Model Checking Metric Temporal Logic over Automata with One Counter

Karin Quaas

11th April 2013

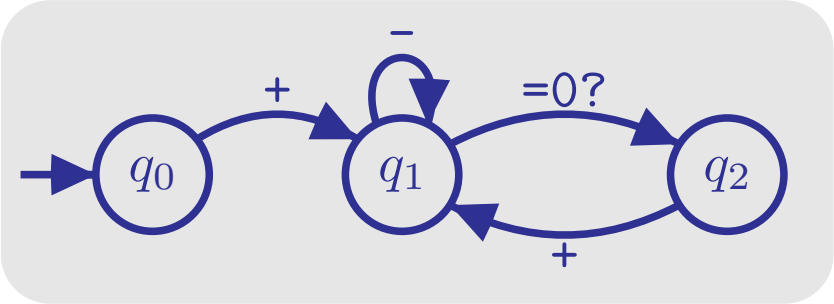
Overview of the Talk

- Automata with one counter:
  - 1-Counter Machines
  - 1-Dimensional Vector Addition State Systems
  - Weighted Automata
- Metric Temporal Logic
- Undecidability Result
- Decidability Result

1-Counter Machines (1CM)

$M = (Q, q_0, \Delta)$, where

- Q is a finite set of control states,
- $q_0 \in Q$ is the initial control state,
- $\Delta \subseteq Q \times \mathrm{Op} \times Q$, where $\mathrm{Op} = \{+, -, =0?\}$



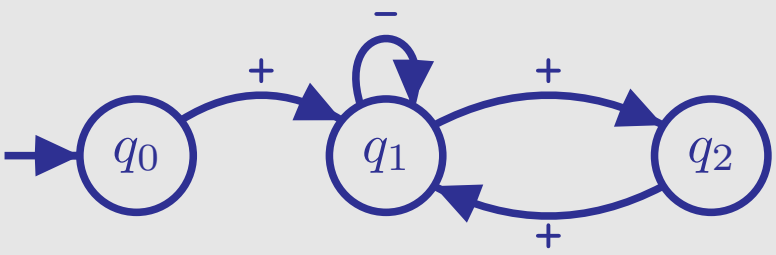
$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{-} (q_1, 0) \xrightarrow{=0?} (q_2, 0) \xrightarrow{+} (q_1, 1) \ldots$$

- Zero Test-edges are blocked if the value of the counter is not zero
- Decrement-edges are blocked if the value of the counter is zero

1-Dimensional (Vector) Addition State Systems (1-VASS)

$M = (Q, q_0, \Delta)$, where

- Q is a finite set of control states,
- $q_0 \in Q$ is the initial control state,
- $\Delta \subseteq Q \times \mathrm{Op} \times Q$, where $\mathrm{Op} = \{+, -\}$ No Zero Test!



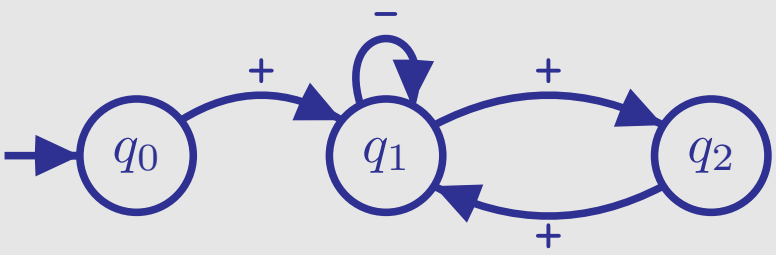
$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{-} (q_1, 0) \xrightarrow{+} (q_2, 1) \xrightarrow{+} (q_1, 2) \ldots$$

- Decrement-edges are blocked if the value of the counter is zero

Weighted Automaton (WA)

$M = (Q, q_0, \Delta)$, where

- Q is a finite set of control states,
- $q_0 \in Q$ is the initial control state,
- $\Delta \subseteq Q \times \mathsf{Op} \times Q$, where $\mathsf{Op} = \{+, -\}$ No Zero Test!



$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{-} (q_1, 0) \xrightarrow{-} (q_1, -1) \xrightarrow{+} (q_2, 0) \ldots$$

- No edges are blocked. The counter may have a negative value.

Metric Temporal Logic (MTL) - Syntax

The set of MTL formulae over a finite set Q is defined by induction:

- q is a formula,
- if φ and ψ are formulae, then so are $\neg\varphi$ and $\varphi \wedge \psi$,
- if φ and ψ are formulae, then so are $\bigcirc_I \varphi$ and $\varphi \mathsf{U}_I \psi$,

where $q \in Q$ and $I \subseteq \mathbb{Z}$ is an interval with endpoints in $\mathbb{Z} \cup \{-\infty, \infty\}$.
If $I = \mathbb{Z}$, then we may omit I.

Abbreviations:

$\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$
$\varphi \rightarrow \psi := \neg\varphi \vee \psi$
$\mathtt{true} := \varphi \vee \neg\varphi$
$\Diamond_I \varphi := \mathtt{true} \mathsf{U}_I \varphi$
$\Box_I \varphi := \neg\Diamond_I \neg\varphi$

