Reachability Problems 2019

# On the Termination of Counter Machines with Incrementing Errors


Christopher Hampson

*Department of Informatics*

*King's College London*

# (Reliable) Counter Machines

- **Counter Machine (CM)** $\mathcal{M} = \langle\, Q,\ C,\ q_{\mathsf{init}},\ \Delta \,\rangle$ *Minsky 1967*

# (Reliable) Counter Machines

- **Counter Machine (CM)** $\mathcal{M} = \langle\, Q,\ C,\ q_{\mathsf{init}},\ \Delta\, \rangle$ *Minsky 1967*
  - **Finite set of control states** $Q = \{q_1, q_2, \ldots, q_n\}$
  - **Initial state** $q_{\mathsf{init}} \in Q$
  - **Finite set of counters** $C = \{c_1, c_2, \ldots, c_m\}$
  - **Finite set of State Transitions** $\Delta \subseteq Q \times Op_C \times Q$

# (Reliable) Counter Machines

- **Counter Machine (CM)** $\mathcal{M} = \langle\, Q,\; C,\; q_{\mathsf{init}},\; \Delta \,\rangle$ *Minsky 1967*
  - Finite set of control states $Q = \{q_1, q_2, \ldots, q_n\}$
  - Initial state $q_{\mathsf{init}} \in Q$
  - Finite set of counters $C = \{c_1, c_2, \ldots, c_m\}$
  - Finite set of State Transitions $\Delta \subseteq Q \times Op_C \times Q$
    - Increment $(c_i)^{++} \in Op_C$
    - Decrement $(c_i)^{--} \in Op_C$
    - Emptiness check $(c_i)^{??} \in Op_C$

    } for all $c_i \in C$

# Lossy / Incrementing Counter Machines

*Mayr 2003*

- **Lossy Conter Machines (LCMs)**
  - LCMs are counter machines whose possible **computations / runs** are determined by the following **consecution relation**:

$$\sigma_0 \xrightarrow{\mathcal{M}\downarrow} \sigma_1 \quad \iff \quad \exists \sigma'_0 \, \exists \sigma'_1 \left( \sigma_0 \geq \sigma'_0 \xrightarrow{\mathcal{M}} \sigma'_1 \geq \sigma_1 \right)$$

where

$$(q, \vec{v}) \geq (q', \vec{v}') \quad \iff \quad q = q' \text{ and } \vec{v}(c_i) \geq \vec{v}'(c_i) \text{ for all } c_i \in C$$

**(counters may spontaneously *decrease* before / after every operation)**

# Lossy / Incrementing Counter Machines

*Demri–Lazić 2009, Ouaknine–Worrell 2006, Ouaknine–Worrell 2007*

- **Incrementing Counter Machines (ICMs)**
  - ICMs are counter machines whose possible **computations / runs** are determined by the following **consecution relation**:

$$\sigma_0 \xrightarrow{\mathcal{M}\downarrow} \sigma_1 \quad \Longleftrightarrow \quad \exists \sigma_0' \, \exists \sigma_1' \left( \sigma_0 \leq \sigma_0' \xrightarrow{\mathcal{M}} \sigma_1' \leq \sigma_1 \right)$$

where

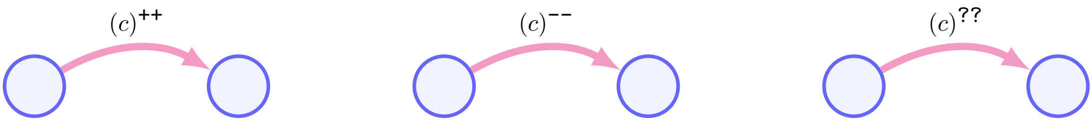
$$(q, \vec{v}) \leq (q', \vec{v}') \quad \Longleftrightarrow \quad q = q' \text{ and } \vec{v}(c_i) \leq \vec{v}'(c_i) \text{ for all } c_i \in C$$

**(counters may spontaneously *increase* before / after every operation)**

# Lossy / Incrementing Counter Machines

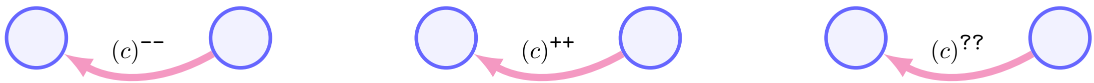
- **Dual Counter Machine**

  - The **dual / opposite** of $\mathcal{M}$ is given by reversing all transitions



# Lossy / Incrementing Counter Machines

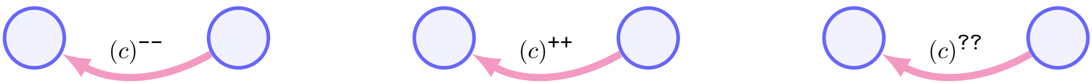
- **Dual Counter Machine**
  - The **dual / opposite** of $\mathcal{M}$ is given by reversing all transitions




