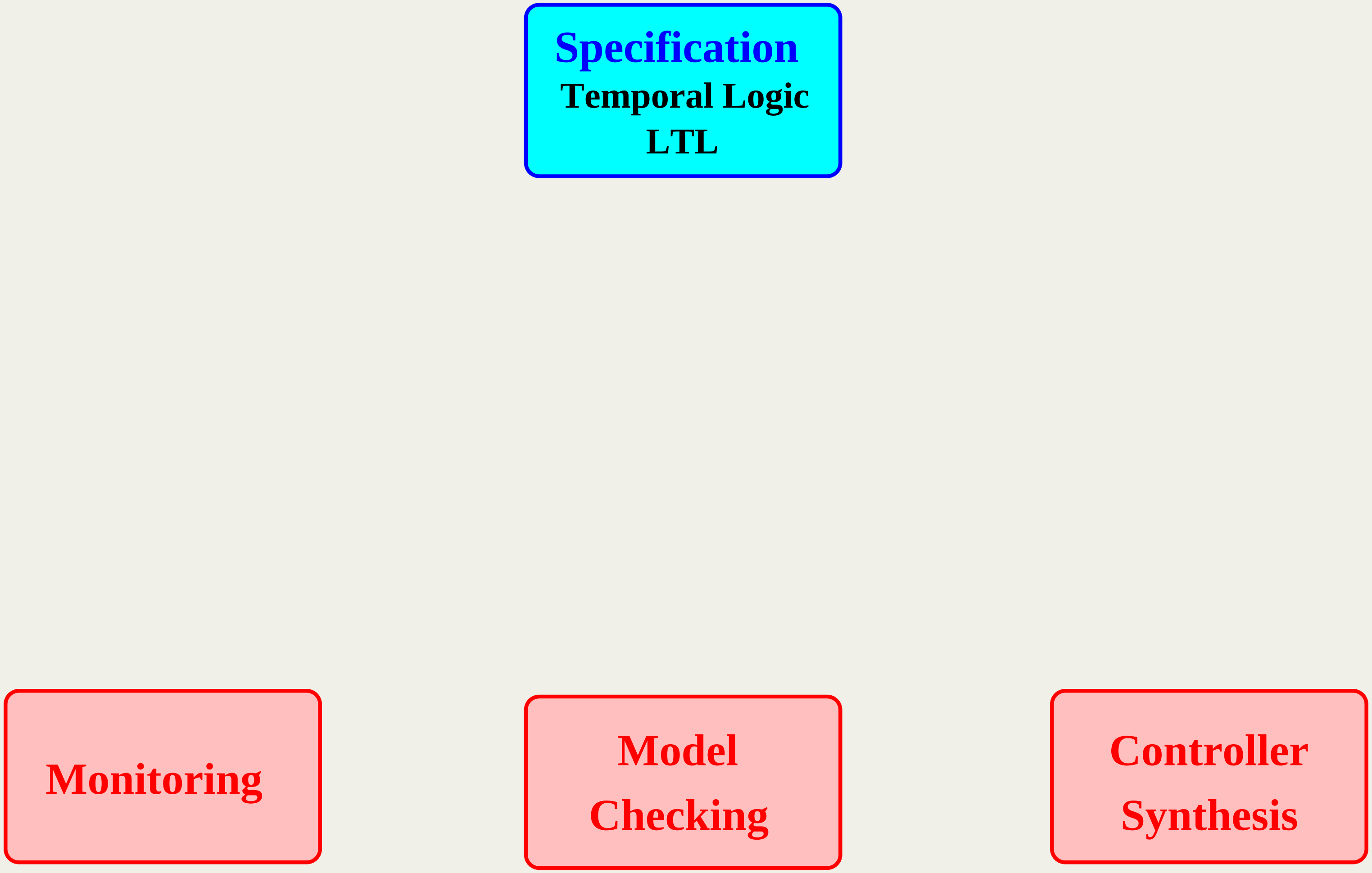
From MTL to Deterministic Timed Automata

Dejan Nickovic
IST Austria

Nir Piterman
Imperial College London
(University of Leicester)

