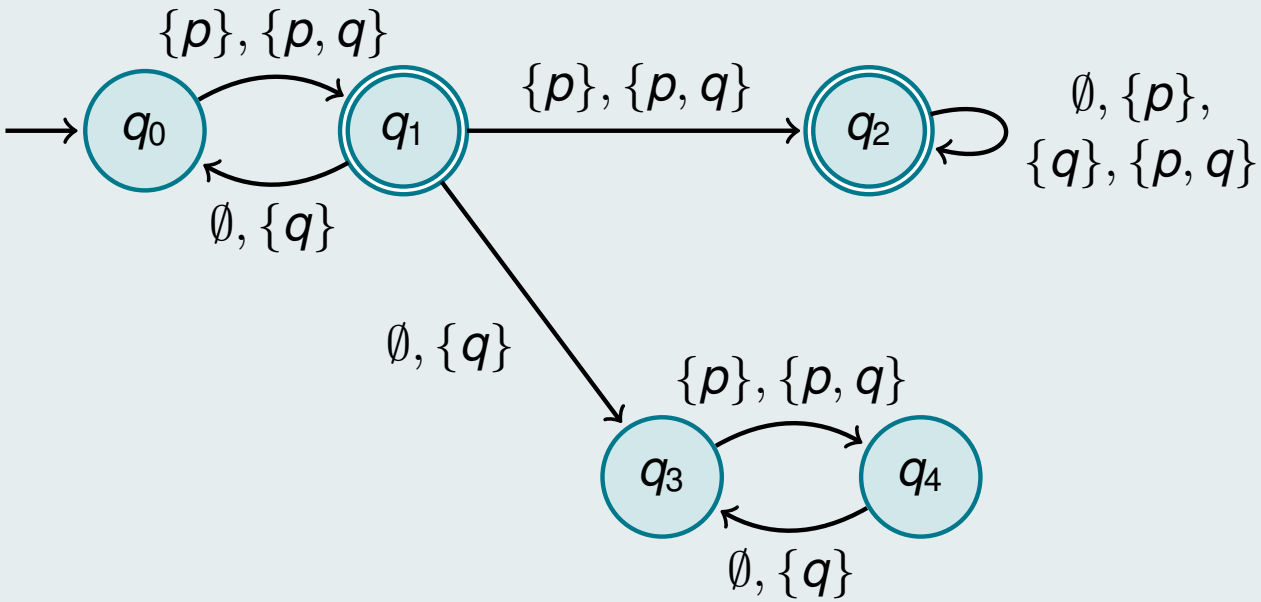
Monitor construction – Idea I

$$[u \models \varphi] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \models \varphi \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \not\models \varphi \\ ? & \text{else} \end{cases}$$

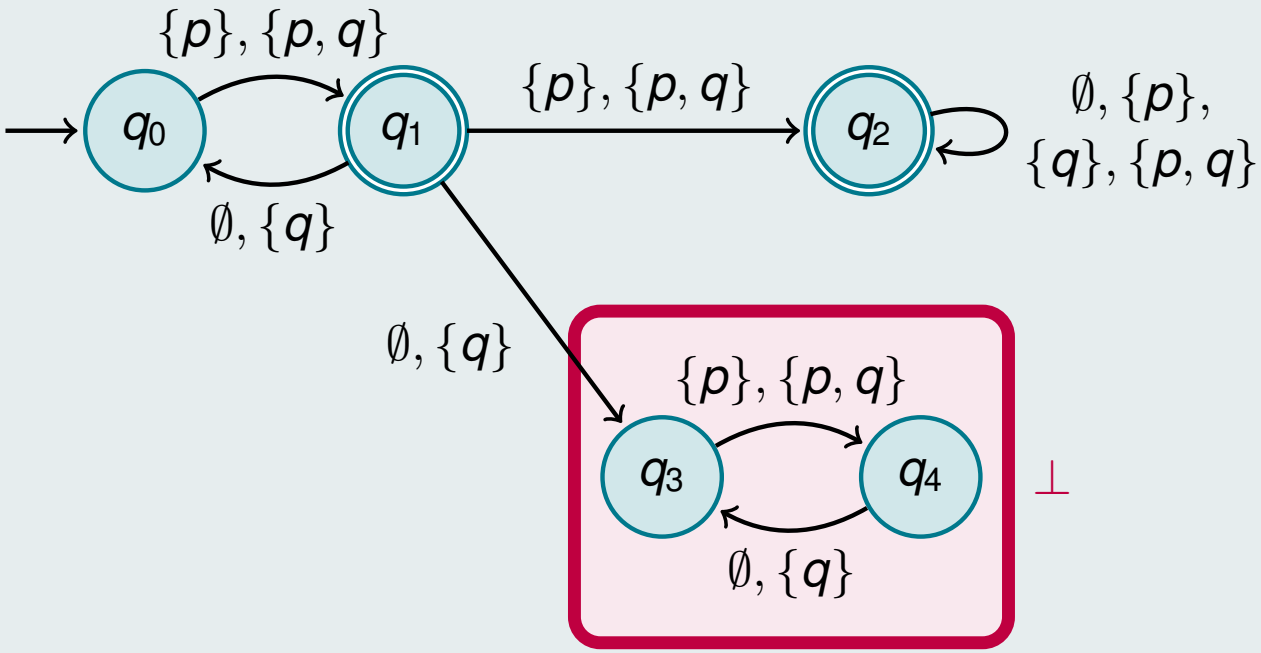
For example consider $AP = \{p, q\}$ and $\Sigma = 2^{AP}$:



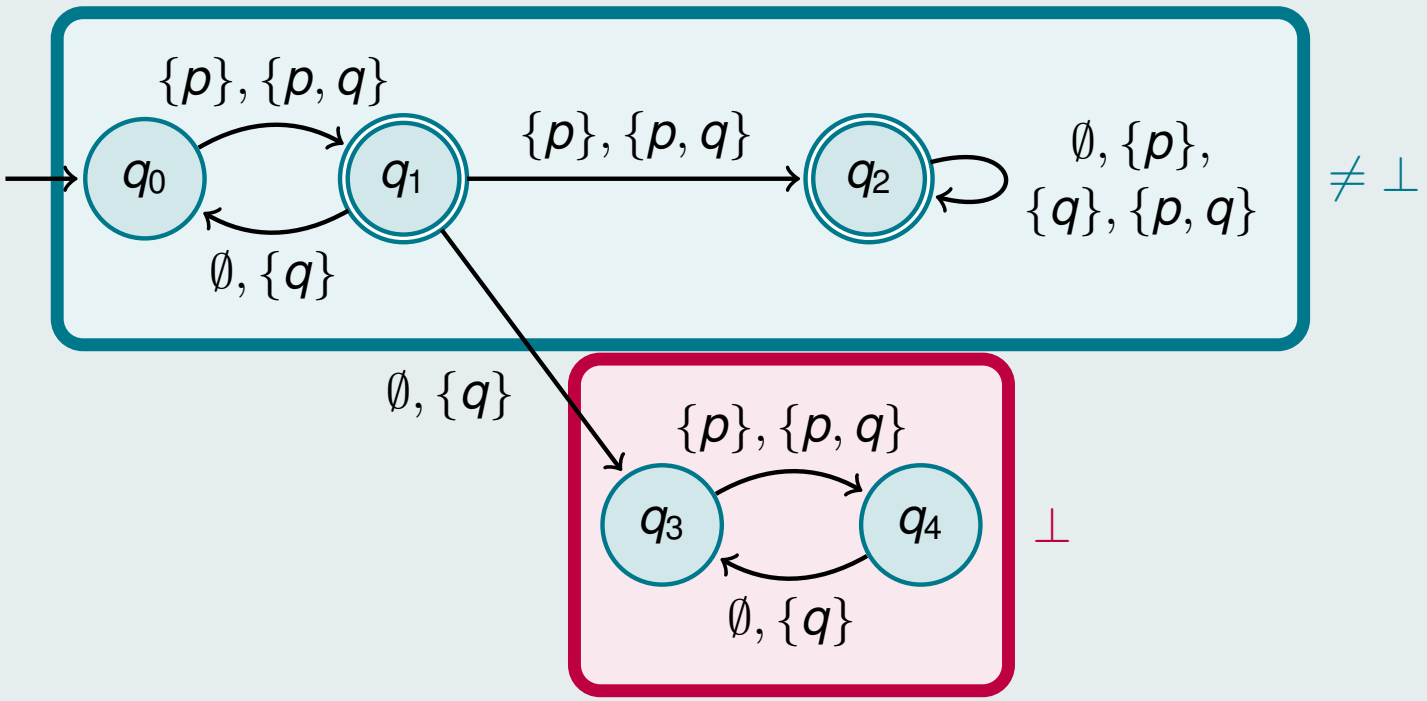
monitor construction – Idea II



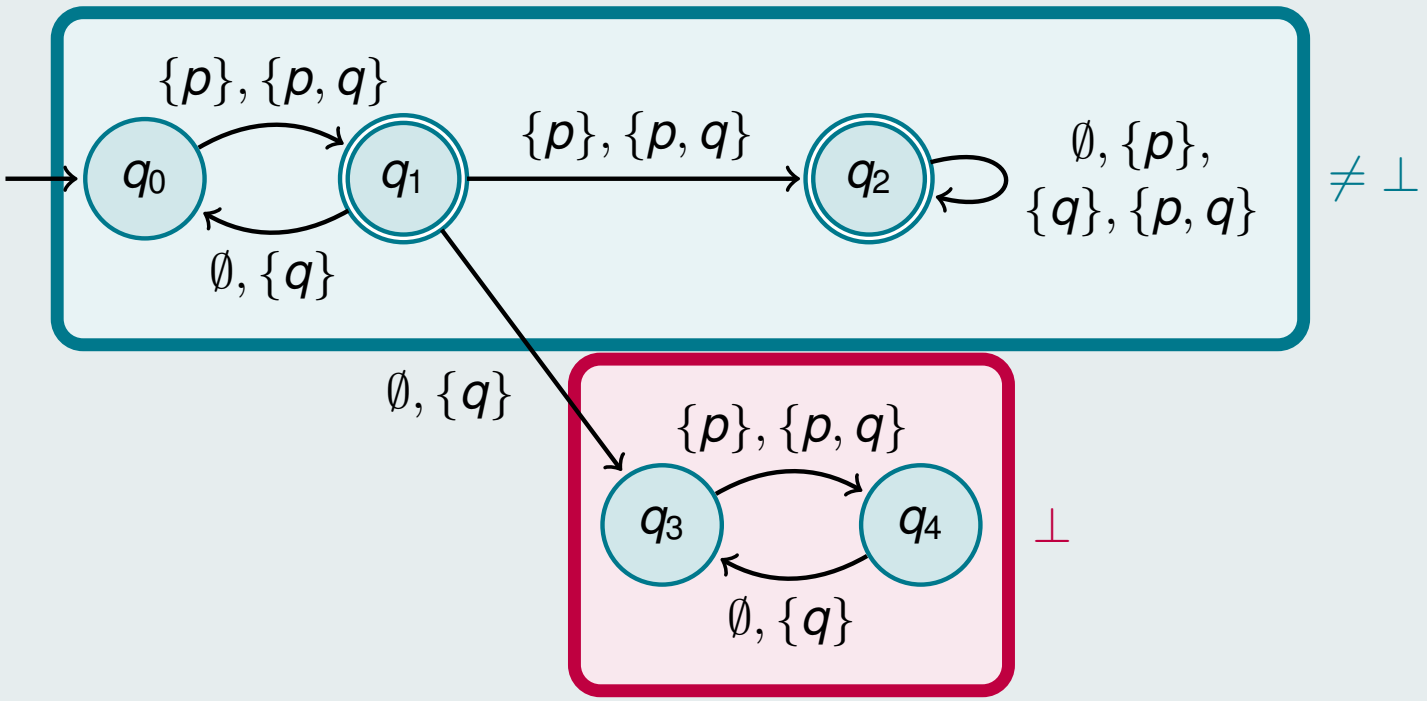
monitor construction – Idea II



monitor construction – Idea II



monitor construction – Idea II



NFA

$\mathcal{F}_\varphi : Q_\varphi \to \{\top, \bot\}$ Emptiness per state

The complete construction

The construction

LTL	BA	Emptiness per State	NFA
φ	$\longrightarrow \mathcal{A}^{\varphi}$	$\longrightarrow \mathcal{F}^{\varphi}$	$\longrightarrow \hat{\mathcal{A}}^{\varphi}$

LTL_3 Evaluation

$$[u \models \varphi] = \begin{cases} \top \\ \bot & \text{if } u \notin \mathcal{L}(\hat{\mathcal{A}}^{\varphi}) \\ ? \end{cases}$$

The complete construction

The construction

LTL	BA	Emptiness per State	NFA

$$\varphi \longrightarrow \mathcal{A}^{\varphi} \longrightarrow \mathcal{F}^{\varphi} \longrightarrow \hat{\mathcal{A}}^{\varphi}$$

$$\neg\varphi$$

LTL_3 Evaluation