Metric Temporal Logic (MTL) - Semantics

Let $\gamma = (q_0, c_0) \to (q_1, c_1) \to (q_2, c_2) \to \dots$ be a computation of a 1-CM (1-VASS, WA), and let $i \in \mathbb{N}$.
The satisfaction relation for MTL is defined by induction:

$$
\begin{array}{rcl}
(\gamma, i) \models q & \text{iff} & q = q_i \\
(\gamma, i) \models \neg\varphi & \text{iff} & (\gamma, i) \models \varphi \text{ is not the case} \\
(\gamma, i) \models \varphi \wedge \psi & \text{iff} & (\gamma, i) \models \varphi \text{ and } (\gamma, i) \models \psi \\
(\gamma, i) \models \bigcirc_I \varphi & \text{iff} & (\gamma, i+1) \models \varphi \text{ and } c_{i+1} - c_i \in I \\
(\gamma, i) \models \varphi \mathsf{U}_I \psi & \text{iff} & \exists j \geq i. (\gamma, j) \models \psi, c_j - c_i \in I \text{ and} \\
 & & \forall i \leq k < j. (\gamma, k) \models \varphi \\
(\gamma, i) \models \Diamond_I \varphi & \text{iff} & \exists j \geq i. (\gamma, j) \models \varphi \text{ and } c_j - c_i \in I \\
(\gamma, i) \models \Box_I \varphi & \text{iff} & \forall j \geq i. \text{ If } c_j - c_i \in I \text{ then } (\gamma, j) \models \varphi
\end{array}
$$

We write $\gamma \models \varphi$ if $(\gamma, 0) \models \varphi$.

The Model Checking Problem

INPUT: A 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

The Model Checking Problem

INPUT: A 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

State of the Art:

- MTL-Model Checking WA with non-negative weights is EXPSPACE-complete (Laroussinie et al. 2002).

The Model Checking Problem

INPUT: A 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

State of the Art:

- MTL-Model Checking WA with non-negative weights is EXPSPACE-complete (Laroussinie et al. 2002).
- Freeze LTL-Model Checking 2-VASS is undecidable. (Demri et al. 2010)
- Freeze LTL-Model Checking 1-CM is undecidable (Demri et al. 2008)

- Freeze LTL: are the counter values equal at two different arbitrary positions?
- MTL formulae can be translated into equivalent formulae of Freeze LTL interval extension
- It is conjectured that Freeze LTL interval extension is expressively stronger than MTL

The Model Checking Problem

INPUT: A 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

State of the Art:

- MTL-Model Checking WA with non-negative weights is EXPSPACE-complete (Laroussinie et al. 2002).
- Freeze LTL-Model Checking 2-VASS is undecidable. (Demri et al. 2010)
- Freeze LTL-Model Checking 1-CM is undecidable (Demri et al. 2008)

- Freeze LTL: are the counter values equal at two different arbitrary positions?
- MTL formulae can be translated into equivalent formulae of Freeze LTL interval extension
- It is conjectured that Freeze LTL interval extension is expressively stronger than MTL

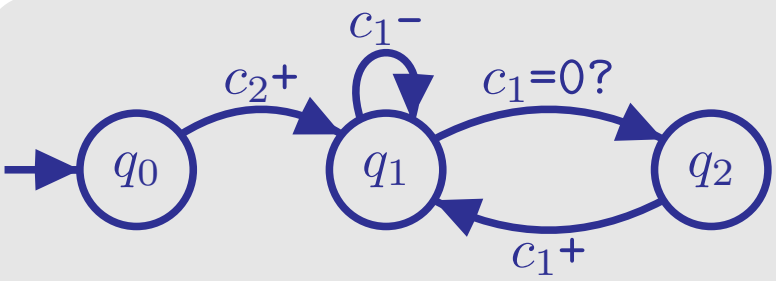
Theorem
MTL-model checking WA (1-VASS, 1-CM) is undecidable.

Theorem
MTL-model checking WA (1-VASS, 1-CM) is undecidable.

Proof
Reduction of the reachability problem for 2-Counter Machines

A 2-Counter Machine (2-CM) is a tuple $M = (Q, q_0, \Delta)$, where

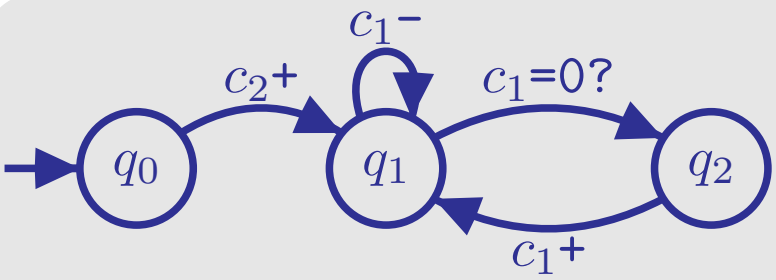
- Q is a finite set of control states,
- $q_0 \in Q$ is the initial control state,
- $\Delta \subseteq Q \times \mathrm{Op} \times Q$, where $\mathrm{Op} = \{c_1\texttt{+}, c_1\texttt{-}, c_1\texttt{=0?}, c_2\texttt{+}, c_2\texttt{-}, c_2\texttt{=0?}\}$



$$(q_0, 0, 0) \xrightarrow{c_2\texttt{+}} (q_1, 0, 1) \xrightarrow{c_1\texttt{=0?}} (q_2, 0, 1) \xrightarrow{c_1\texttt{+}} (q_1, 1, 1) \xrightarrow{c_1\texttt{-}} (q_1, 0, 1) \ldots$$

A 2-Counter Machine (2-CM) is a tuple $M = (Q, q_0, \Delta)$, where

- Q is a finite set of control states,
- $q_0 \in Q$ is the initial control state,
- $\Delta \subseteq Q \times \mathsf{Op} \times Q$, where $\mathsf{Op} = \{c_1\mathsf{+}, c_1\mathsf{-}, c_1\mathsf{=0?}, c_2\mathsf{+}, c_2\mathsf{-}, c_2\mathsf{=0?}\}$



$$(q_0, 0, 0) \xrightarrow{c_2\mathsf{+}} (q_1, 0, 1) \xrightarrow{c_1\mathsf{=0?}} (q_2, 0, 1) \xrightarrow{c_1\mathsf{+}} (q_1, 1, 1) \xrightarrow{c_1\mathsf{-}} (q_1, 0, 1) \ldots$$

The Reachability Problem

INPUT: A 2-CM $M = (Q, q_0, \Delta)$, $q \in Q$.
QUESTION: Is there a computation of M ending in q?