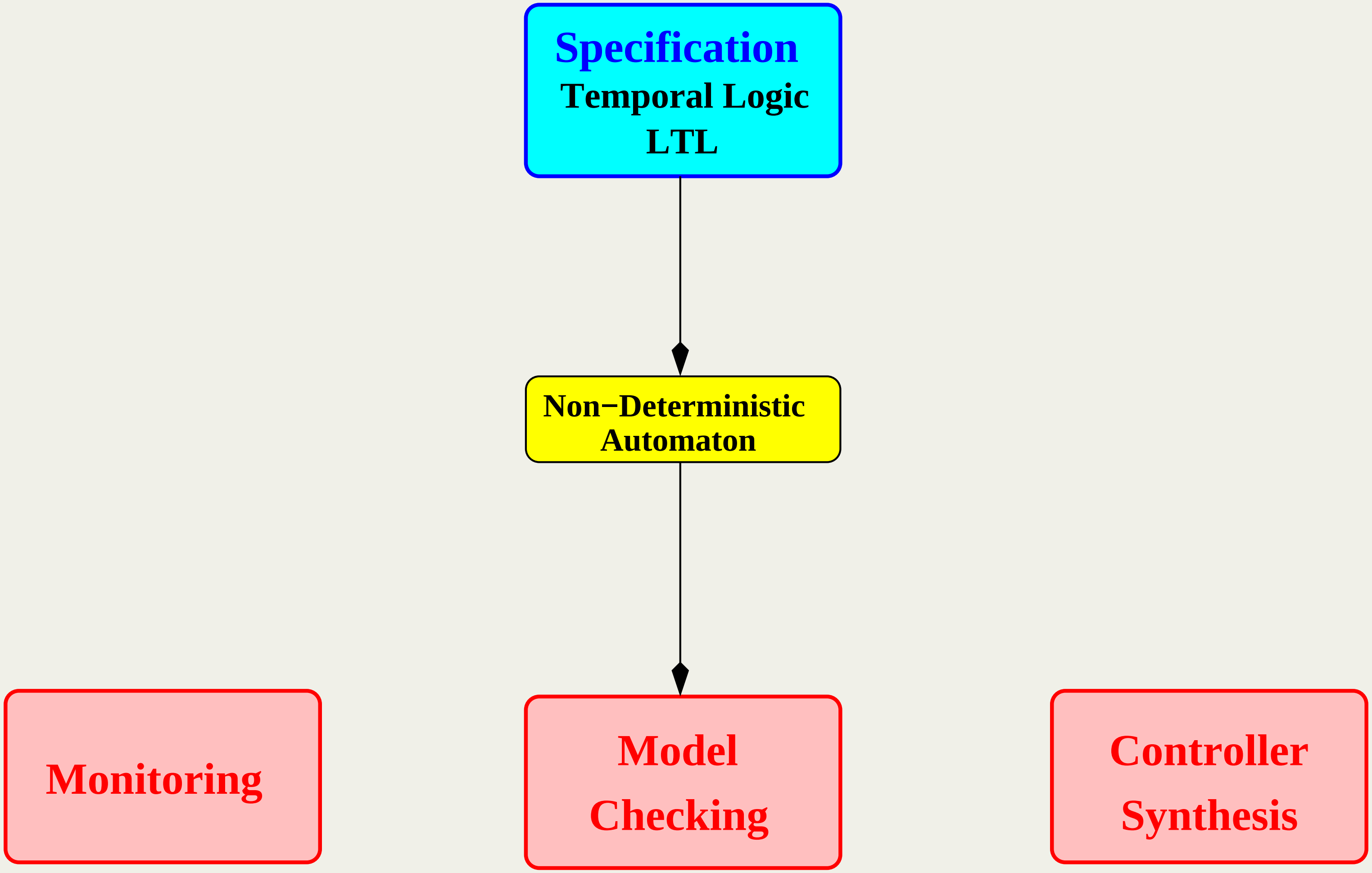
Introduction

Property-based analysis and synthesis of **digital** systems



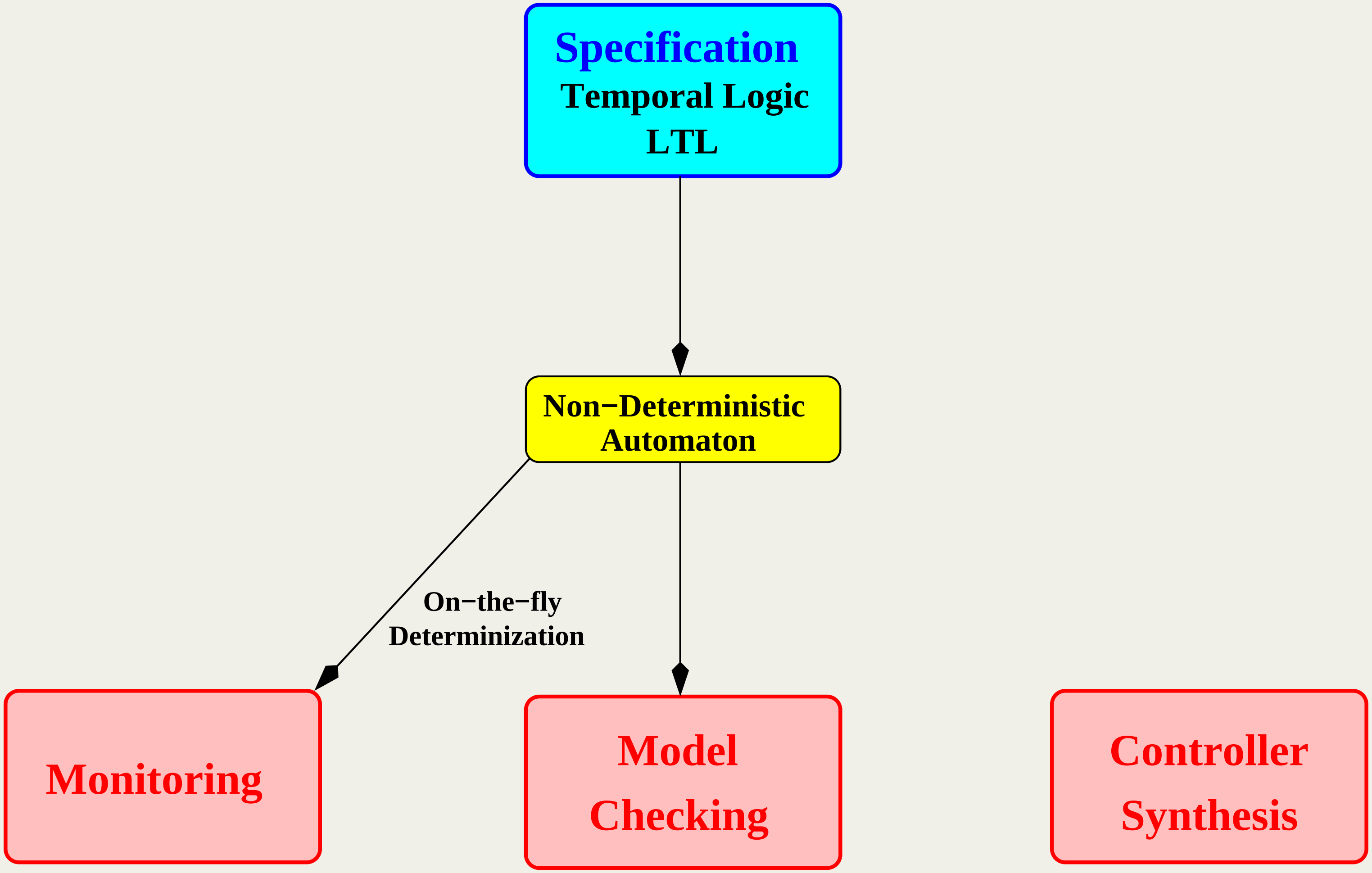
Introduction

Property-based analysis and synthesis of **digital** systems

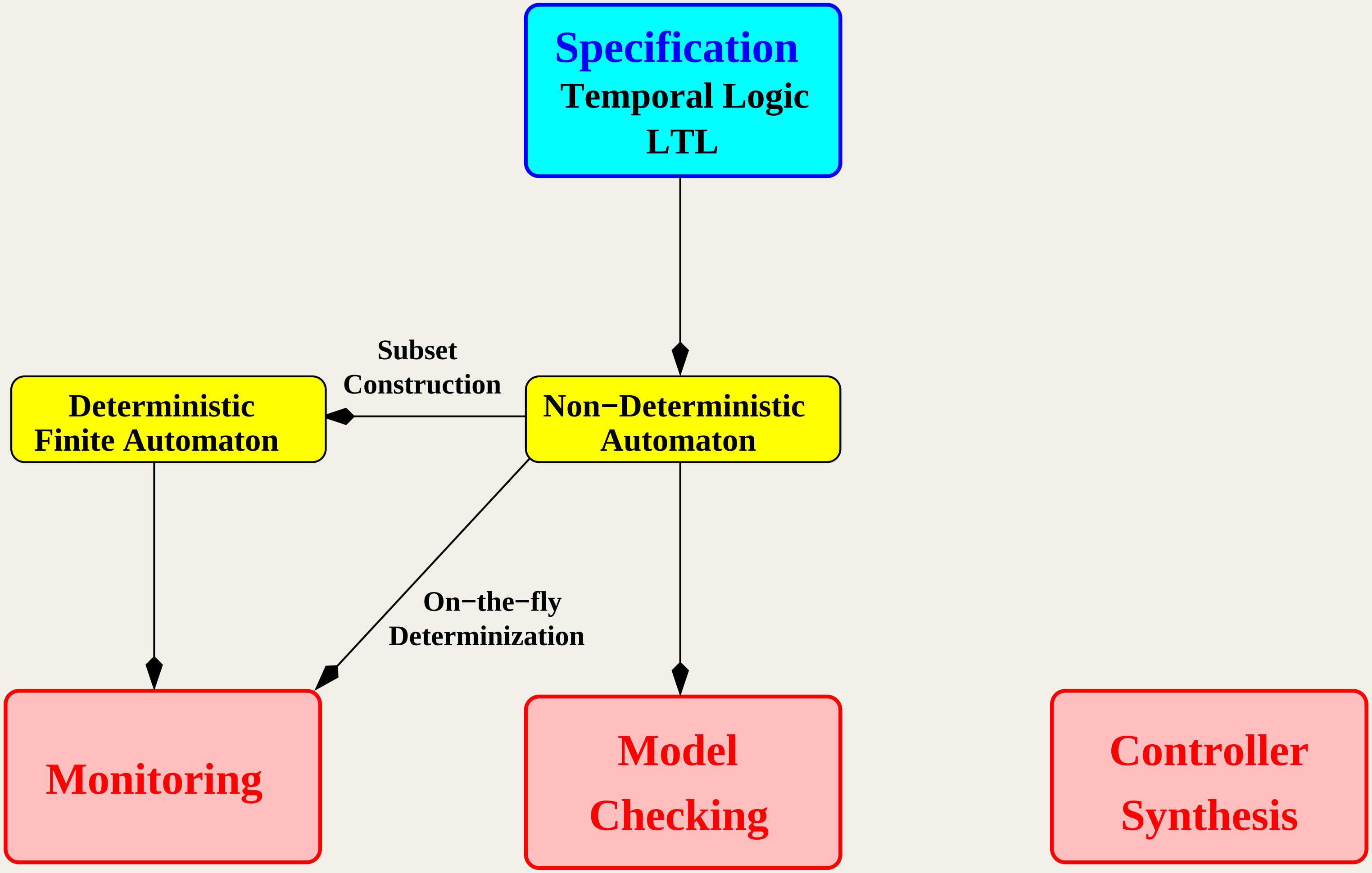


Introduction

Property-based analysis and synthesis of **digital** systems

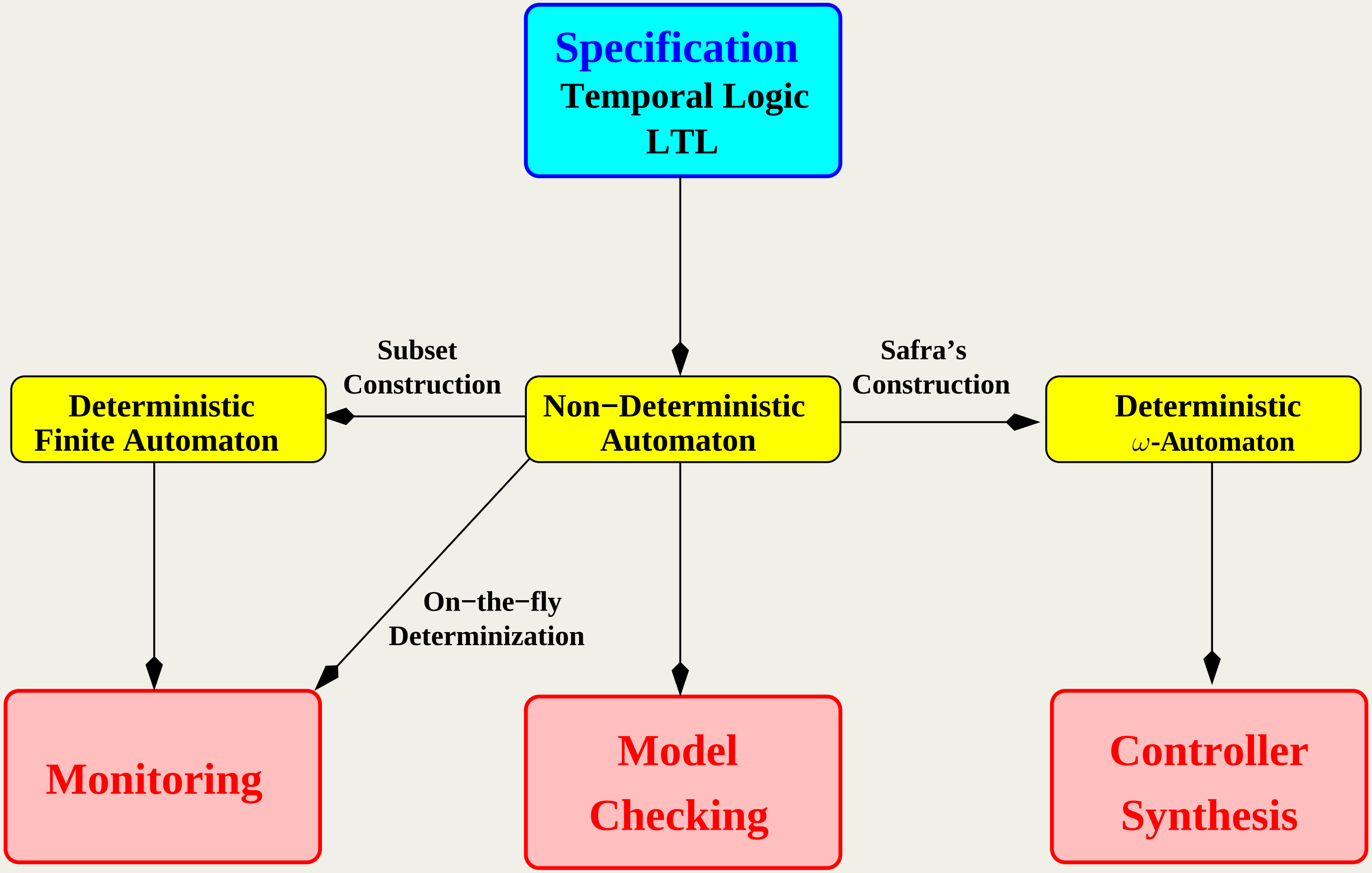


Property-based analysis and synthesis of **digital** systems



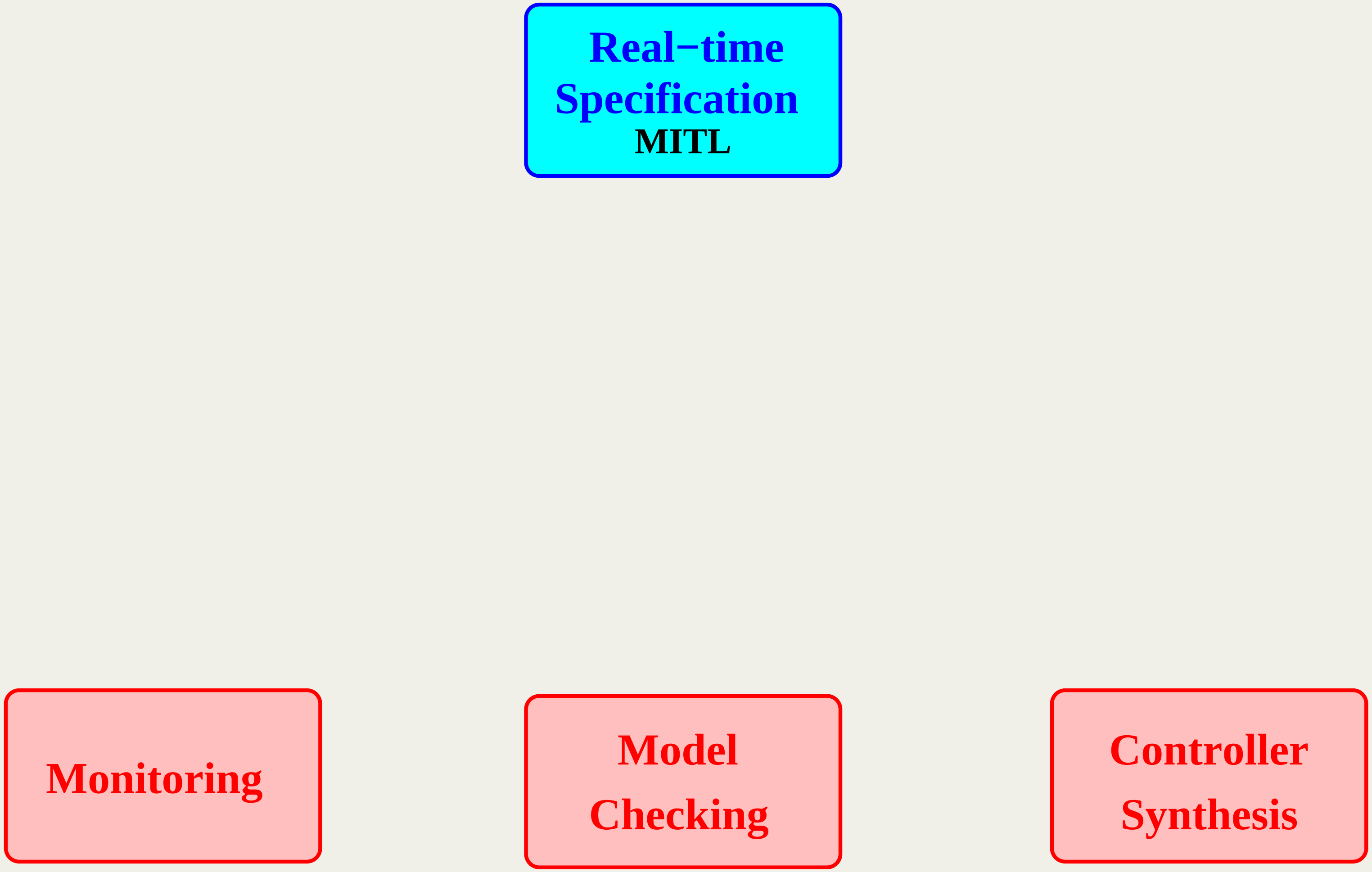
Introduction

Property-based analysis and synthesis of **digital** systems

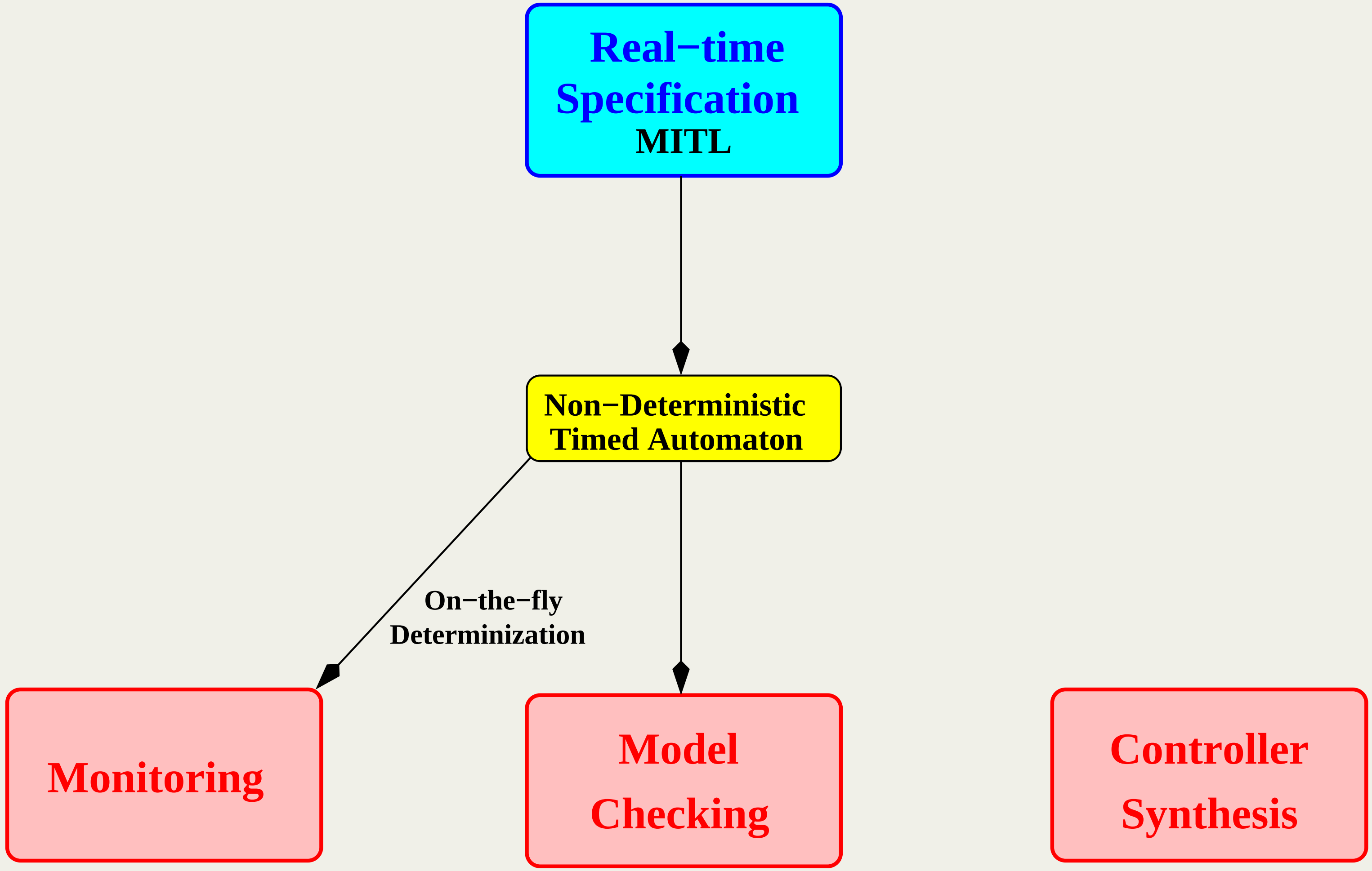


Introduction

Property-based analysis and synthesis of **real-time** systems

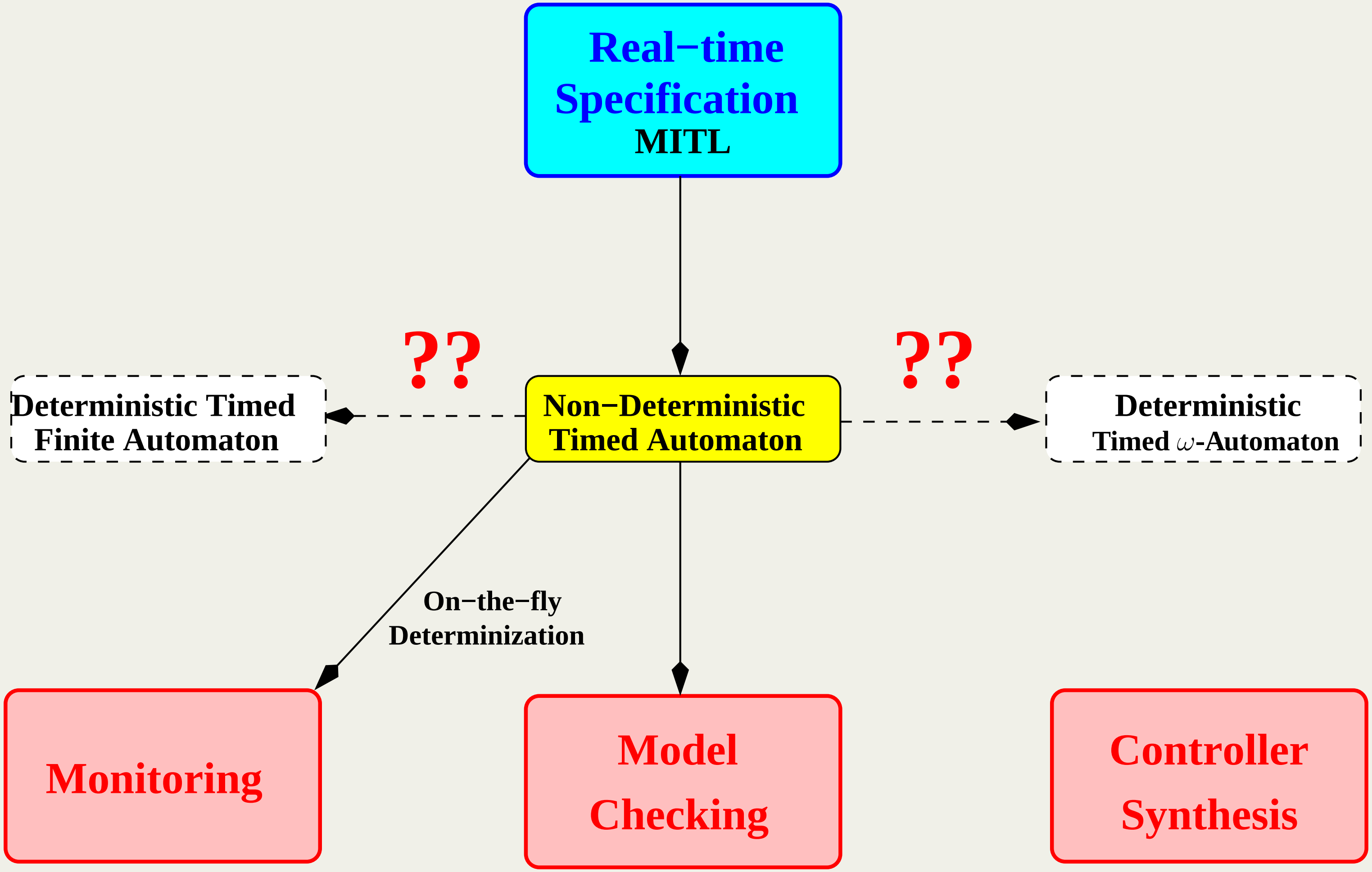


Property-based analysis and synthesis of **real-time** systems



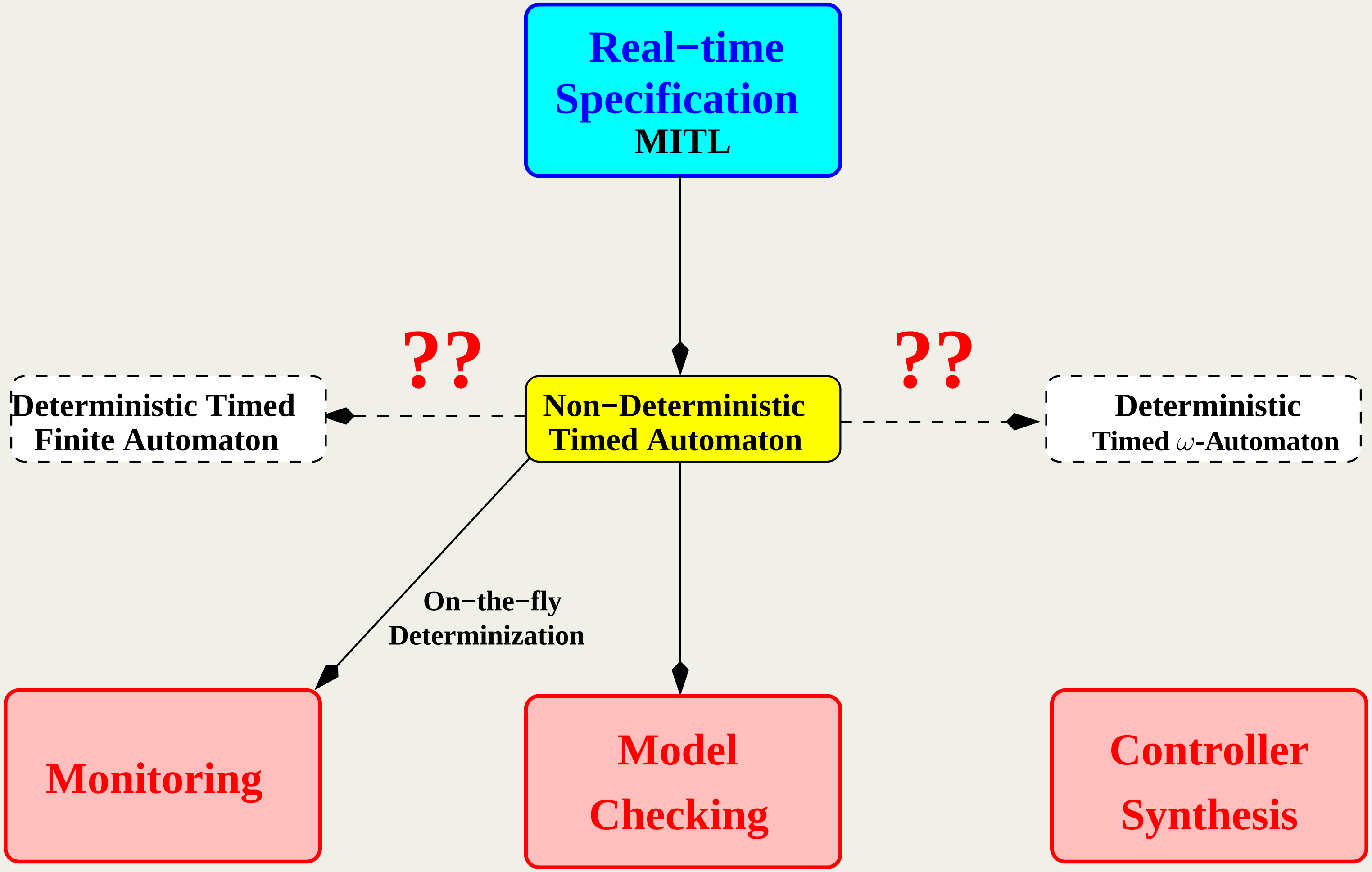
Introduction

Property-based analysis and synthesis of **real-time** systems



Introduction

Property-based analysis and synthesis of **real-time** systems



Timed automata are **non-determinizable** in general!!

Metric Temporal Logic - MTL

- AP - set of atomic propositions
- Signal over AP - $w\ :\ \mathbb{R}_{\geq 0} \rightarrow 2^{AP}$
- w_p - projection of w to proposition $p \in AP$

Syntax:

$$\varphi :== p \mid \neg\varphi_1 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathcal{U}_I \varphi_2$$

where p belongs to the set AP of atomic propositions and I is an interval of the form $[b, b], [a, b], [a, b), (a, b], (a, b), [a, \infty), (a, \infty)$ where $0 \leq a < b$.