$$[u \models \varphi] = \begin{cases} \top \\ \bot & \text{if } u \notin \mathcal{L}(\hat{\mathcal{A}}^{\varphi}) \\ ? \end{cases}$$

The complete construction

The construction

LTL	BA	Emptiness per State	NFA
φ	$\longrightarrow \mathcal{A}^{\varphi}$	$\longrightarrow \mathcal{F}^{\varphi}$	$\longrightarrow \hat{\mathcal{A}}^{\varphi}$
$\neg\varphi$	$\rightarrow \mathcal{A}^{\neg\varphi}$	$\longrightarrow \mathcal{F}^{\neg\varphi}$	$\longrightarrow \hat{\mathcal{A}}^{\neg\varphi}$

LTL_3 Evaluation

$$[u \models \varphi] = \begin{cases} \top & \text{if } u \notin \mathcal{L}(\hat{\mathcal{A}}^{\neg\varphi}) \\ \bot & \text{if } u \notin \mathcal{L}(\hat{\mathcal{A}}^{\varphi}) \\ ? & \text{else} \end{cases}$$

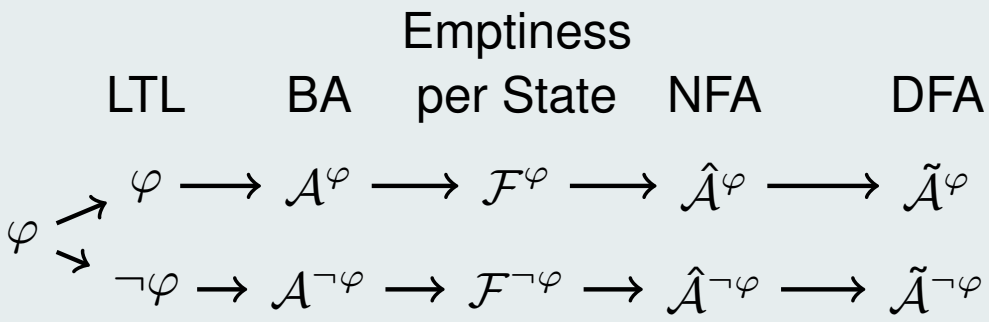
The complete construction

The construction

LTL	BA	Emptiness per State	NFA
φ	$\mathcal{A}^{\varphi}$	$\mathcal{F}^{\varphi}$	$\hat{\mathcal{A}}^{\varphi}$
$\neg\varphi$	$\mathcal{A}^{\neg\varphi}$	$\mathcal{F}^{\neg\varphi}$	$\hat{\mathcal{A}}^{\neg\varphi}$