This problem is undecidable.

Theorem
MTL-model checking WA (1-VASS, 1-CM) is undecidable.

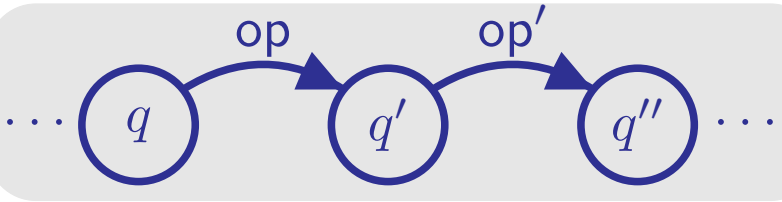
Proof
Reduction of the reachability problem for 2-Counter Machines

We present a procedure how to translate every 2-CM M'' and q into a WA M' and an MTL-Formula φ such that

there is a computation of M'' ending in q
iff
there is a computation γ of M' such that $\gamma \models \varphi$.

Proof

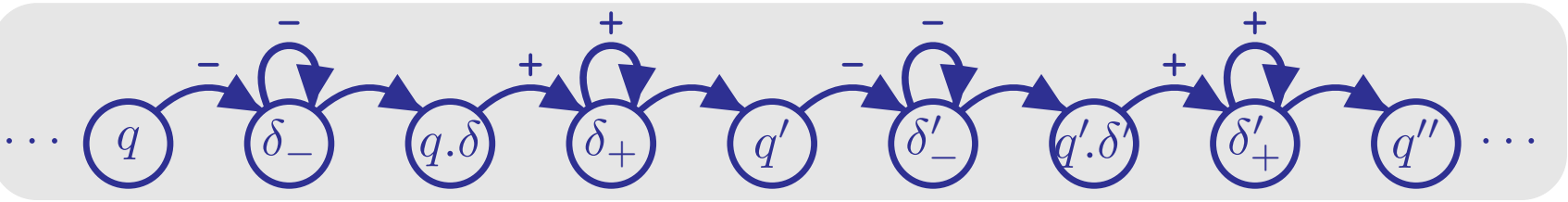
Encoding two counters...



$\delta = (q, \mathsf{op}, q')$
$\delta' = (q', \mathsf{op}', q'')$

$\ldots (q, c, d) \xrightarrow{\mathsf{op}} (q', c', d') \xrightarrow{\mathsf{op}'} (q'', c'', d'') \ldots$

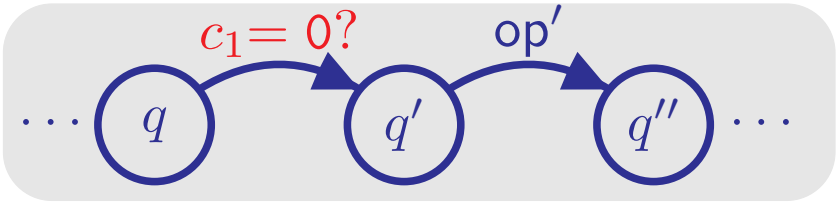
...into one counter:



$\ldots (q, c+d) \ \ldots \ (q.\delta, c) \ \ldots \ (q', c'+d') \ \ldots \ (q'.\delta', c') \ \ldots \ (q'', c''+d'') \ldots$

Proof

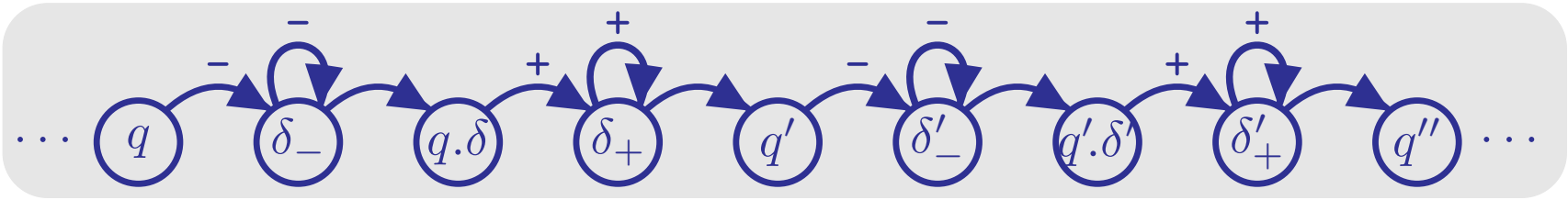
Encoding two counters...