- **Derived operators:** $\Diamond_I \varphi = T\ \mathcal{U}_I \varphi$ and $\Box_I \varphi = \neg\Diamond_I \neg\varphi$
- MITL - restricion of MTL to non-singular modalities

MTL - Metric Temporal Logic

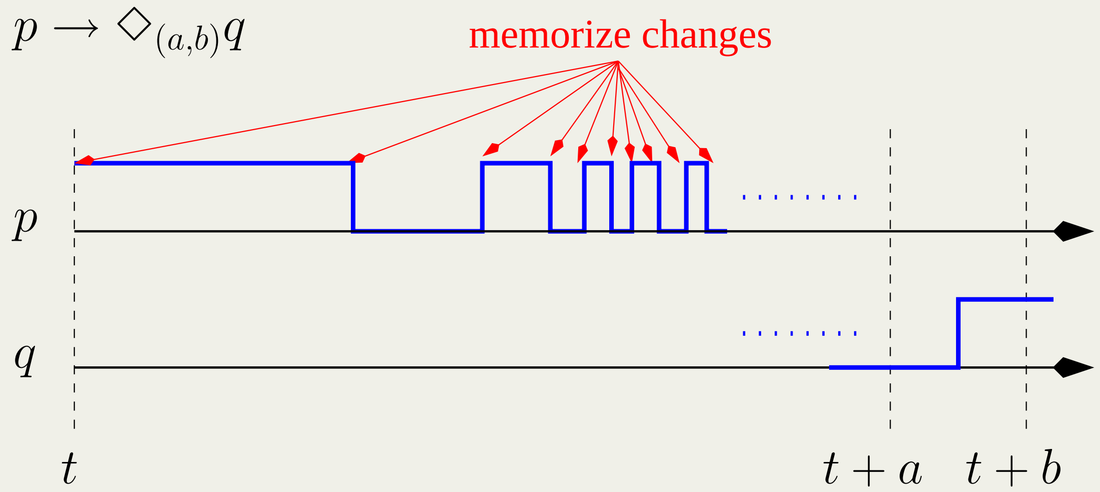
Semantics:

$$
\begin{array}{lll}
(w,t) \models p & \leftrightarrow & w_p[t] = 1 \\
(w,t) \models \neg\varphi & \leftrightarrow & (w,t) \not\models \varphi \\
(w,t) \models \varphi_1 \vee \varphi_2 & \leftrightarrow & (w,t) \models \varphi_1 \text{ or } (w,t) \models \varphi_2 \\
(w,t) \models \varphi_1 \mathcal{U}_I \varphi_2 & \leftrightarrow & \exists t' \in t + I \text{ st } (w,t) \models \varphi_2 \wedge \\
 & & \forall t'' \in (t,t')\ (w,t'') \models \varphi_1
\end{array}
$$

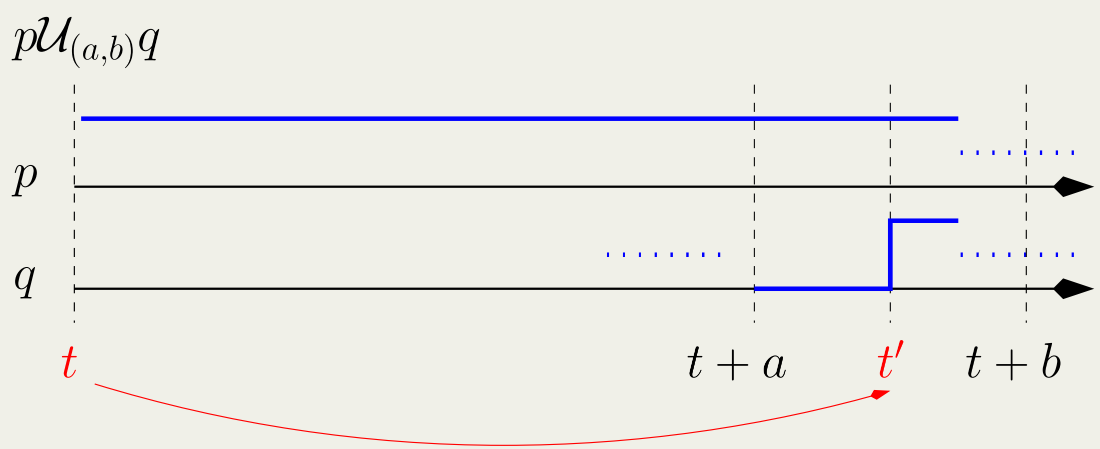
Formula φ satisfied by w if $(w,0) \models \varphi$

MTL and Non-Determinism

1. Unbounded variability

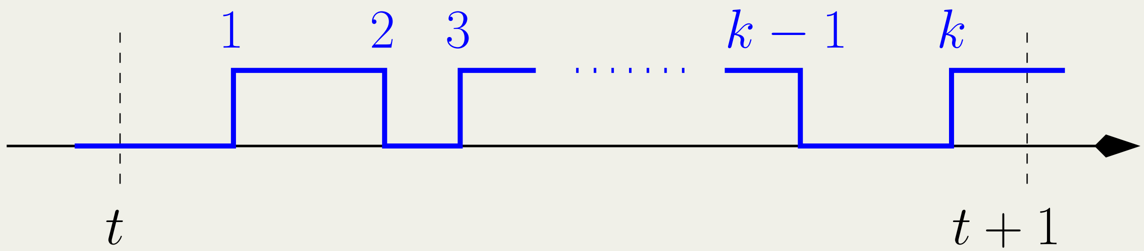


2. Acausality



Signals with Bounded Variability

- Signal w is of **bounded variability** k if for every proposition p, it changes its value at most k times in every interval of length 1



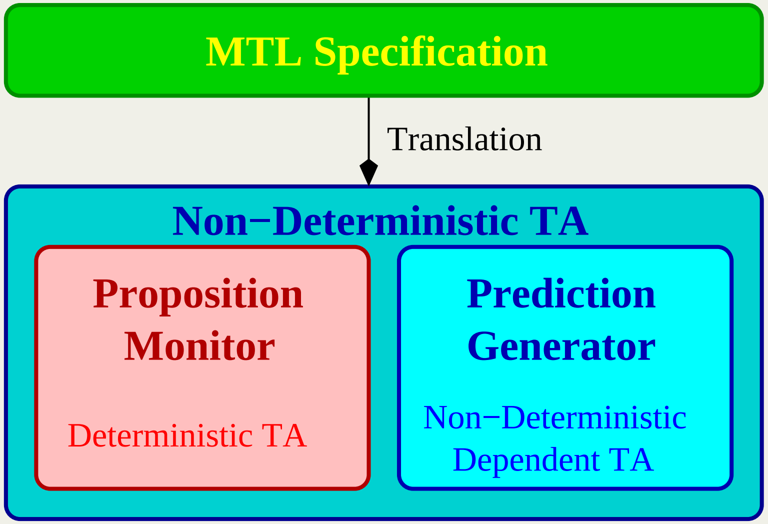
- Reasonable assumption for many applications
  - Almost all systems have a bound on the frequency they operate
- From now on, we assume that every input signal is of bounded variability

From MTL to Deterministic Timed Automata - Overview

- Translation from MTL to deterministic TA assuming **bounded variability** of input signals