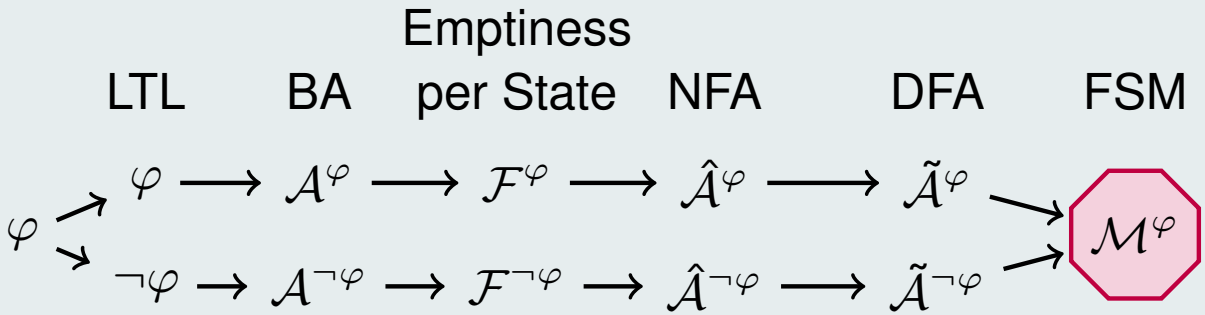
The complete construction

The construction



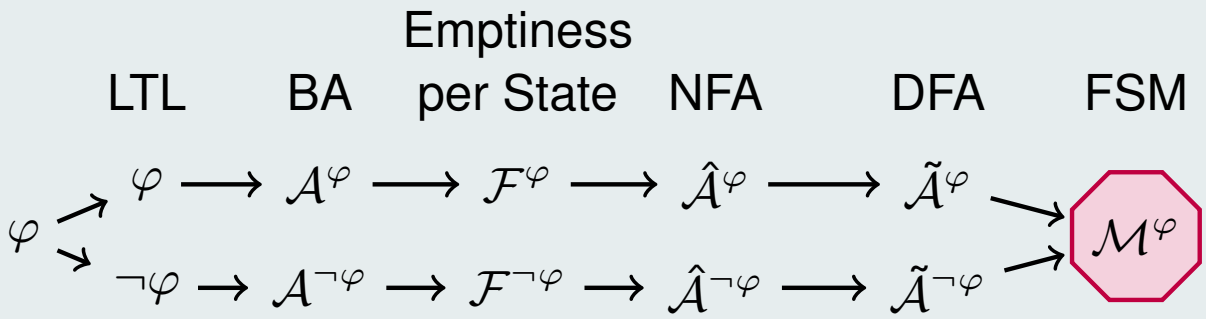
The complete construction

The construction



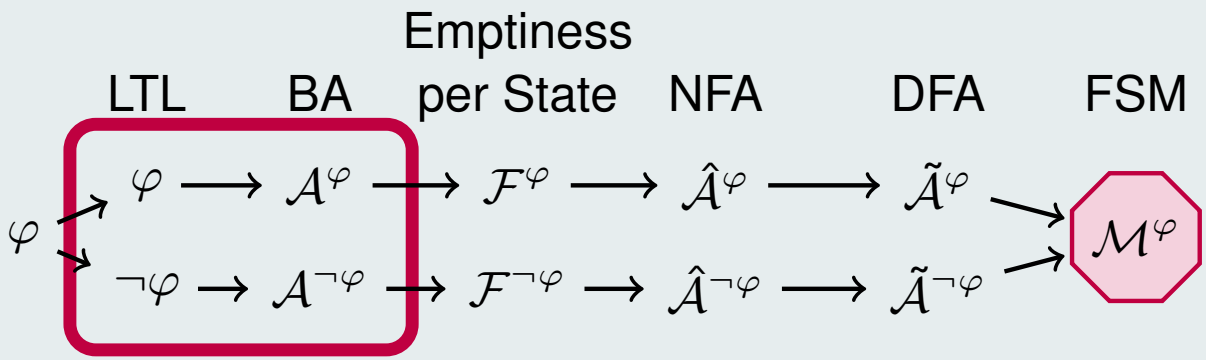
Complexity

The construction



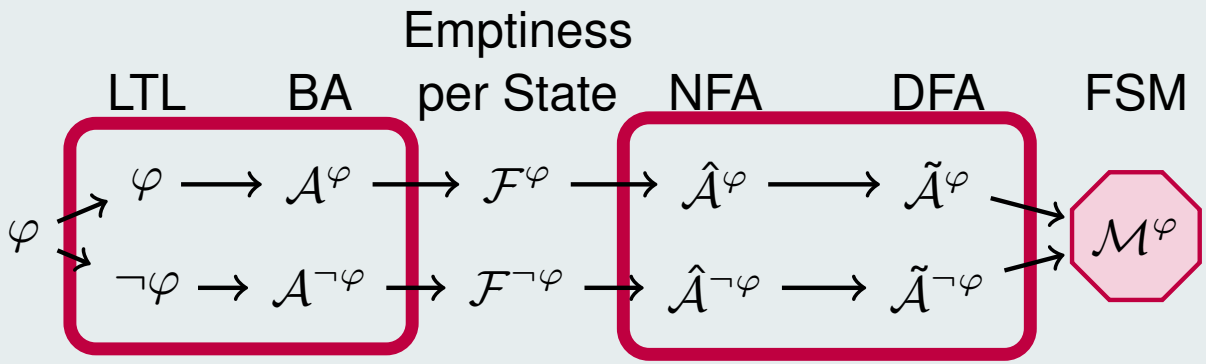
Complexity

The construction



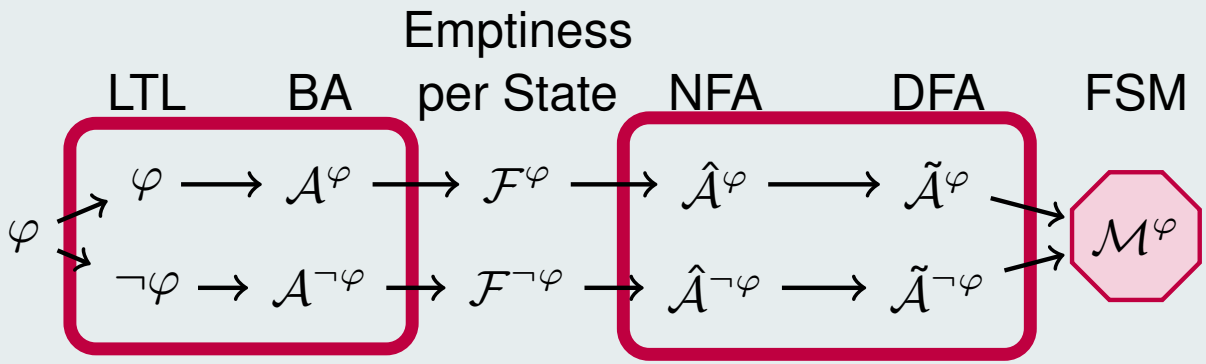