$\delta = (q, c_1\text{=}0?, q')$
$\delta' = (q', \mathsf{op}', q'')$

$\ldots (q,c,d) \xrightarrow{c_1=0?} (q',c',d') \xrightarrow{\mathsf{op}'} (q'',c'',d'') \ldots$ $\Rightarrow c = 0, c' = c, d' = d$

...into one counter:

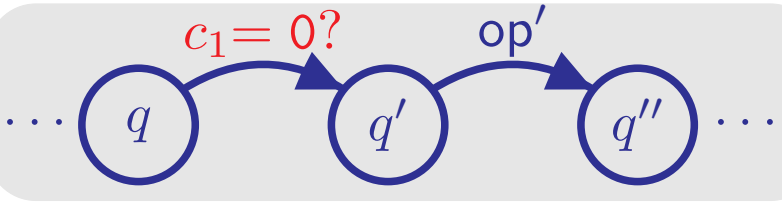


$\ldots (q, c+d) \ \ldots \ (q.\delta, c) \ \ldots \ (q', c'+d') \ \ldots \ (q'.\delta', c') \ \ldots \ (q'', c''+d'') \ldots$

Ensure the correct semantics, e.g. zero tests:

Proof

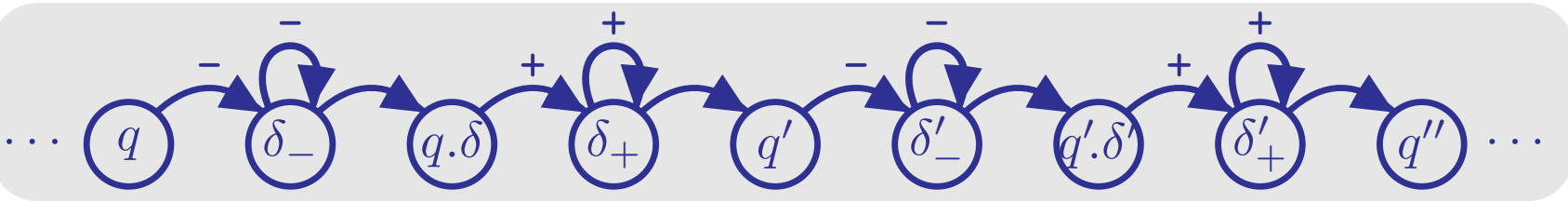
Encoding two counters...



$\delta = (q, c_1\text{=}0?, q')$
$\delta' = (q', \mathsf{op}', q'')$

$\ldots (q,c,d) \xrightarrow{c_1=0?} (q',c',d') \xrightarrow{\mathsf{op}'} (q'',c'',d'') \ldots$ $\Rightarrow c = 0, c' = c, d' = d$

...into one counter:



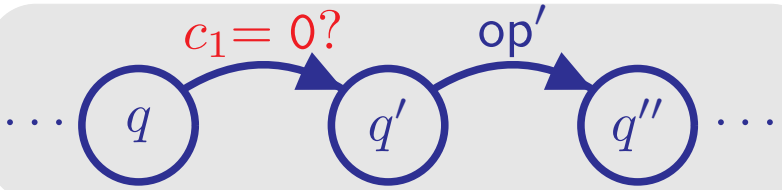
$\ldots (q, c+d) \ \ldots \ (q.\delta, c) \ \ldots \ (q', c'+d') \ \ldots \ (q'.\delta', c') \ \ldots \ (q'', c''+d'') \ldots$

Ensure the correct semantics, e.g. zero tests:

$\varphi_{\mathsf{zero1}} = (\Box_{[1,\infty)} \neg q.\delta) \wedge (\Box_{(-\infty,-1]} \neg q.\delta)$

Proof

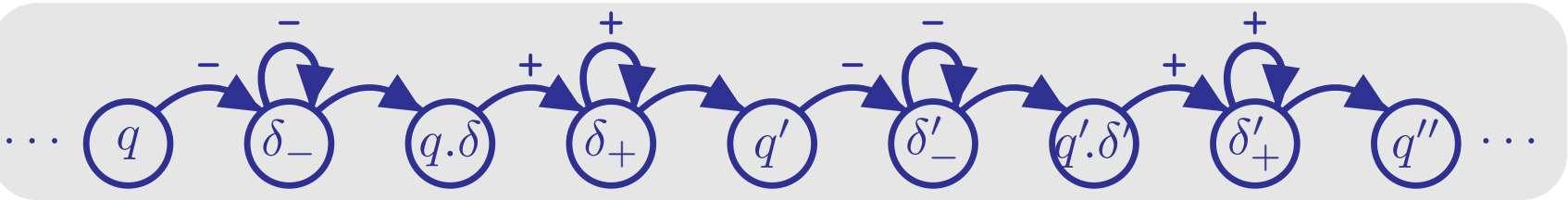
Encoding two counters...



$\delta = (q, c_1\text{=}0?, q')$
$\delta' = (q', \mathsf{op}', q'')$

$\ldots (q,c,d) \xrightarrow{c_1=0?} (q',c',d') \xrightarrow{\mathsf{op}'} (q'',c'',d'') \ldots$ $\quad \Rightarrow c = 0, c' = c, d' = d$

...into one counter:



$\ldots (q, c+d) \ \ldots \ (q.\delta, c) \ \ldots \ (q', c'+d') \ \ldots \ (q'.\delta', c') \ \ldots \ (q'', c''+d'') \ldots$

Ensure the correct semantics, e.g. zero tests:

$\varphi_{\mathsf{zero1}} = (\Box_{[1,\infty)} \neg q.\delta) \wedge (\Box_{(-\infty,-1]} \neg q.\delta)$

$\varphi_{\mathsf{nochange}} = \Box[(q \wedge \bigcirc \delta_-) \rightarrow ((q \vee \delta_- \vee q.\delta \vee \delta_+) \mathsf{U}_{[0,0]} q')]$

Theorem
MTL-model checking WA (1-VASS, 1-CM) is undecidable.

Proof
Reduction of the undecidable reachability problem for 2-Counter Machines

We presented a procedure how to translate every 2-CM M'' and q
into a WA M' and an MTL-Formula φ such that

there is a computation of M'' ending in q
iff
there is a computation γ of M' such that $\gamma \models \varphi$.

Theorem
MTL-model checking WA (1-VASS, 1-CM) is undecidable.