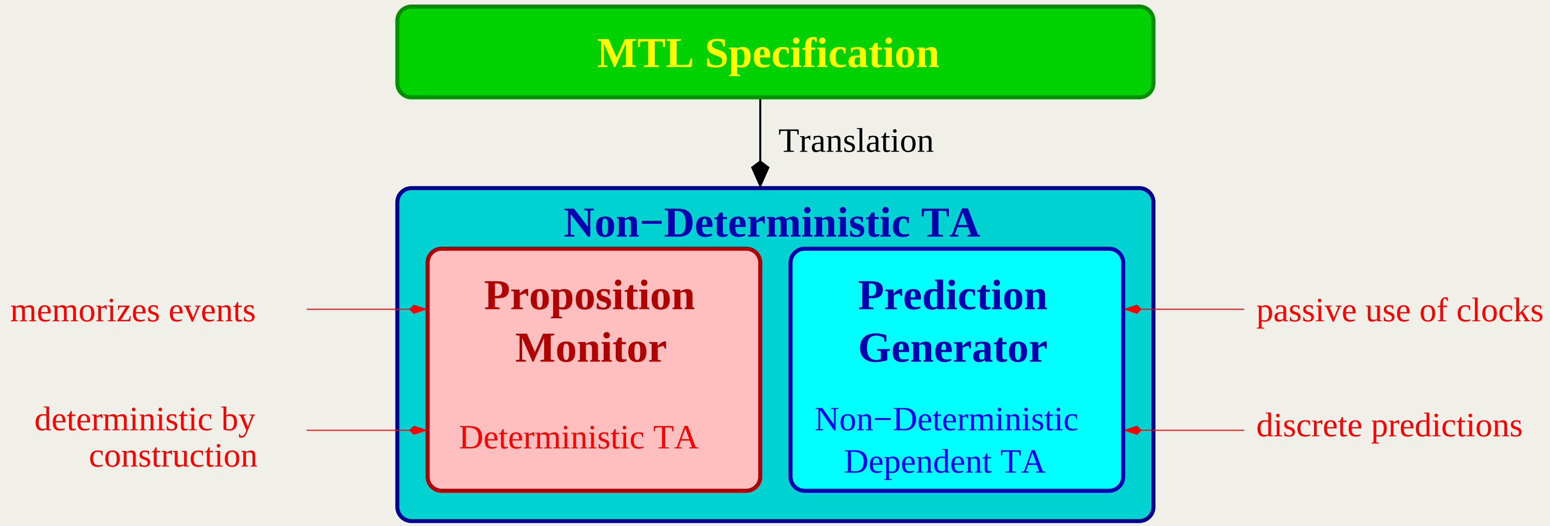
From MTL to Deterministic Timed Automata - Overview

- Translation from MTL to deterministic TA assuming **bounded variability** of input signals



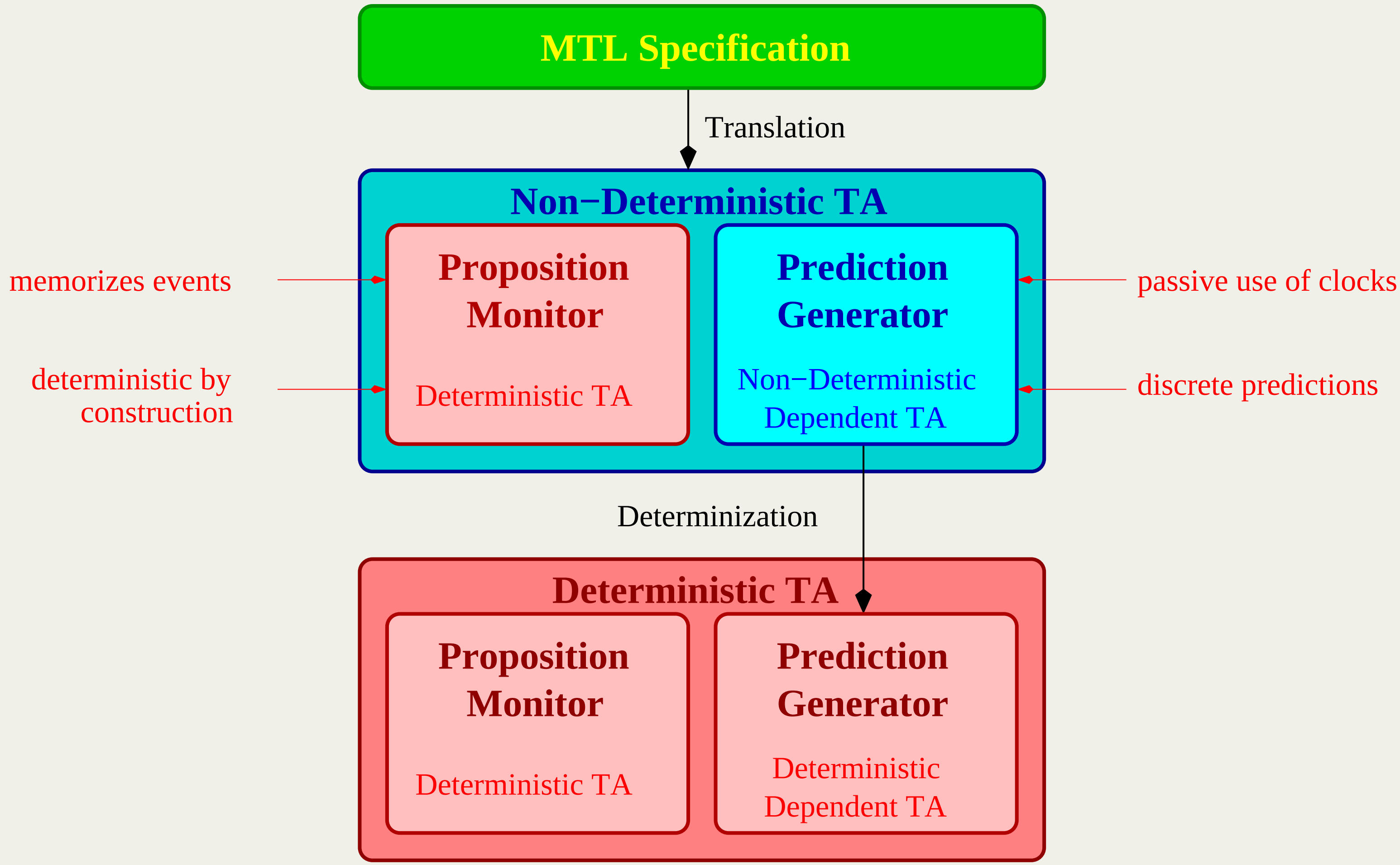
From MTL to Deterministic Timed Automata - Overview

- Translation from MTL to deterministic TA assuming **bounded variability** of input signals



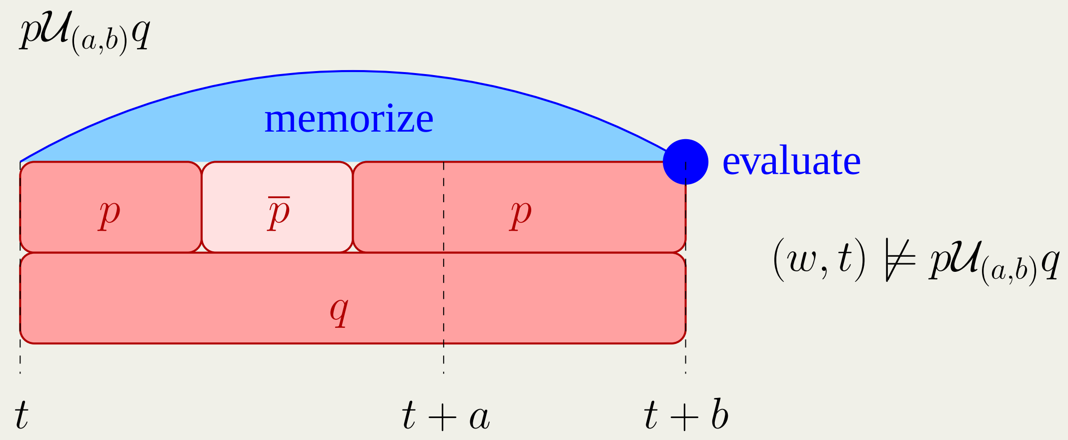
From MTL to Deterministic Timed Automata - Overview

- Translation from MTL to deterministic TA assuming **bounded variability** of input signals



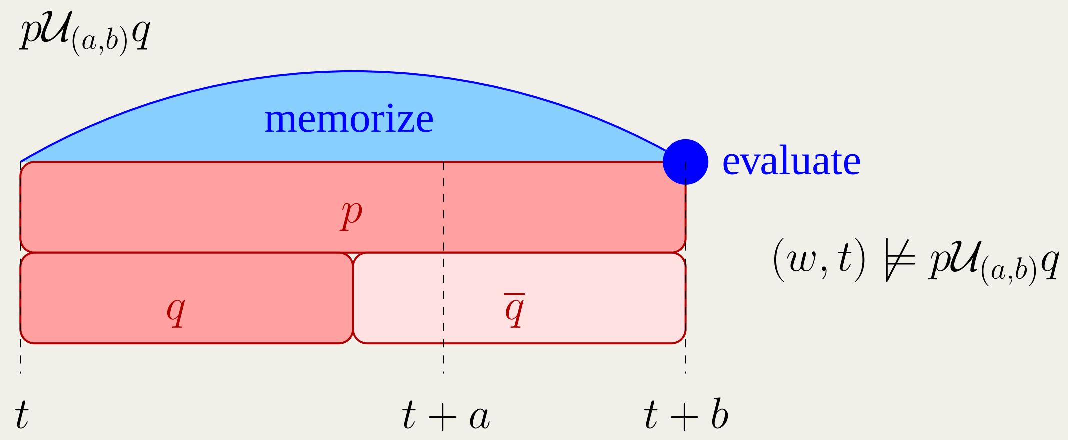
Evaluating MTL Formulas - Overview

- Computation of the truth value of a formula φ at time t with a delay at time $t+f$ where f is a bound



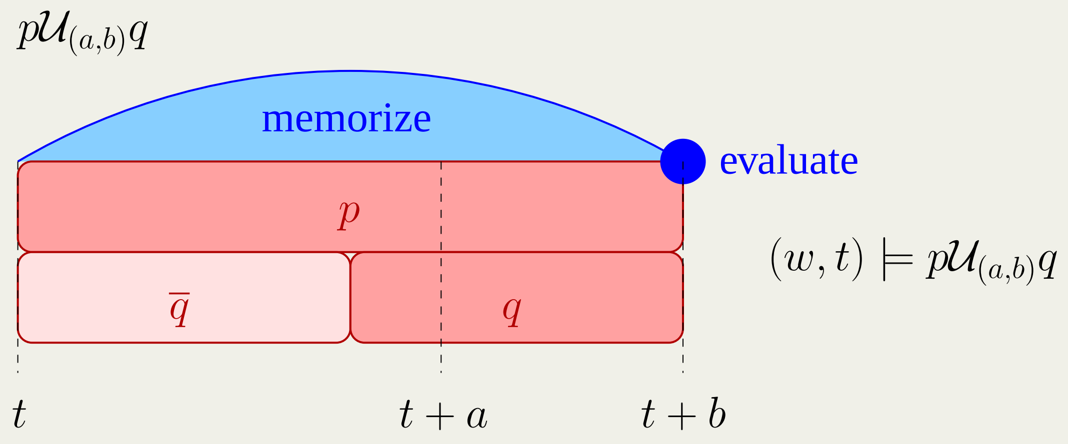
Evaluating MTL Formulas - Overview

- Computation of the truth value of a formula φ at time t with a delay at time $t+f$ where f is a bound



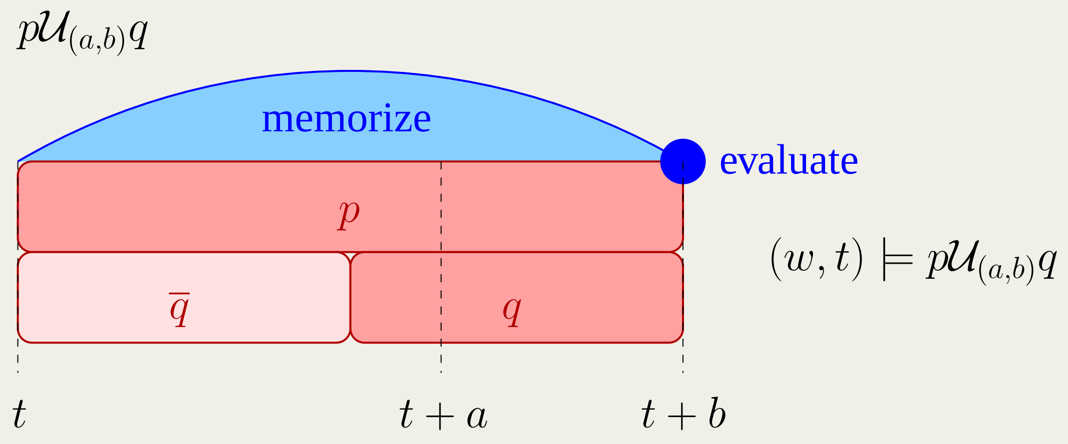
Evaluating MTL Formulas - Overview

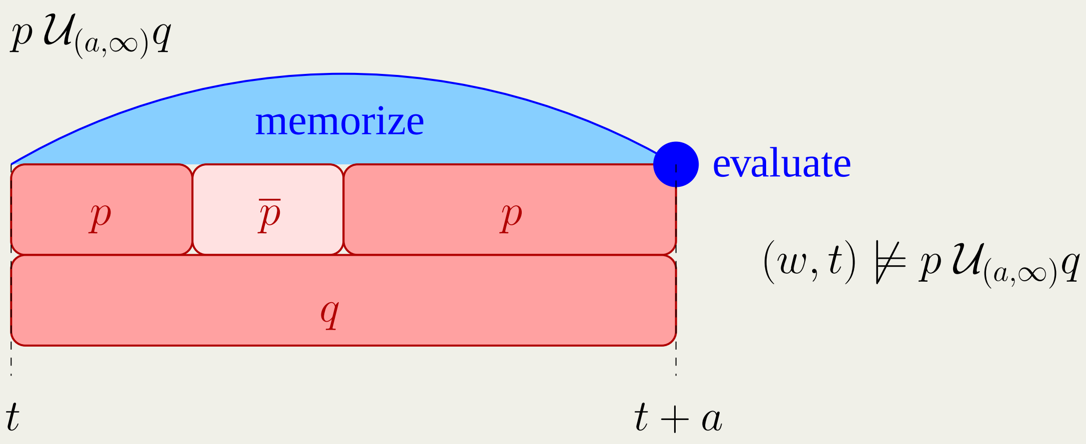
- Computation of the truth value of a formula φ at time t with a delay at time $t + f$ where f is a bound