Complexity

The construction



Complexity

The construction

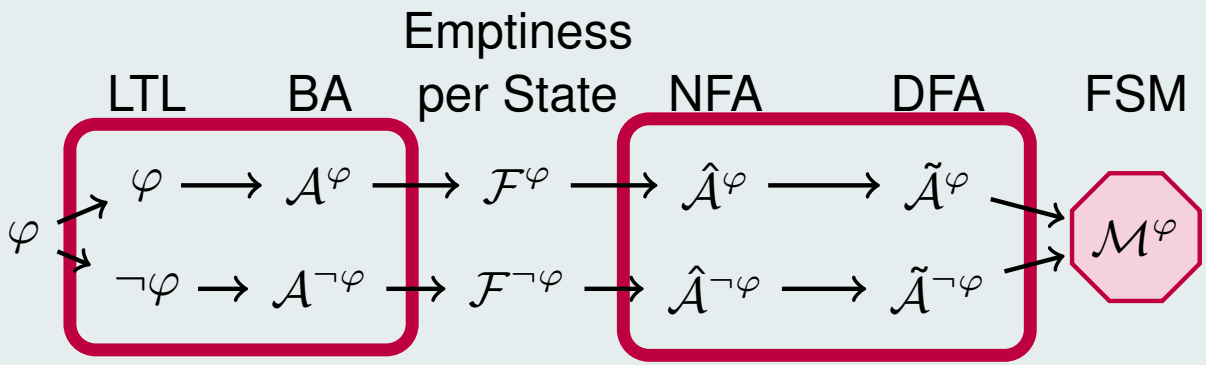


Complexity

$$|M| \in 2^{2^{O(|\varphi|)}}$$

Complexity

The construction



Complexity

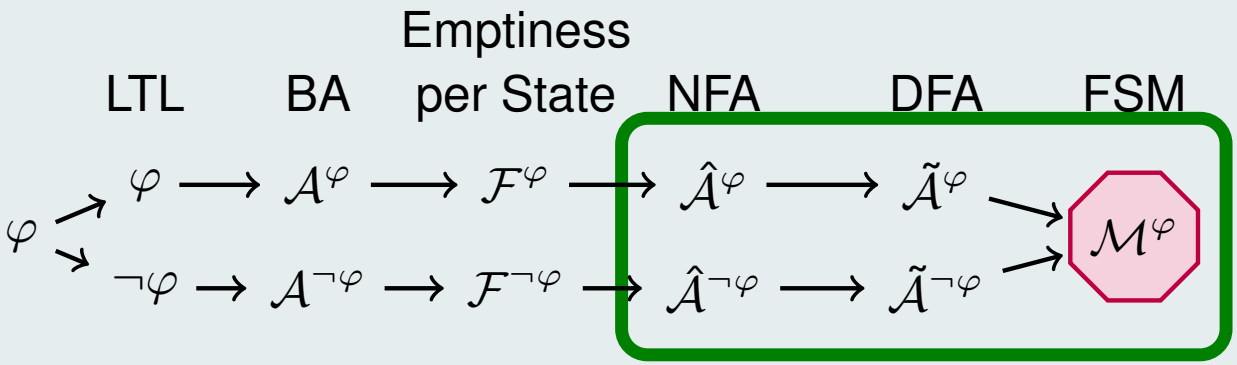
$$|M| \in 2^{2^{O(|\varphi|)}}$$

Optimal result!

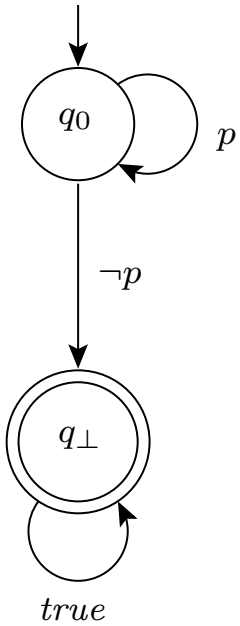
FSM can be minimised (Myhill-Nerode)