Proof
Reduction of the undecidable reachability problem for 2-Counter Machines

We presented a procedure how to translate every 2-CM M'' and q
into a WA M' and an MTL-Formula φ such that

there is a computation of M'' ending in q
iff
there is a computation γ of M' such that $\gamma \models \varphi$.

Remark
Reduction also works if the formulae may only contain intervals of the form

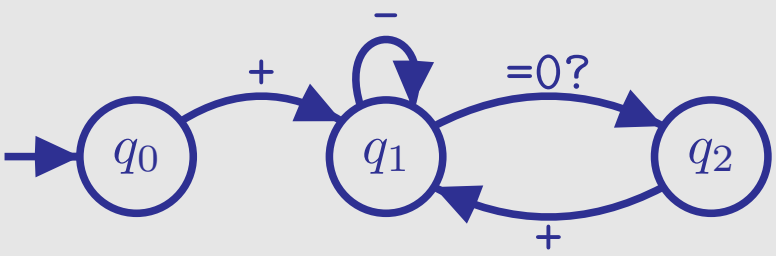
$$\mathtt{Z}, (-\infty, -1], [0, \infty)$$

The Model Checking Problem

INPUT: A deterministic 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

Deterministic 1-Counter Machines

For each configuration (q, c) there is at most one successor configuration.



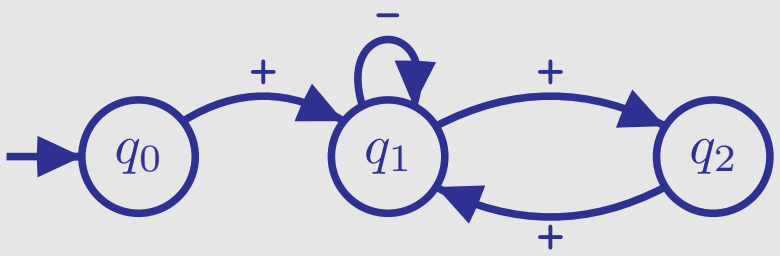
$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{-} (q_1, 0) \xrightarrow{=0?} (q_2, 0) \xrightarrow{+} (q_1, 1) \ldots$$

This 1-CM is deterministic:

$$\mathsf{succ}(q_1, c) = \begin{cases} (q_1, c-1) & \text{if } c \neq 0 \\ (q_2, 0) & \text{if } c = 0 \end{cases}$$

Deterministic Weighted Automata

For each configuration (q, c) there is at most one successor configuration.



$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{-} (q_1, 0) \xrightarrow{+} (q_2, 1) \xrightarrow{+} (q_1, 2) \ldots$$

This WA is not deterministic:

$$\mathsf{succ}(q_1, c) = \{(q_1, c-1), (q_2, c+1)\}$$

The Model Checking Problem

INPUT: A deterministic 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

State of the Art:

- Freeze LTL-Model Checking of deterministic 1-CM is PSPACE-complete. (Demri et al. 2008)

- Freeze LTL: are the counter values equal at two different positions?
- MTL formulae can be translated into equivalent formulae of Freeze LTL interval extension
- It is conjectured that Freeze LTL interval extension is expressively stronger than MTL

The Model Checking Problem

INPUT: A deterministic 1-CM (1-VASS, WA) M, an MTL formula φ.
QUESTION: Is there some computation γ of M such that $\gamma \models \varphi$?

State of the Art:

- Freeze LTL-Model Checking of deterministic 1-CM is TODO. (Demri et al. 2008)

- Freeze LTL: are the counter values equal at two different positions?
- MTL formulae can be translated into equivalent formulae of Freeze LTL interval extension
- It is conjectured that Freeze LTL interval extension is expressively stronger than MTL

Theorem
Freeze LTL interval extension-model checking of deterministic 1-CM (1-VASS, WA) is decidable.

Corollary
MTL-model checking of deterministic 1-CM (1-VASS, WA) is decidable.

Theorem
Freeze LTL interval extension-model checking of deterministic 1-CM (1-VASS, WA) is decidable.

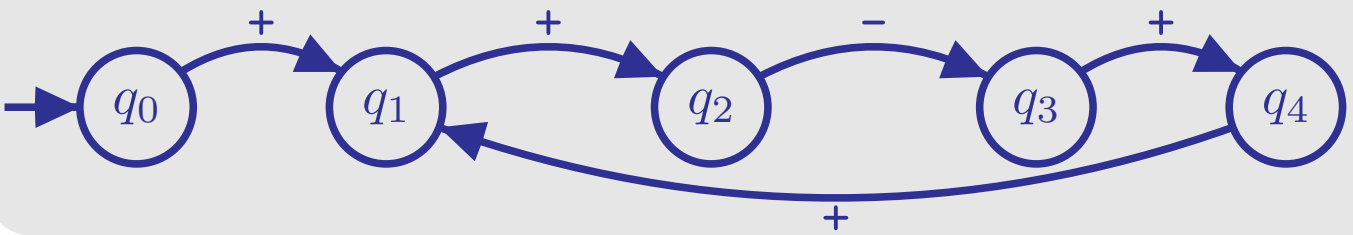
Proof
Reduction to the Büchi-acceptance problem for Büchi automata.