Evaluating MTL Formulas - Overview

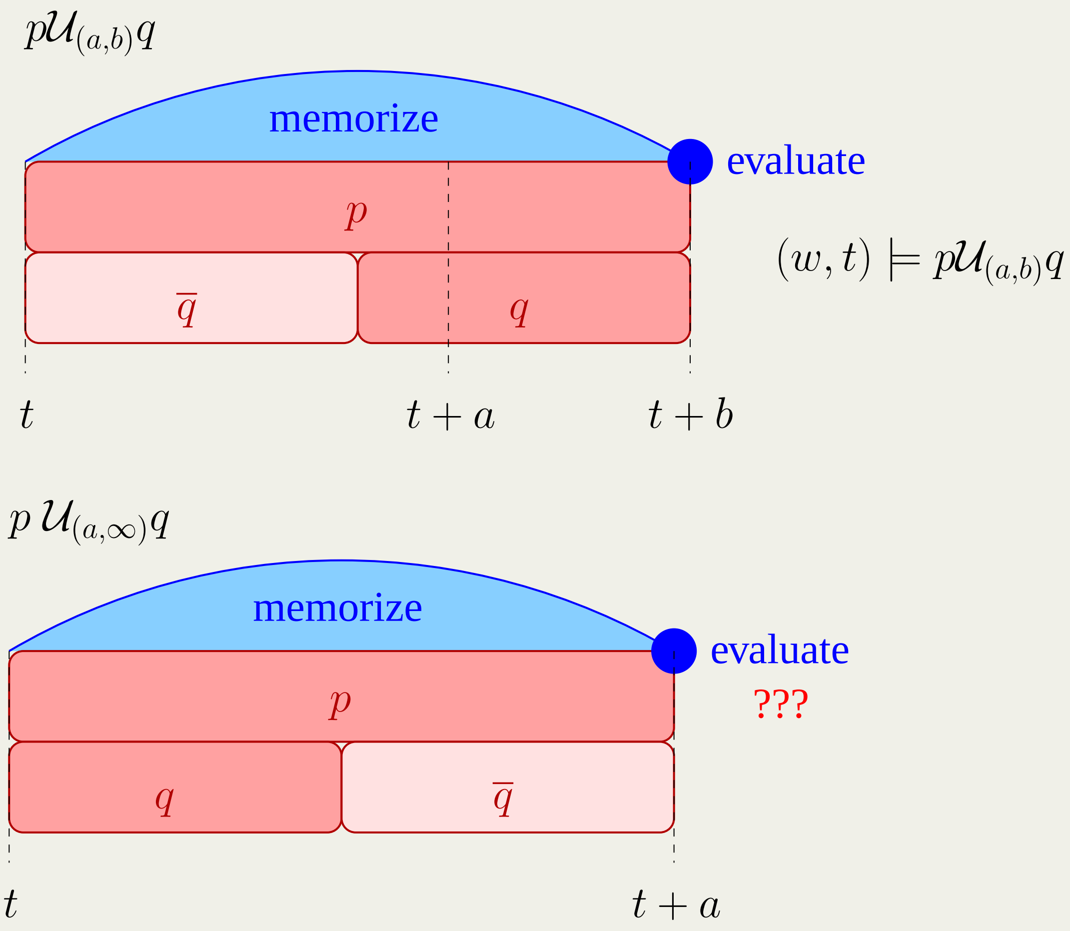
- Computation of the truth value of a formula φ at time t with a delay at time $t+f$ where f is a bound





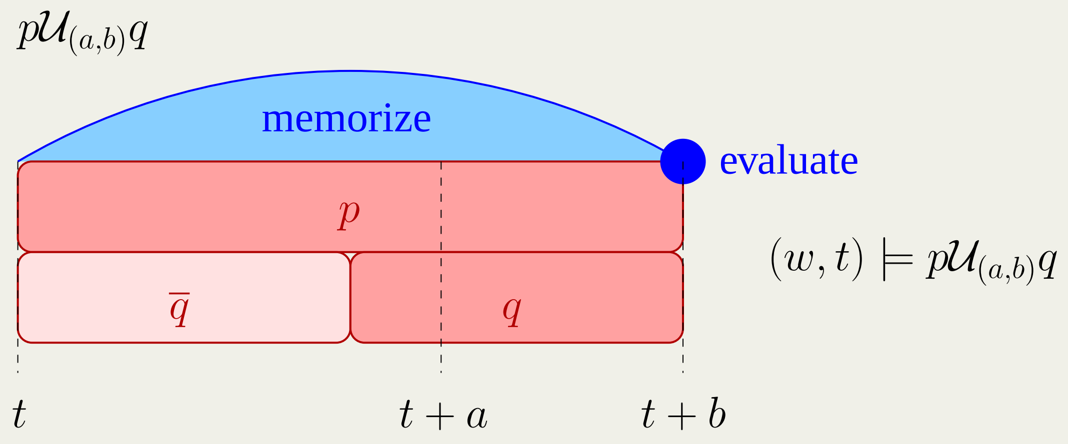
Evaluating MTL Formulas - Overview

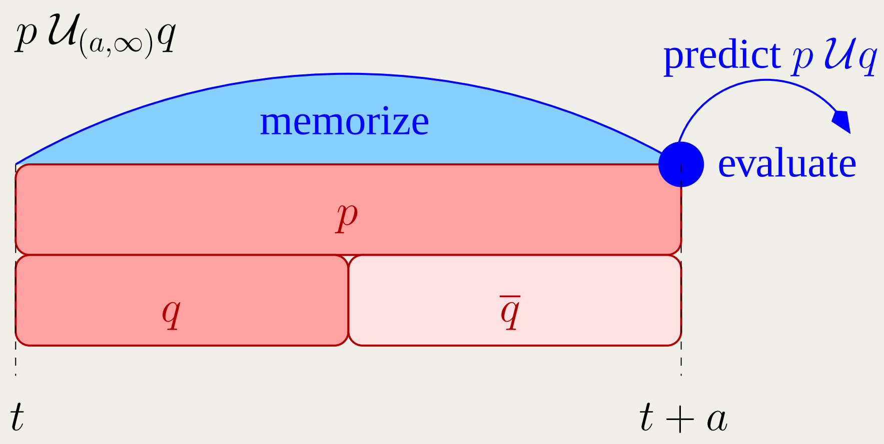
- Computation of the truth value of a formula φ at time t with a delay at time $t + f$ where f is a bound



Evaluating MTL Formulas - Overview

- Computation of the truth value of a formula φ at time t with a delay at time $t + f$ where f is a bound





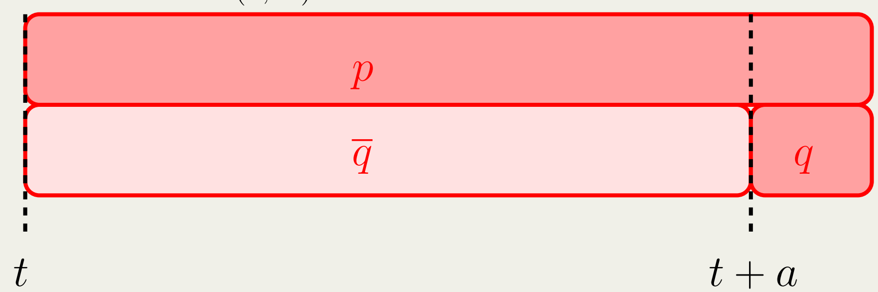
Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \text{future}(\varphi))$

$$
\begin{aligned}
\text{future}(p) &= p \\
\text{future}(\neg\varphi_1) &= \text{future}(\varphi_1) \\
\text{future}(\varphi_1 \vee \varphi_2) &= max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1 \, \mathcal{U}_{(a,b)} \varphi_2) &= b + max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1 \, \mathcal{U}_{(a,\infty)} \varphi_2) &= 2 + a + max(\text{future}(\varphi_1), \text{future}(\varphi_2))
\end{aligned}
$$

- Why 2 additional **lookaheads** for $\text{future}(\varphi_1 \, \mathcal{U}_{(a,\infty)} \varphi_2)$?

$[t, t + a)$ never sufficient to determine whether $p \, \mathcal{U}_{(a,\infty)}$ holds at t



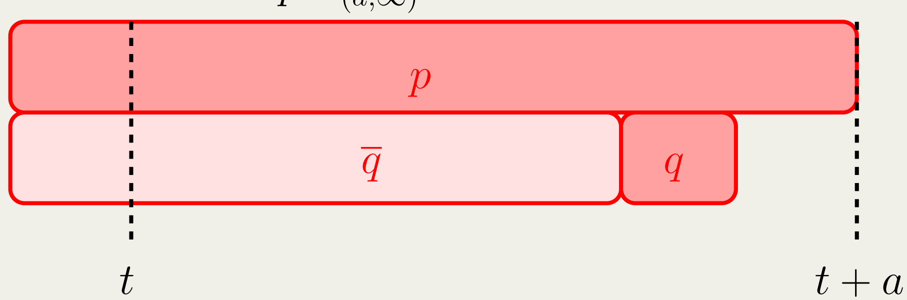
Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \text{future}(\varphi))$

$$
\begin{aligned}
\text{future}(p) &= p \\
\text{future}(\neg\varphi_1) &= \text{future}(\varphi_1) \\
\text{future}(\varphi_1 \vee \varphi_2) &= max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1\, \mathcal{U}_{(a,b)}\varphi_2) &= b + max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1\, \mathcal{U}_{(a,\infty)}\varphi_2) &= 2 + a + max(\text{future}(\varphi_1), \text{future}(\varphi_2))
\end{aligned}
$$

- Why 2 additional **lookaheads** for $\text{future}(\varphi_1\, \mathcal{U}_{(a,\infty)}\varphi_2)$?

$[t, t + a)$ never sufficient to determine whether $p\, \mathcal{U}_{(a,\infty)}$ holds at t



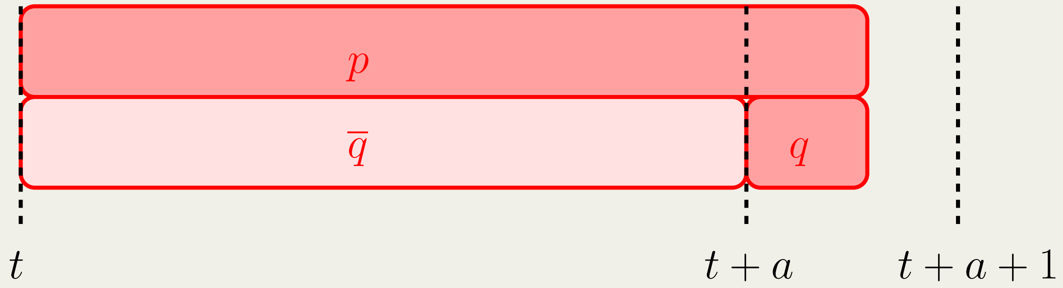
Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \mathsf{future}(\varphi))$

$$
\begin{aligned}
\mathsf{future}(p) &= p \\
\mathsf{future}(\neg\varphi_1) &= \mathsf{future}(\varphi_1) \\
\mathsf{future}(\varphi_1 \vee \varphi_2) &= max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2)) \\
\mathsf{future}(\varphi_1\, \mathcal{U}_{(a,b)} \varphi_2) &= b + max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2)) \\
\mathsf{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2) &= 2 + a + max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2))
\end{aligned}
$$

- Why 2 additional **lookaheads** for $\mathsf{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2)$?

$[t, t + a + 1)$ sometimes sufficient to determine whether $p\, \mathcal{U}_{(a,\infty)}$ holds at t



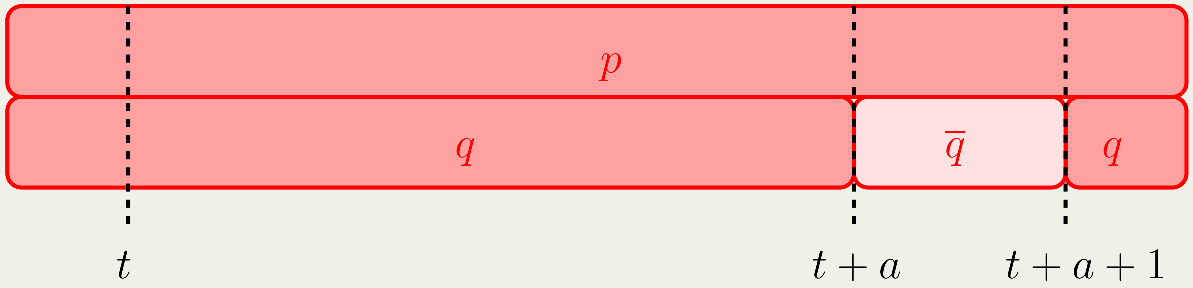
Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \mathsf{future}(\varphi))$

$$
\begin{aligned}
\mathsf{future}(p) &= p \\
\mathsf{future}(\neg\varphi_1) &= \mathsf{future}(\varphi_1) \\
\mathsf{future}(\varphi_1 \vee \varphi_2) &= max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2)) \\
\mathsf{future}(\varphi_1 \, \mathcal{U}_{(a,b)} \varphi_2) &= b + max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2)) \\
\mathsf{future}(\varphi_1 \, \mathcal{U}_{(a,\infty)} \varphi_2) &= 2 + a + max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2))
\end{aligned}
$$

- Why 2 additional **lookaheads** for $\mathsf{future}(\varphi_1 \, \mathcal{U}_{(a,\infty)} \varphi_2)$?