On-the-fly Construction

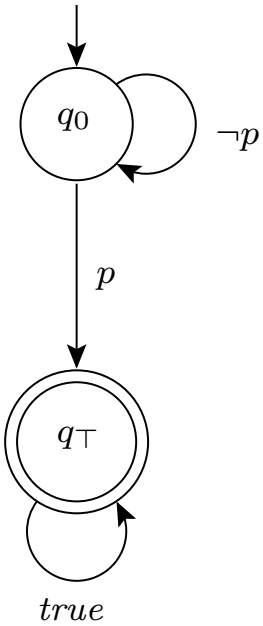
The construction



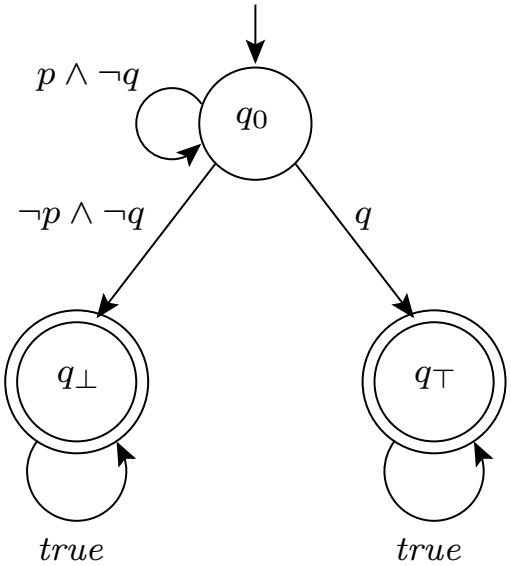
Examples



$\varphi \equiv \Box p$

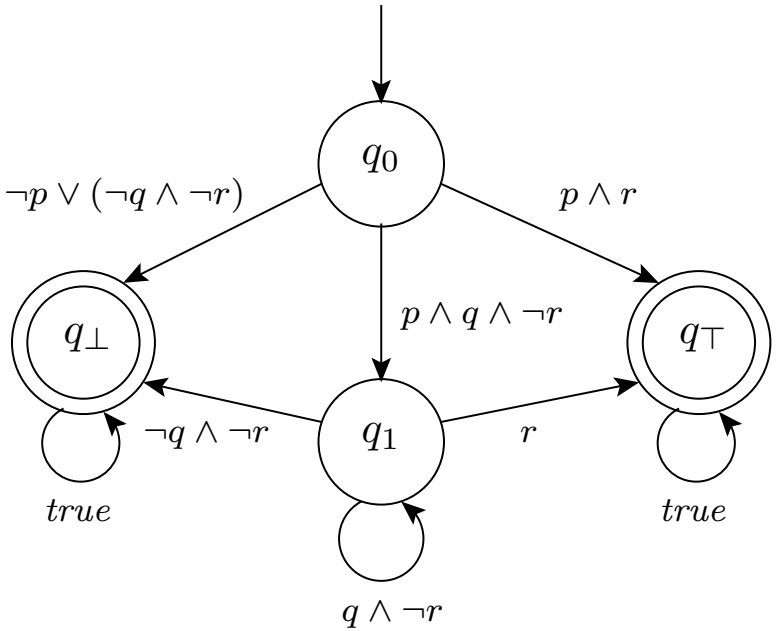


$\varphi \equiv \Diamond p$



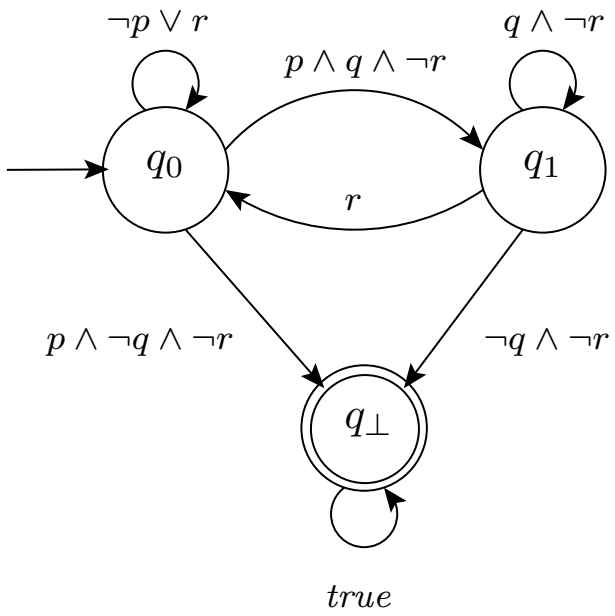
$\varphi \equiv p \, \mathbf{U} \, q$

Examples



$$\varphi = p \wedge (q \,\mathbf{U}\, r)$$

Examples



$$\varphi = \Box(p \Rightarrow (q \,\mathbf{U}\, r))$$

Outline

Runtime Verification

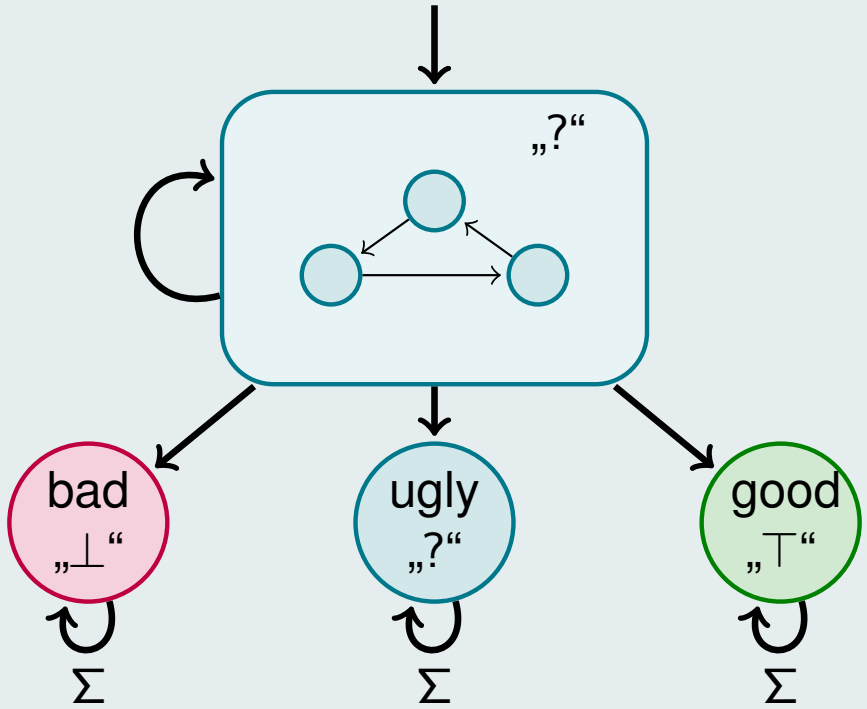
Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

Monitorability

When does anticipation help?