We present a procedure how to translate every deterministic 1-CM M and MTL formula φ into a Büchi automaton A such that

there is a computation γ of M with $\gamma \models \varphi$
iff
there is a Büchi accepting run of A

Proof

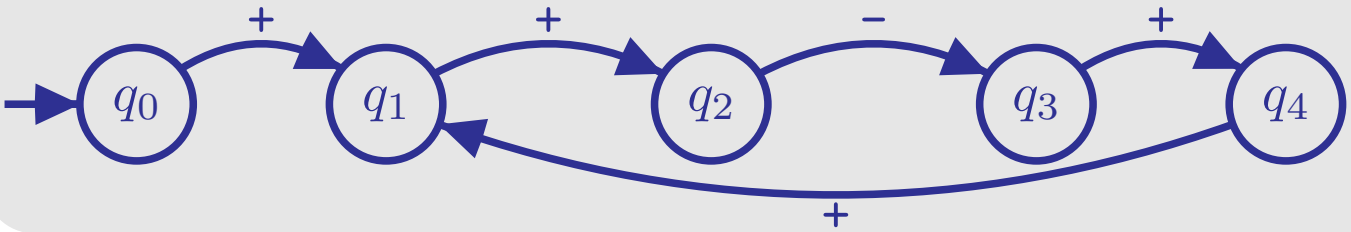
The unique computation of M has a regular structure:



$$
\begin{aligned}
(q_0,0) \xrightarrow{+} (q_1,1) \xrightarrow{+} (q_2,2) \xrightarrow{-} (q_3,1) \xrightarrow{+} (q_4,2) \xrightarrow{+} \\
(q_1,3) \xrightarrow{+} (q_2,4) \xrightarrow{-} (q_3,3) \xrightarrow{+} (q_4,4) \xrightarrow{+} \\
(q_1,5) \xrightarrow{+} (q_2,6) \xrightarrow{-} (q_3,5) \xrightarrow{+} (q_4,6) \xrightarrow{+} \\
\ldots
\end{aligned}
$$

Proof

The unique computation of M has a regular structure:



$$
\begin{aligned}
(q_0,0) \xrightarrow{+} (q_1,1) \xrightarrow{+} (q_2,2) \xrightarrow{-} (q_3,1) \xrightarrow{+} (q_4,2) \xrightarrow{+} \\
(q_1,3) \xrightarrow{+} (q_2,4) \xrightarrow{-} (q_3,3) \xrightarrow{+} (q_4,4) \xrightarrow{+} \\
(q_1,5) \xrightarrow{+} (q_2,6) \xrightarrow{-} (q_3,5) \xrightarrow{+} (q_4,6) \xrightarrow{+} \\
\ldots
\end{aligned}
$$

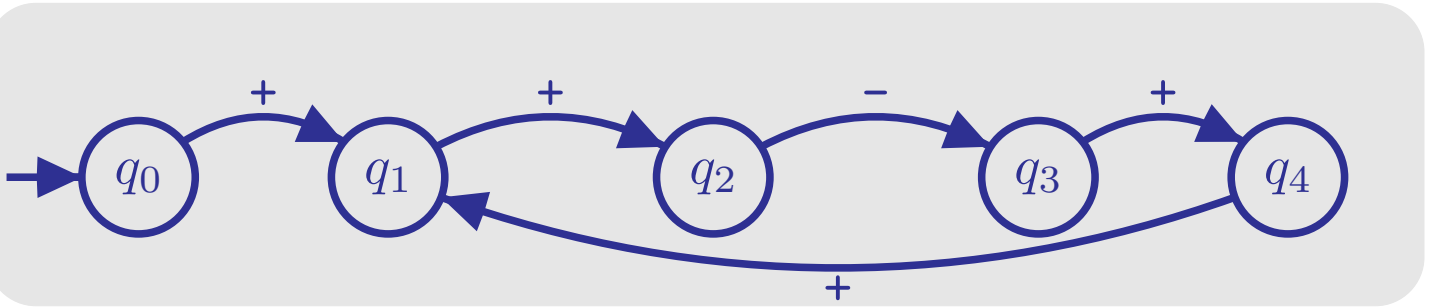
$\}+2$
$\}+2$
$\}+2$

prefix — repeated infinitely often

Infinitely many counter values occur, but with regularity.

Proof

The unique computation of M has a regular structure:

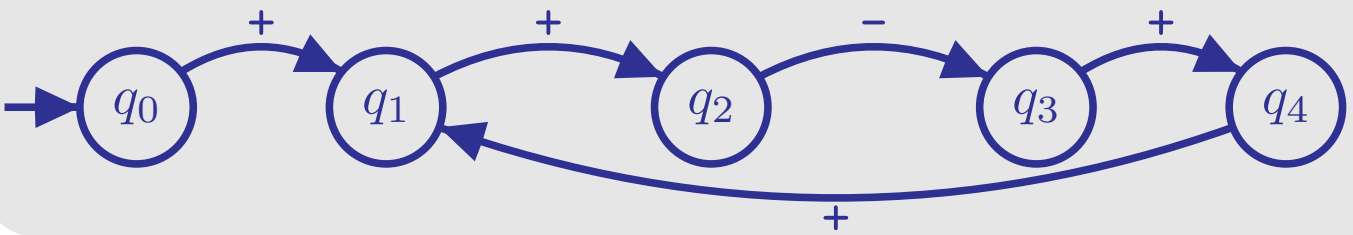


$$
\begin{aligned}
(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{+} (q_2, 2) \xrightarrow{-} (q_3, 1) \xrightarrow{+} (q_4, 2) \xrightarrow{+} \\
(q_1, 3) \xrightarrow{+} (q_2, 4) \xrightarrow{-} (q_3, 3) \xrightarrow{+} (q_4, 4) \xrightarrow{+} \\
(q_1, 5) \xrightarrow{+} (q_2, 6) \xrightarrow{-} (q_3, 5) \xrightarrow{+} (q_4, 6) \xrightarrow{+} \\
\ldots
\end{aligned}
$$

$$\mathsf{offset}(i, I) = \{j \mid c_{i+j} - c_i \in I\}$$

Proof

The unique computation of M has a regular structure:



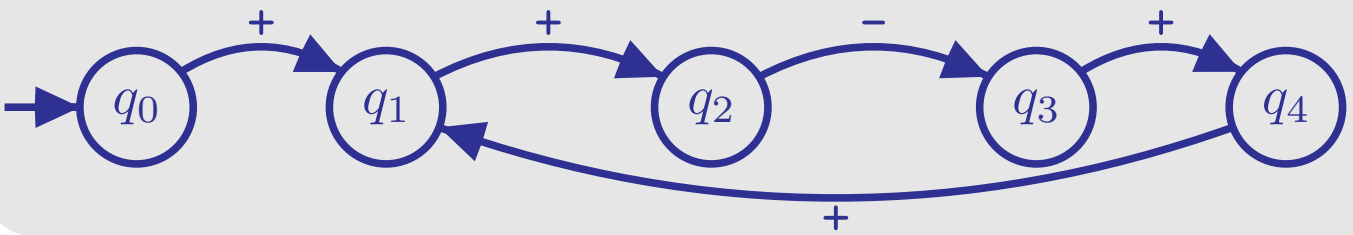
$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{+} (q_2, 2) \xrightarrow{-} (q_3, 1) \xrightarrow{+} (q_4, 2) \xrightarrow{+}$$
$$(q_1, 3) \xrightarrow{+} (q_2, 4) \xrightarrow{-} (q_3, 3) \xrightarrow{+} (q_4, 4) \xrightarrow{+}$$
$$(q_1, 5) \xrightarrow{+} (q_2, 6) \xrightarrow{-} (q_3, 5) \xrightarrow{+} (q_4, 6) \xrightarrow{+}$$
$$\ldots$$

$$\text{offset}(i, I) = \{j \mid c_{i+j} - c_i \in I\}$$

e.g., $\text{offset}(2, [2, 3]) = \{j \mid c_{2+j} - c_2 \in [2, 3]\}$

Proof

The unique computation of M has a regular structure:



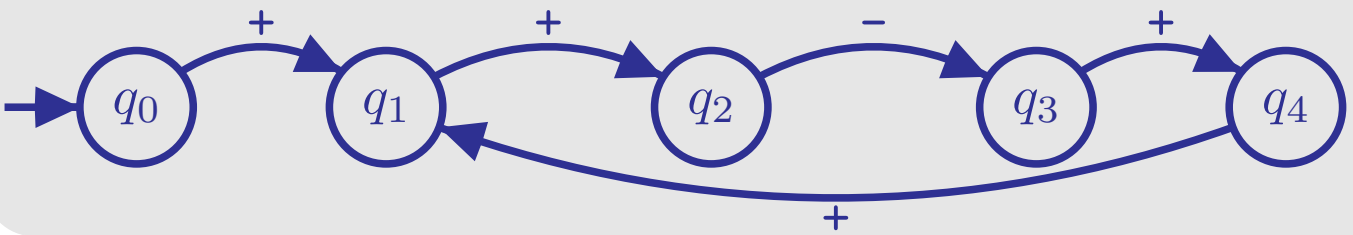
$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{+} (q_2, 2) \xrightarrow{-} (q_3, 1) \xrightarrow{+} (q_4, 2) \xrightarrow{+}$$
$$(q_1, 3) \xrightarrow{+} (q_2, 4) \xrightarrow{-} (q_3, 3) \xrightarrow{+} (q_4, 4) \xrightarrow{+}$$
$$(q_1, 5) \xrightarrow{+} (q_2, 6) \xrightarrow{-} (q_3, 5) \xrightarrow{+} (q_4, 6) \xrightarrow{+}$$
$$\ldots$$

$$\text{offset}(i, I) = \{j \mid c_{i+j} - c_i \in I\}$$

$$\text{e.g., } \text{offset}(2, [2, 3]) = \{j \mid c_{2+j} - c_2 \in [2, 3]\}$$
$$= \{j \mid c_{2+j} = 4 \text{ or } c_{2+j} = 5\}$$

Proof

The unique computation of M has a regular structure:



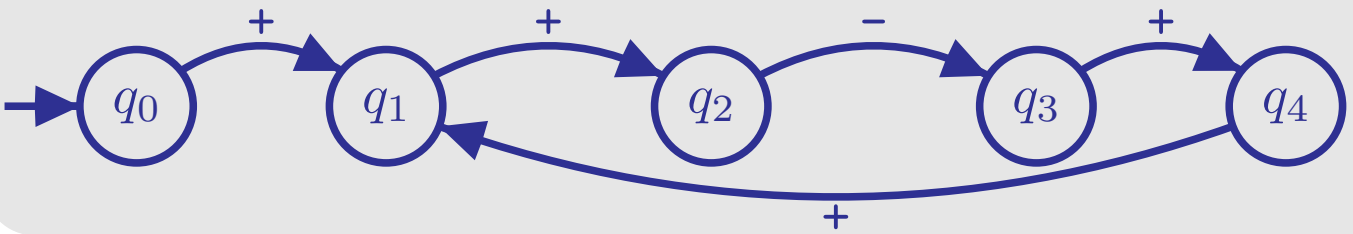
$$(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{+} (q_2, 2) \xrightarrow{-} (q_3, 1) \xrightarrow{+} (q_4, 2) \xrightarrow{+}$$
$$(q_1, 3) \xrightarrow{+} (q_2, 4) \xrightarrow{-} (q_3, 3) \xrightarrow{+} (q_4, 4) \xrightarrow{+}$$
$$(q_1, 5) \xrightarrow{+} (q_2, 6) \xrightarrow{-} (q_3, 5) \xrightarrow{+} (q_4, 6) \xrightarrow{+}$$
$$\ldots$$

$$\text{offset}(i, I) = \{j \mid c_{i+j} - c_i \in I\}$$

$$\begin{aligned} \text{e.g., } \text{offset}(2, [2, 3]) &= \{j \mid c_{2+j} - c_2 \in [2, 3]\} \\ &= \{j \mid c_{2+j} = 4 \text{ or } c_{2+j} = 5\} \\ &= \{4, 6, 7, 9\} \end{aligned}$$