Elimination of 0-duration errors



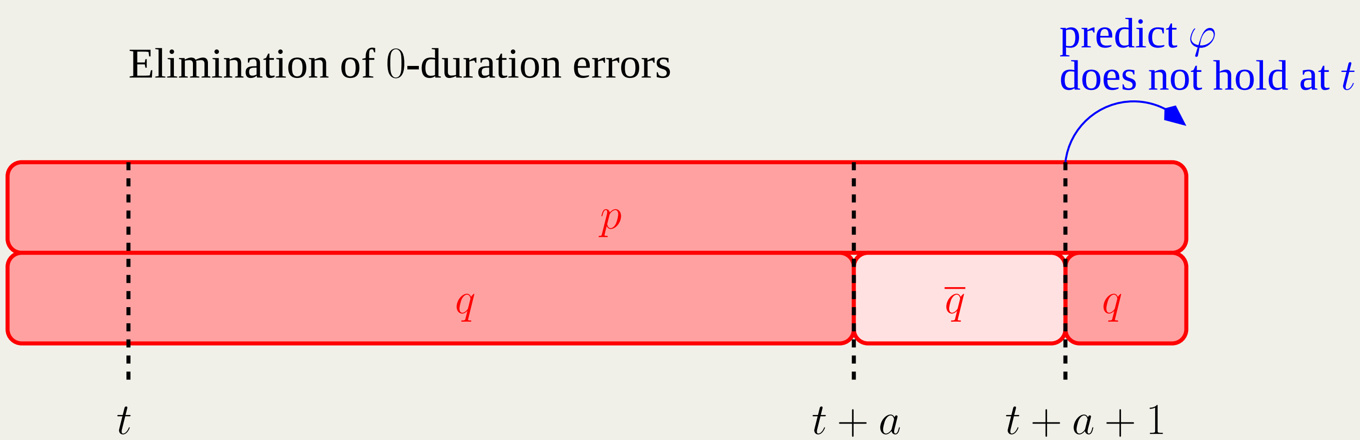
Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \text{future}(\varphi))$

$$
\begin{aligned}
\text{future}(p) &= p \\
\text{future}(\neg\varphi_1) &= \text{future}(\varphi_1) \\
\text{future}(\varphi_1 \vee \varphi_2) &= max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1\, \mathcal{U}_{(a,b)} \varphi_2) &= b + max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2) &= 2 + a + max(\text{future}(\varphi_1), \text{future}(\varphi_2))
\end{aligned}
$$

- Why 2 additional **lookaheads** for $\text{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2)$?

Elimination of 0-duration errors

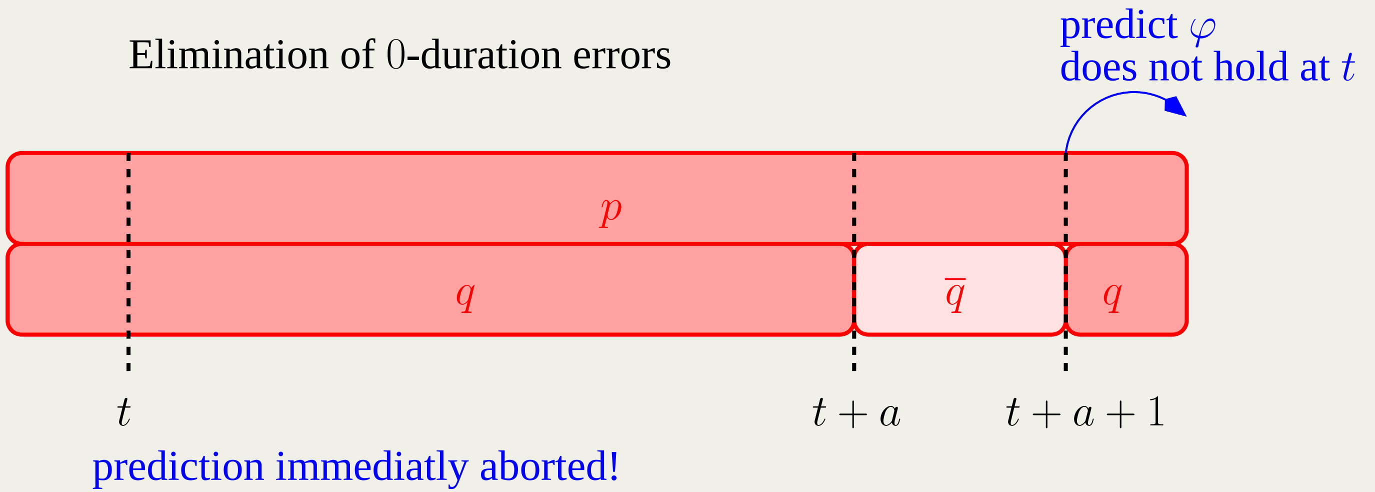


Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \mathsf{future}(\varphi))$

$$
\begin{aligned}
\mathsf{future}(p) &= p \\
\mathsf{future}(\neg\varphi_1) &= \mathsf{future}(\varphi_1) \\
\mathsf{future}(\varphi_1 \vee \varphi_2) &= max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2)) \\
\mathsf{future}(\varphi_1\, \mathcal{U}_{(a,b)} \varphi_2) &= b + max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2)) \\
\mathsf{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2) &= 2 + a + max(\mathsf{future}(\varphi_1), \mathsf{future}(\varphi_2))
\end{aligned}
$$

- Why 2 additional **lookaheads** for $\mathsf{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2)$?

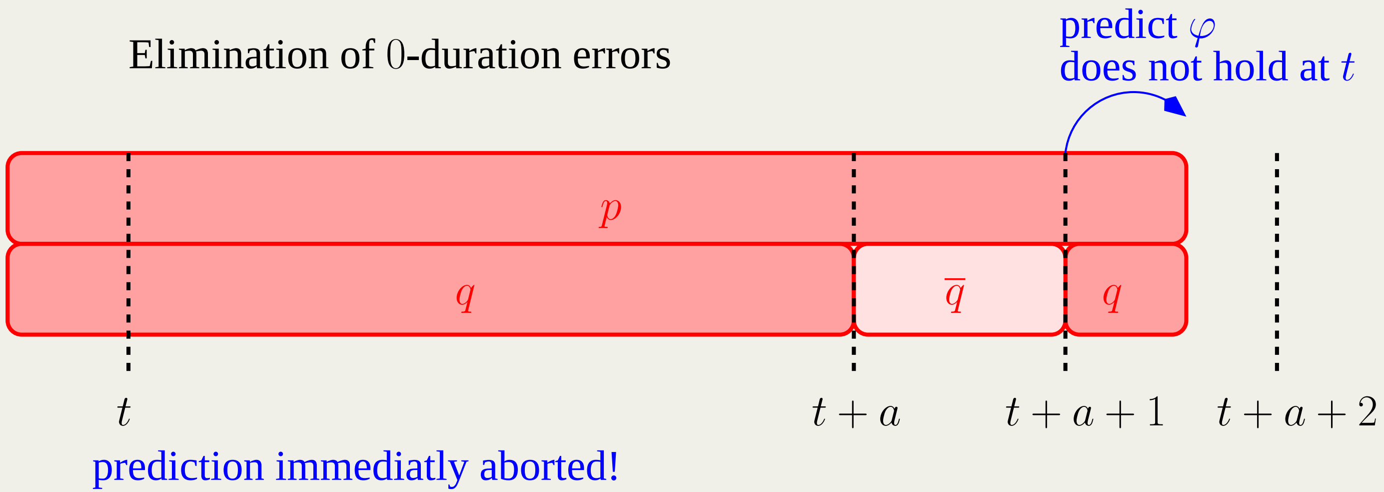


Evaluating MTL Formulas - future Function

- Computation of the truth value of a formula φ at time t by looking in the interval $[t, t + \text{future}(\varphi))$

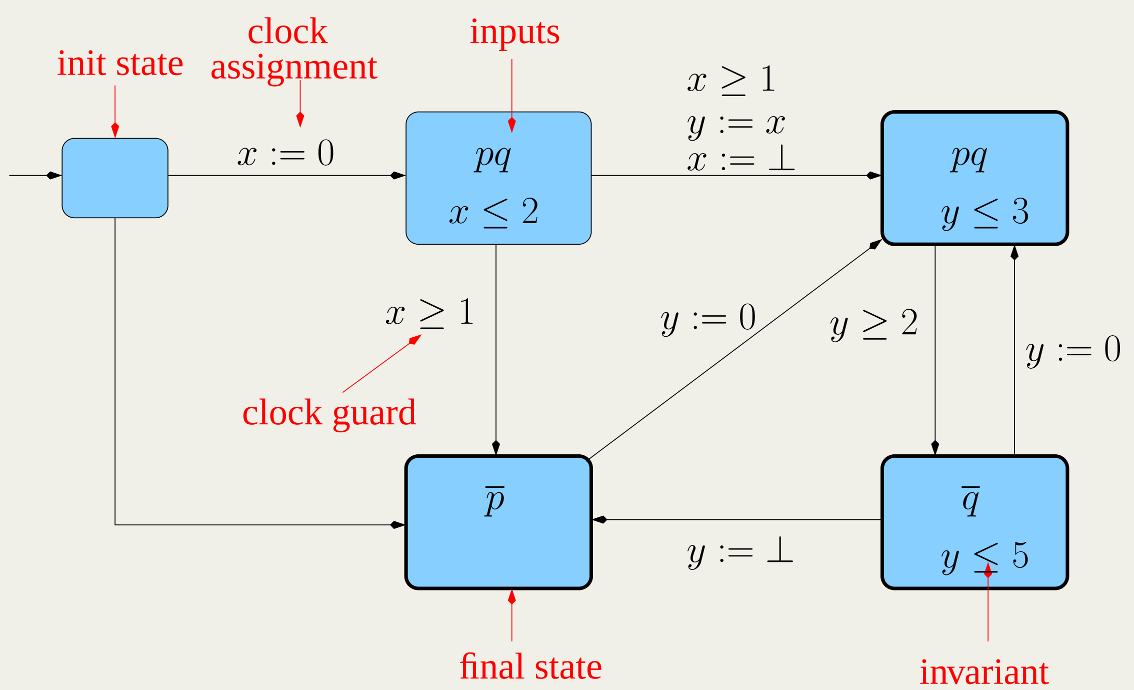
$$
\begin{array}{lcl}
\text{future}(p) & = & p \\
\text{future}(\neg\varphi_1) & = & \text{future}(\varphi_1) \\
\text{future}(\varphi_1 \vee \varphi_2) & = & max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1\, \mathcal{U}_{(a,b)} \varphi_2) & = & b + max(\text{future}(\varphi_1), \text{future}(\varphi_2)) \\
\text{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2) & = & 2 + a + max(\text{future}(\varphi_1), \text{future}(\varphi_2))
\end{array}
$$

- Why 2 additional **lookaheads** for $\text{future}(\varphi_1\, \mathcal{U}_{(a,\infty)} \varphi_2)$?



Timed Automata

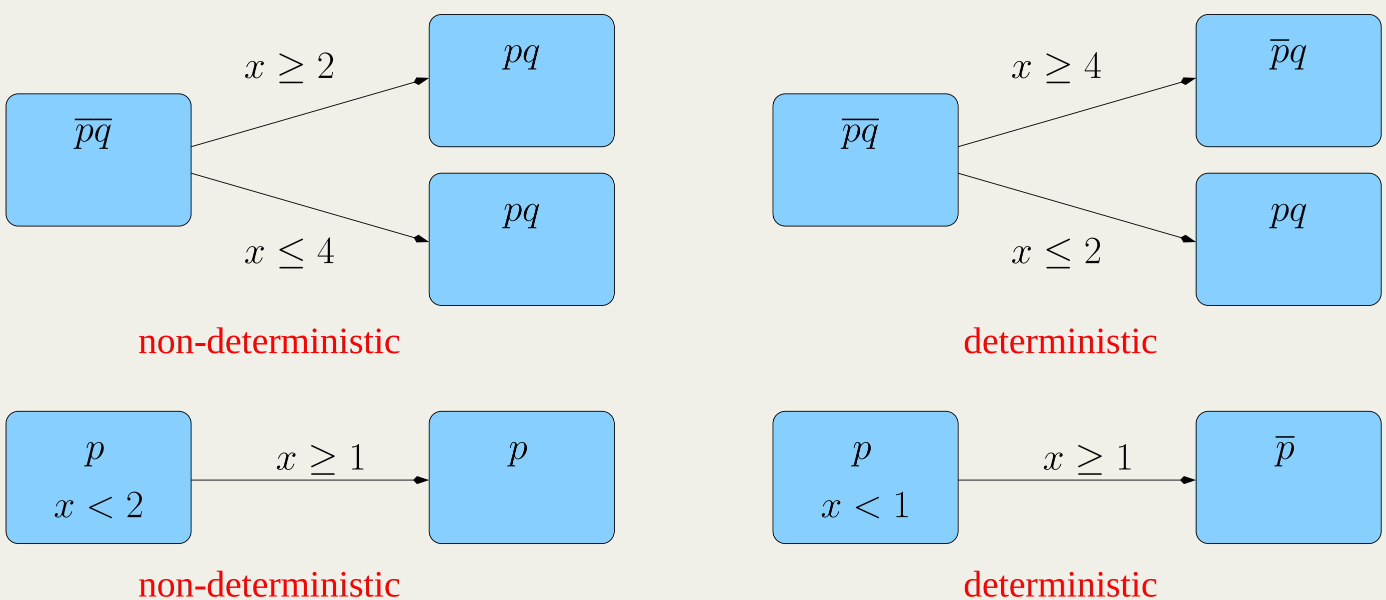
- Variant of **timed automata**
  - Reads multi-dimensional Boolean signals
  - Clock assignments of the form $x := 0$, $x := y$ and $x := \bot$
  - Generalized Büchi and parity accepance conditions