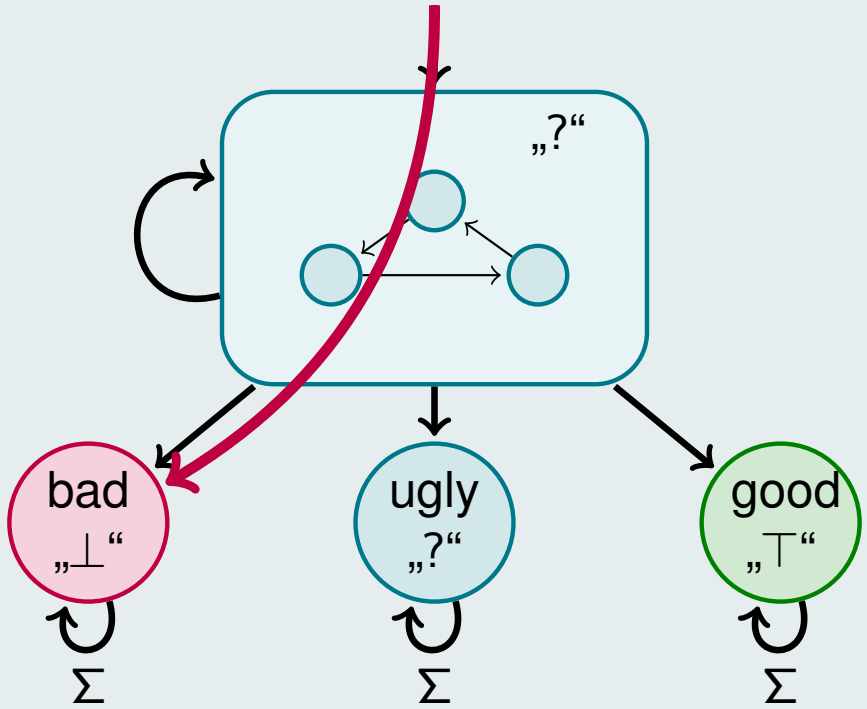
Monitors revisited

Structure of Monitors



Monitors revisited

Structure of Monitors



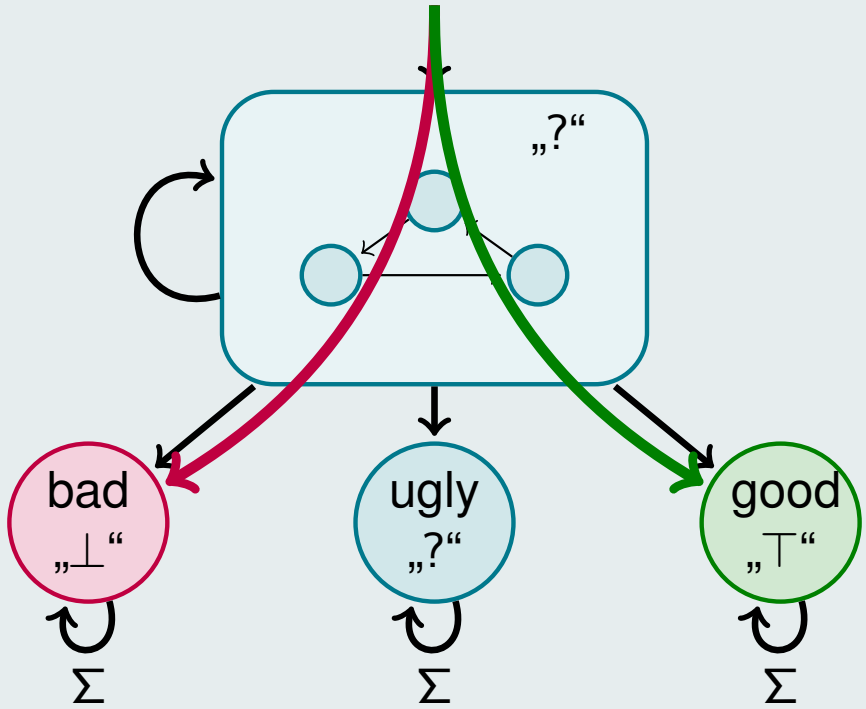
Classification of Prefixes of Words

- Bad prefixes

[Kupferman & Vardi'01]

Monitors revisited

Structure of Monitors

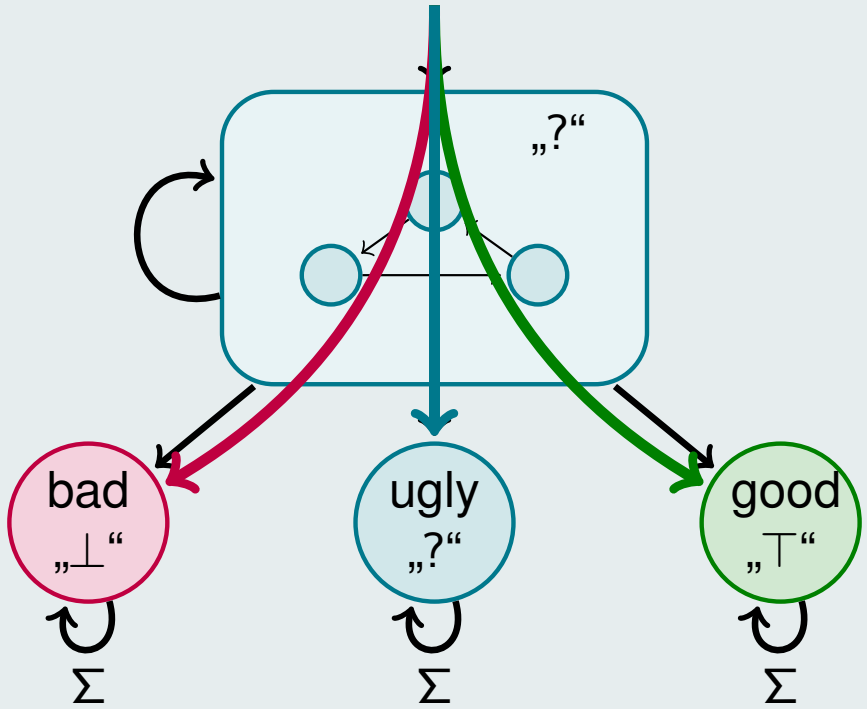


Classification of Prefixes of Words

- Bad prefixes [Kupferman & Vardi'01]
- Good prefixes [Kupferman & Vardi'01]

Monitors revisited

Structure of Monitors



Classification of Prefixes of Words

- Bad prefixes [Kupferman & Vardi'01]
- Good prefixes [Kupferman & Vardi'01]
- Ugly prefixes

Monitorable

Non-Monitorable

φ is non-monitorable after u, if u cannot be extended to a bad oder good prefix.

Monitorable

φ is monitorable if there is no such u.