Proof

The unique computation of M has a regular structure:

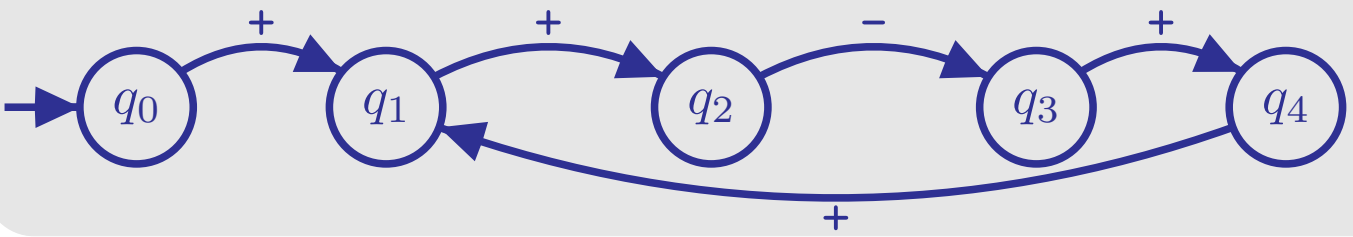


$$
\begin{aligned}
(q_0, 0) \xrightarrow{+} (q_1, 1) \xrightarrow{+} (q_2, 2) \xrightarrow{-} (q_3, 1) \xrightarrow{+} (q_4, 2) \xrightarrow{+} \\
(q_1, 3) \xrightarrow{+} (q_2, 4) \xrightarrow{-} (q_3, 3) \xrightarrow{+} (q_4, 4) \xrightarrow{+} \\
(q_1, 5) \xrightarrow{+} (q_2, 6) \xrightarrow{-} (q_3, 5) \xrightarrow{+} (q_4, 6) \xrightarrow{+} \\
\ldots
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{offset}(i, I) &= \{j \mid c_{i+j} - c_i \in I\} \\
\text{e.g., } \mathsf{offset}(2, [2,3]) &= \{j \mid c_{2+j} - c_2 \in [2,3]\} \\
&= \{j \mid c_{2+j} = 4 \text{ or } c_{2+j} = 5\} \\
&= \{4, 6, 7, 9\} \\
&= \mathsf{offset}(6, [2,3]) = \mathsf{offset}(10, [2,3]) = \ldots
\end{aligned}
$$

Proof

The unique computation of M has a regular structure:

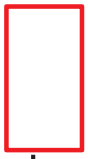


$$
\begin{array}{l}
(q_0,0) \xrightarrow{+} \boxed{(q_1,1)} \xrightarrow{+} \boxed{(q_2,2)} \xrightarrow{-} \boxed{(q_3,1)} \xrightarrow{+} \boxed{(q_4,2)} \xrightarrow{+} \\
\qquad\quad \boxed{(q_1,3)} \xrightarrow{+} \boxed{(q_2,4)} \xrightarrow{-} \boxed{(q_3,3)} \xrightarrow{+} \boxed{(q_4,4)} \xrightarrow{+} \\
\qquad\quad \boxed{(q_1,5)} \xrightarrow{+} \boxed{(q_2,6)} \xrightarrow{-} \boxed{(q_3,5)} \xrightarrow{+} \boxed{(q_4,6)} \xrightarrow{+} \\
\qquad\quad \boxed{\ldots}
\end{array}
$$

Configurations in $\boxed{}$ have the same behaviour with respect to formulae.

Proof

We define an equivalence relation $\equiv$ over the set of configurations of M.

-  form the equivalence classes induced by $\equiv$,
- the index of $\equiv$ is finite,
- each equivalence class can be symbolically represented in a finite manner,
- the symbolic representations and the subformulas of φ form the states of the Büchi automaton.

Proof

We define an equivalence relation $\equiv$ over the set of configurations of M.

- $\square$ form the equivalence classes induced by $\equiv$,
- the index of $\equiv$ is finite,
- each equivalence class can be symbolically represented in a finite manner,
- the symbolic representations and the subformulas of φ form the states of the Büchi automaton.

It holds that

there is a computation γ of M with $\gamma \models \varphi$

iff

there is a Büchi accepting run of A

Proof

We define an equivalence relation $\equiv$ over the set of configurations of M.

- $\square$ form the equivalence classes induced by $\equiv$,
- the index of $\equiv$ is finite,
- each equivalence class can be symbolically represented in a finite manner,
- the symbolic representations and the subformulas of φ form the states of the Büchi automaton.

It holds that

there is a computation γ of M with $\gamma \models \varphi$
iff
there is a Büchi accepting run of A

Theorem

Freeze LTL interval extension-model checking of deterministic 1-CM (1-VASS, WA) is decidable.

Open Questions

- Complexity of Model Checking Deterministic automata?
- Is Freeze LTL Interval Extension expressively stronger than MTL?
- What about MTL Model Checking Non-deterministic automata, if intervals are restricted to $[0, 0]$ (like in Freeze LTL)?

Open Questions

- Complexity of Model Checking Deterministic automata?
- Is Freeze LTL Interval Extension expressively stronger than MTL?
- What about MTL Model Checking Non-deterministic automata, if intervals are restricted to $[0, 0]$ (like in Freeze LTL)?

Thank you for your attention!