- Run ξ: alternation of **discrete** and **time** steps

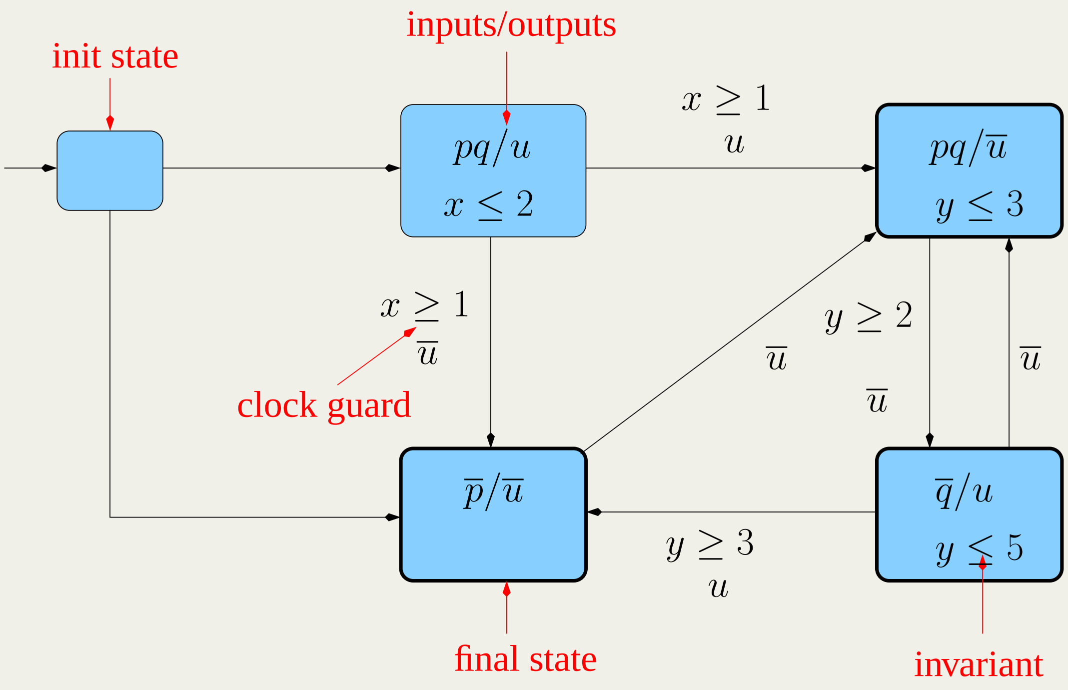
Deterministic Timed Automata

- A timed automaton is **deterministic** if the following conditions hold:
  1. For any 2 transitions with the same source state, either the **labels** of the 2 target states are **different** or the **intersection** of the 2 transition **guards** is **unsatisfiable**
  2. For any transition, either the **labels** of the source and target states are **different**, or the **intersection** between the **source state invariant** and the **transition guard** is either **empty** or **isolated**



Dependent Timed Automata

- DTA $\rightarrow$ **transducers** of runs of TA
  - Both input and output alphabets
  - Input/output labels on states
  - Output labels on transitions
  - Passive read of clock of TA (no assignments)



Composition of TA and DTA

1. Composition of two TAs

$$L(A_1 \mid\mid A_2) = L(A_1) \times L(A_2)$$

2. Composition of two DTAs

For every run ξ and signal w, $B_1 \otimes B_2(w, \xi) = B_2(B_1(w, \xi))$

3. Composition of a TA and a DTA

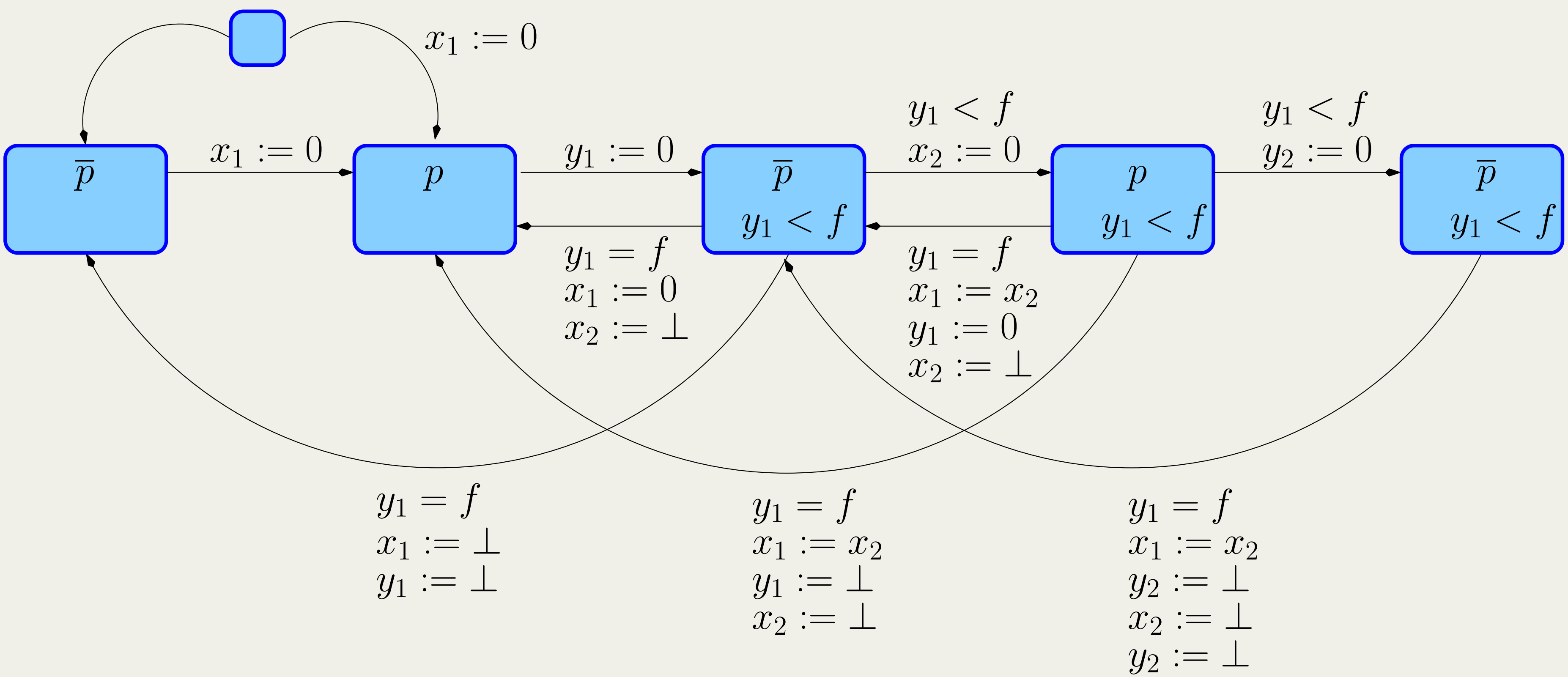
$$L(A_1 \otimes B_2) = \{w \mid \exists \xi_1 \text{ accepting run of } A_1 \text{ carrying } w \text{ and } B_2(w, \xi_1) \neq \emptyset\}$$

From MTL to Non-Deterministic Timed Automata - Overview

- Novel construction for conversion of MTL formulas into **non-deterministic** timed automata
  - Distinguishes between discrete guesses about the future and accumulation of knowledge with clocks
  - **Proposition monitors:** deterministic TA that memorize information about the input
  - Non-deterministic sequence of DTAs that handle arbitrary MTL formulas

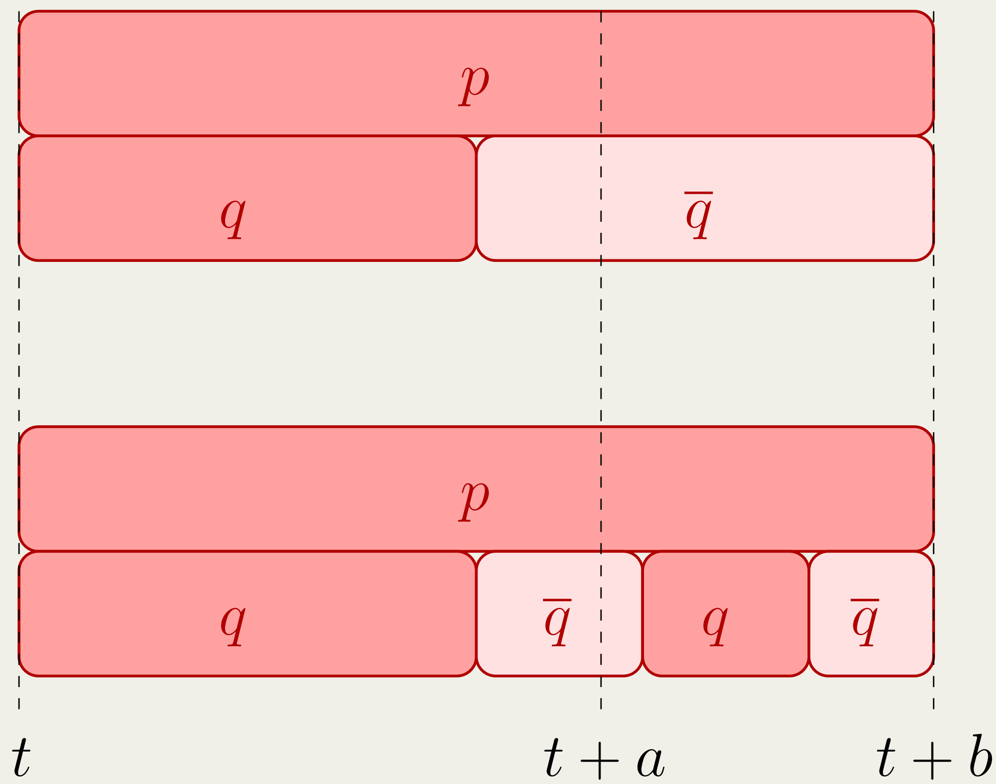
Proposition Monitor

- Proposition monitor for p, where $f = \text{future}(\varphi)$
- Requires $2 \cdot \lceil \frac{fk}{2} \rceil$ clocks, where k is the bounded variability of p



Dependent Timed Automaton for $\varphi_1 \mathcal{U}_{(a,b)} \varphi_2$

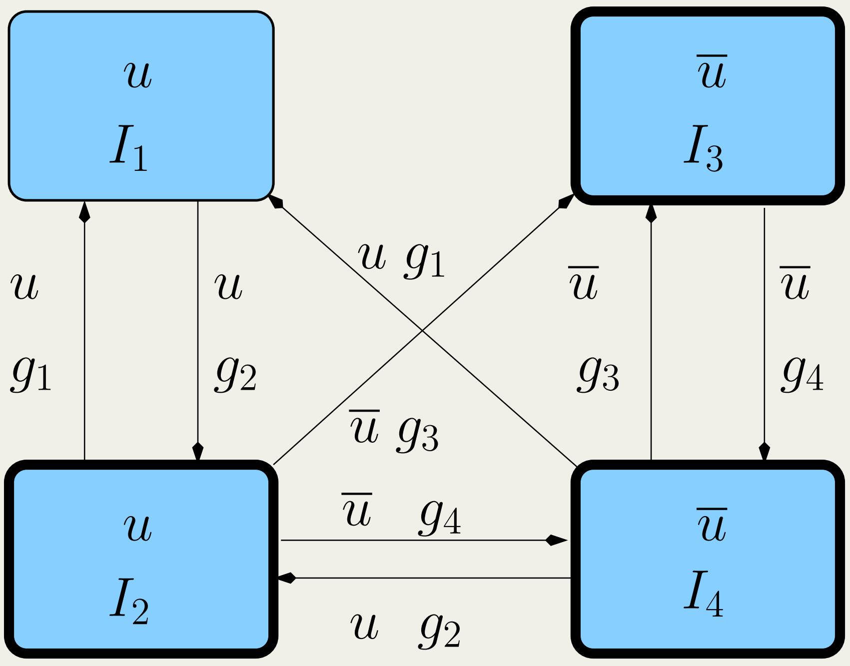
$p\mathcal{U}_{(a,b)}q$



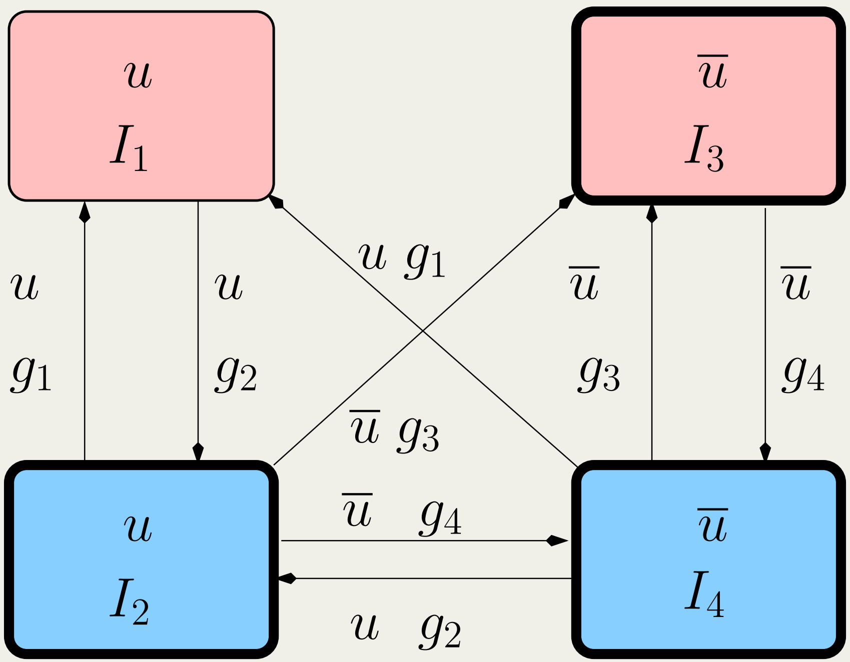
$(w, t) \not\models p\mathcal{U}_{(a,b)}q$