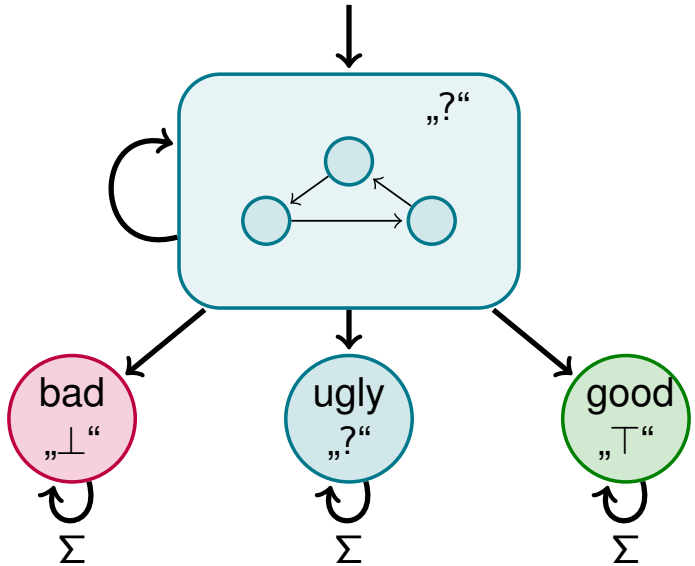
Monitorable

Non-Monitorable

φ is non-monitorable after u, if u cannot be extended to a bad oder good prefix.

Monitorable

φ is monitorable if there is no such u.



Monitorable Properties

Safety Properties

Monitorable Properties

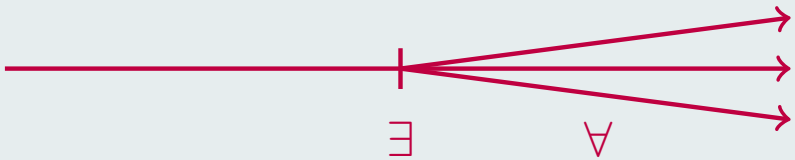
Safety Properties

Monitorable Properties

Safety Properties

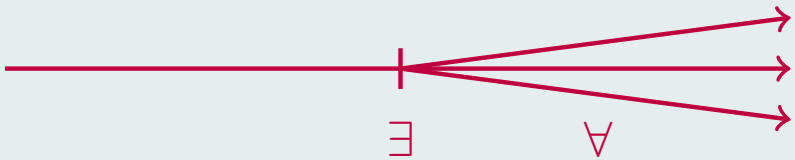
Monitorable Properties

Safety Properties



Monitorable Properties

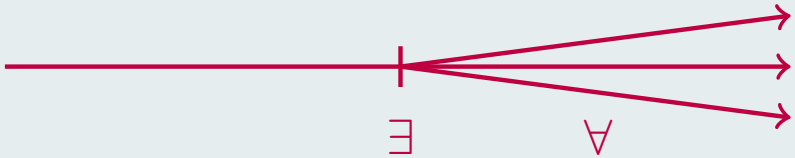
Safety Properties



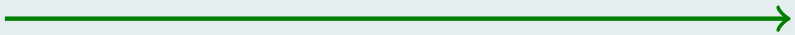
Co-Safety Properties

Monitorable Properties

Safety Properties

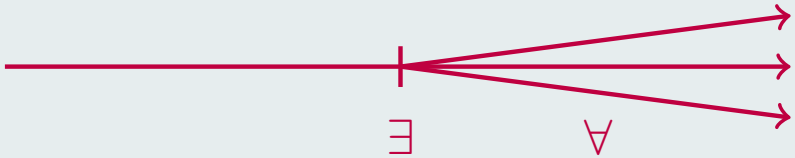


Co-Safety Properties



Monitorable Properties

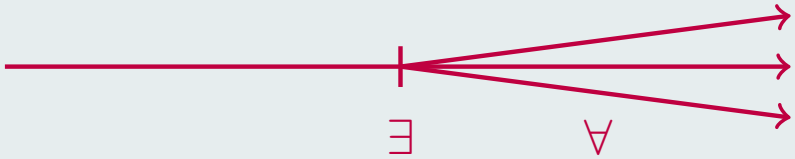
Safety Properties



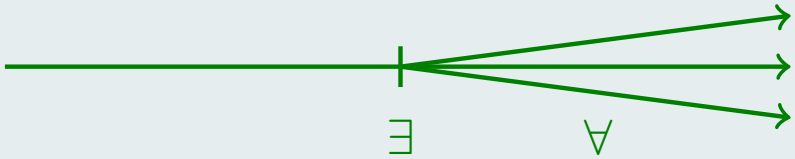
Co-Safety Properties

Monitorable Properties

Safety Properties

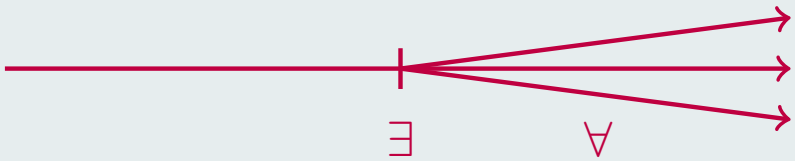


Co-Safety Properties

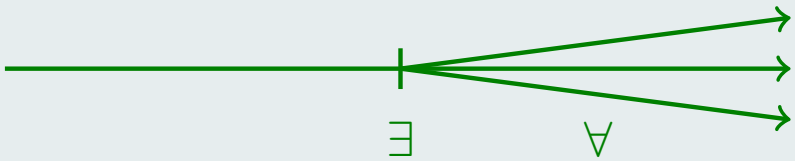


Monitorable Properties

Safety Properties



Co-Safety Properties



Note

Safety and Co-Safety Properties are monitorable.

Safety- and Co-Safety-Properties

Theorem

The class of monitorable properties

- comprises safety- and co-safety properties, but
- is strictly larger than their union.

Proof

Consider $((p \vee q) U r) \vee G p$

Outline

Runtime Verification

Runtime Verification for LTL
- LTL over Finite, Non-Completed Words: Impartiality
- LTL over Non-Completed Words: Anticipation
- LTL over Infinite Words: With Anticipation
- Monitorable Properties
- LTL wrap-up

Intermediate Summary

Semantics

- completed traces
  - two valued semantics
- non-completed traces
  - Impartiality
    - at least three values
  - Anticipation
    - finite traces
    - infinite traces
    - . . .
    - monitorability

Monitors

- left-to-right
- time versus space trade-off
  - rewriting
  - alternating automata
  - non-deterministic automata
  - deterministic automata