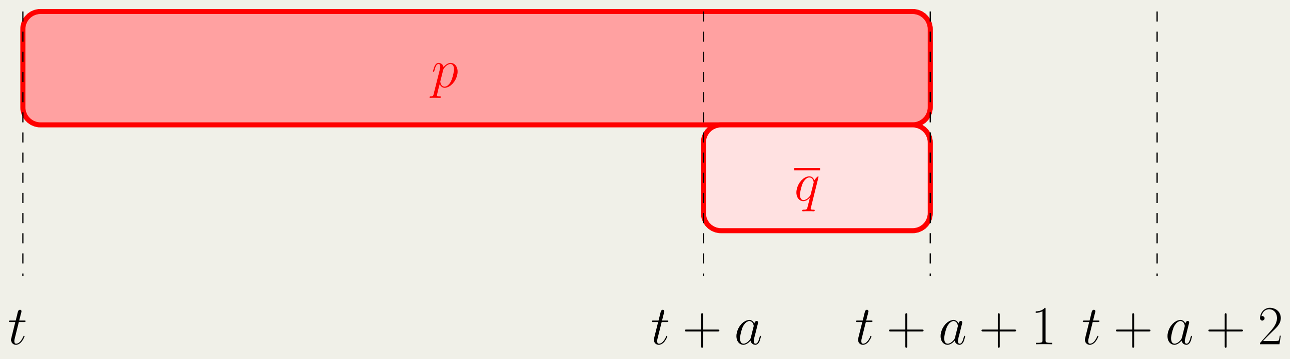
$(w, t) \models p\mathcal{U}_{(a,b)}q$

Dependent Timed Automaton for $\varphi_1 \mathcal{U}_{(a,\infty)} \varphi_2$

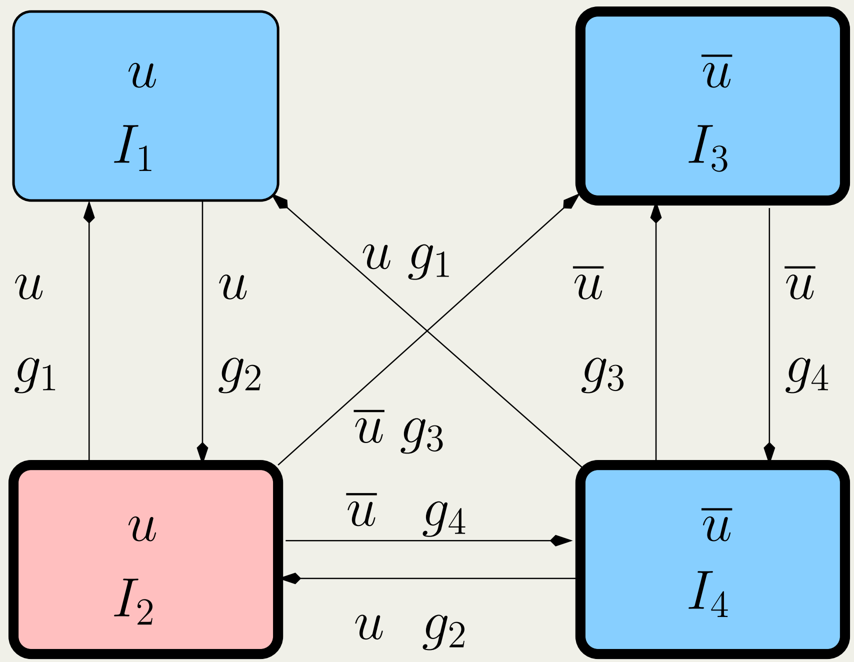


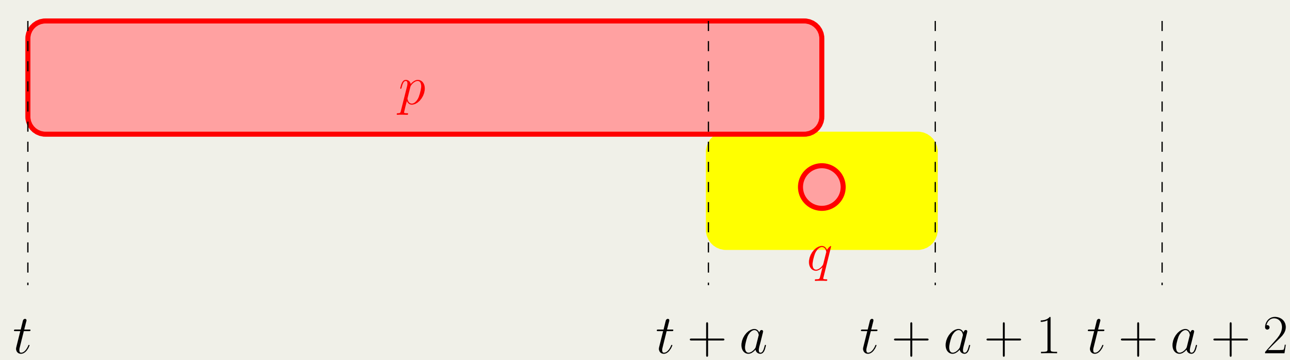
Dependent Timed Automaton for $\varphi_1 \mathcal{U}_{(a,\infty)} \varphi_2$



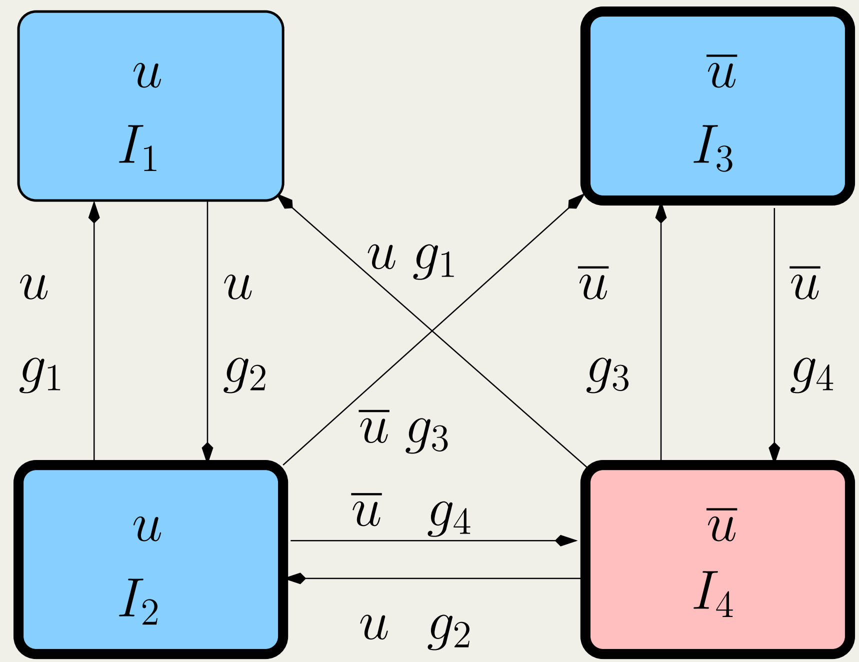


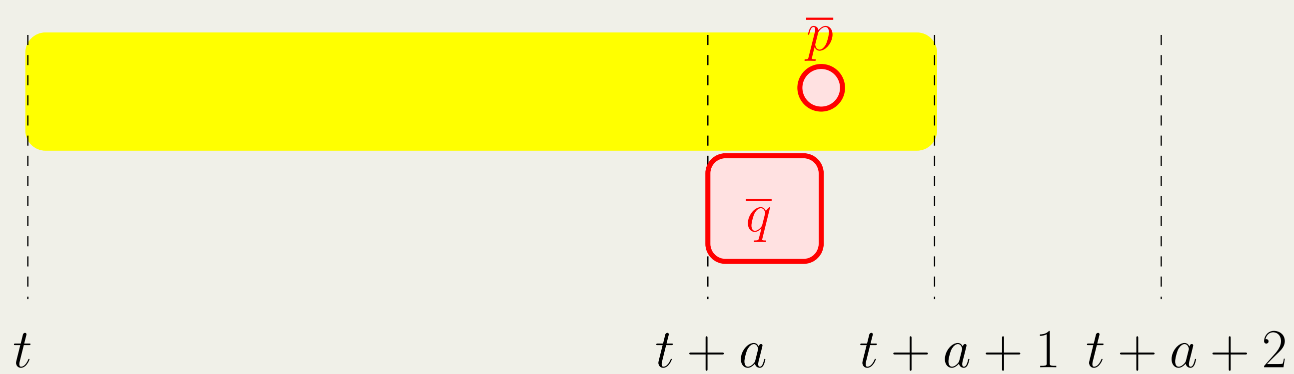
Dependent Timed Automaton for $\varphi_1 \mathcal{U}_{(a,\infty)} \varphi_2$



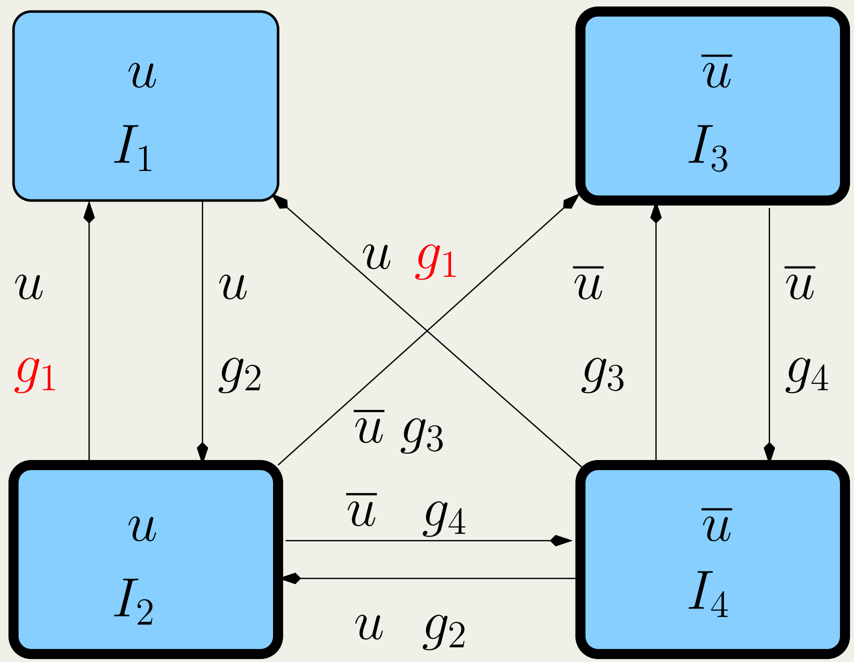


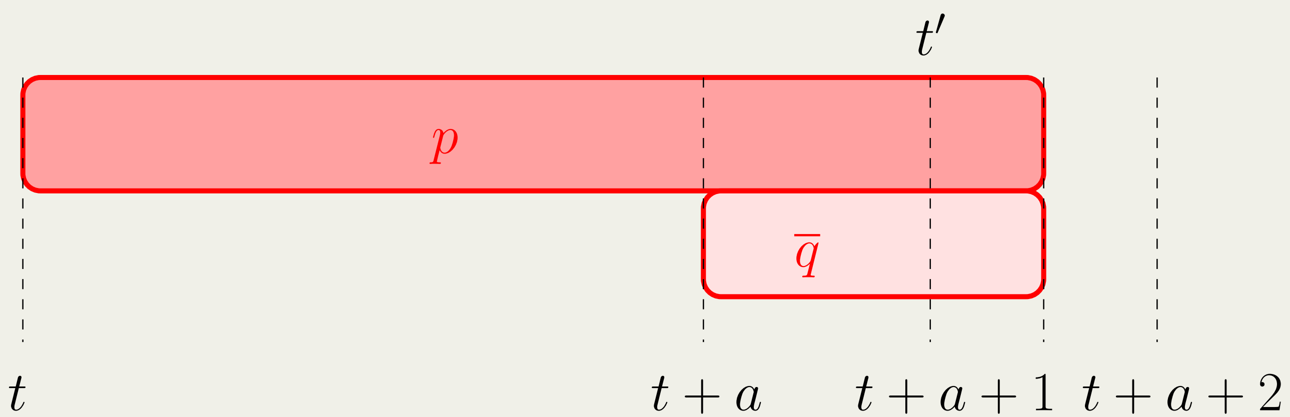
Dependent Timed Automaton for $\varphi_1 \mathcal{U}_{(a,\infty)} \varphi_2$





Dependent Timed Automaton for $\varphi_1 \mathcal{U}_{(a,\infty)} \varphi_2$





Summary: MTL to Non-deterministic TA

- Inductive construction of a timed automaton A_φ that accepts the language of arbitrary MTL formula φ
- For every MTL formula φ with m propositions, n unbounded temporal operators, and inputs of bounded variability k, there exists a non-deterministic TA with $2m\lceil\frac{k\cdot\text{future}(\varphi)}{2}\rceil + 1$ clocks and $((2\lceil\frac{k\cdot\text{future}(\varphi)}{2}\rceil)^m + 1)(2 \cdot 4^n + 1)$ states

Determinizing Timed Automata Obtained from MTL Formulas

- Construction for the conversion of MTL formulas to non-deterministic timed automata
  - $\rightarrow$ can be **determinized!!**
- Subset construction for finite and infinite words
- Piterman's variation of Safra's construction
  - Slight adaptations - mostly syntactic
  - Take into account 'asynchronicity' of transitions from a set of states
- Non-deterministic DTA $B \rightarrow$ deterministic DTA D
- For every deterministic TA A, $L(A \otimes B) = L(A \otimes D)$
- For every MTL formula φ with m propositions, n unbounded temporal operators, and inputs of bounded variability k, there exists a deterministic TA with $2m\lceil\frac{k \cdot \text{future}(\varphi)}{2}\rceil + 1$ clocks and $((2\lceil\frac{k \cdot \text{future}(\varphi)}{2}\rceil)^m + 1) \cdot 2^{2^{nlogn}})$ states

Determinizing Timed Automata Obtained from MTL Formulas

- Construction for the conversion of MTL formulas to non-deterministic timed automata
  - $\rightarrow$ can be **determinized!!**
- Subset construction for finite and infinite words
- Piterman's variation of Safra's construction
  - Slight adaptations - mostly syntactic
  - Take into account 'asynchronicity' of transitions from a set of states
- Non-deterministic DTA $B \rightarrow$ deterministic DTA D
- For every deterministic TA A, $L(A \otimes B) = L(A \otimes D)$
- For every MTL formula φ with m propositions, n unbounded temporal operators, and inputs of bounded variability k, there exists a deterministic TA with $2m\lceil \frac{k \cdot \text{future}(\varphi)}{2} \rceil + 1$ clocks and $((2\lceil \frac{k \cdot \text{future}(\varphi)}{2} \rceil)^m + 1) \cdot 2^{2^{nlogn}})$ states

Conclusions and Future Work

Conclusions:

- Novel construction for translating MTL to timed automata under bounded variability assumption
- Unified framework for model checking, monitoring and controller synthesis
- Exponentially improves on the complexity of securing deterministic timed automata
  - Avoids doubly exponential number of clocks

- Consider MTL with past operators
- Optimize and improve the translation
- Implementation

Conclusions and Future Work

Conclusions:

- Novel construction for translating MTL to timed automata under bounded variability assumption
- Unified framework for model checking, monitoring and controller synthesis
- Exponentially improves on the complexity of securing deterministic timed automata
  - Avoids doubly exponential number of clocks

Future Work:

- Consider MTL with past operators
- Optimize and improve the translation
- Implementation