**(The dual behaves as if the *arrow of time* has been reversed)**

- **Dual Counter Machine**

  - The dual / opposite of $\mathcal{M}$ is given by reversing all transitions




**(The dual behaves as if the *arrow of time* has been reversed)**

**Theorem (Ouknine–Worrel)** Let $\mathcal{M}$ and $\mathcal{M}^{\mathbf{op}}$ be dual counter machines. Then, for all configurations $\sigma_0, \sigma_1 \in \mathbf{Conf}_{\mathcal{M}}$,

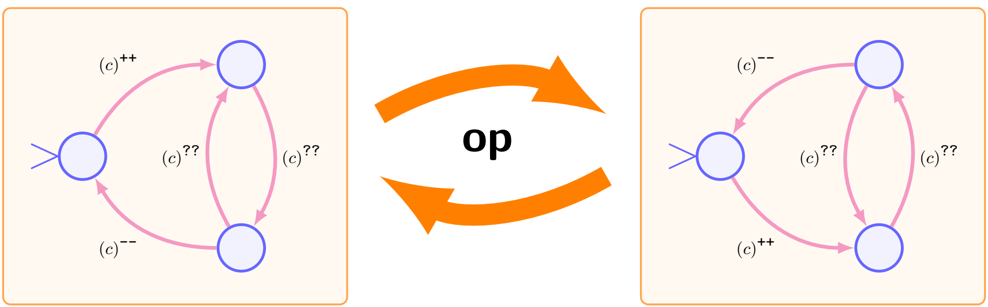
$$\underbrace{\sigma_0 \xrightarrow{\mathcal{M}\downarrow} \dots \xrightarrow{\mathcal{M}\downarrow} \sigma_1}_{\text{lossy}} \iff \underbrace{\sigma_1 \xrightarrow{\mathcal{M}^{\mathrm{op}}\uparrow} \dots \xrightarrow{\mathcal{M}^{\mathrm{op}}\uparrow} \sigma_0}_{\text{incrementing}}$$

## Lossy / Incrementing Counter Machines

**Theorem** Reachability for LCMs and ICMs share the same complexity

**Theorem** Reachability for LCMs and ICMs share the same complexity

## However, symmetry is broken for *Termination*!




**Has non-terminating lossy run**

**Has NO non-terminating incrementing run**

## Summary of Known Termination Results

| | *Lossy* | *Incrementing* |
|---|---|---|
| Channel Systems (with emptiness testing) | HYPERACKERMANN-complete<br>*Chambart–Schnoebelen 2008* | TOWER-complete<br>*Bouyer et al. 2012* |
| Counter Machines | ACKERMANN-complete<br>*Schnoebelen 2010* | **??** |
| Counter Machines (with $k$ counters) | non-ELEMENTARY for $k > 5$<br>*Schnoebelen 2010* | **??** |

**(counter machines can be seen as degenerate channel systems)**

# Main Results

## (a.k.a. Filling in the gaps)

## Main Results – Upper bound

**Theorem** The ICM termination problem is decidable in EXPSPACE

**Theorem** The ICM termination problem is decidable in EXPSPACE

**Proof (sketch):**

**Step 1)** Let $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$ be a **terminating** ICM and let $r$ be an incrementing run of $\mathcal{M}$.

**Question**

How long can $r$ be if $\mathcal{M}$ has no infinite runs?

## Main Results – Upper bound

**Theorem** The ICM termination problem is decidable in EXPSPACE

**Proof (sketch):**

**Step 1)** Let $\mathcal{M} = \langle Q, C, q_{\text{init}}, \Delta \rangle$ be a **terminating** ICM and let $r$ be an incrementing run of $\mathcal{M}$.

**Question**

How long can $r$ be if $\mathcal{M}$ has no infinite runs?

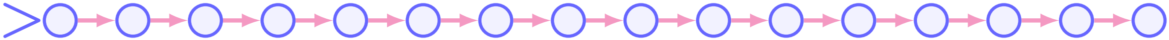
**Step 2)** Note that the only impediments to **infinitely long** runs are transitions of the form $(s, (c)^{??}, t) \in \Delta$.

**(call these transitions $c$-gates)**

## Main Results – Upper bound

**Step 3)** Show by induction the size of $\Sigma \subseteq C$ that **long intervals** containing only $\Sigma$-gates have **repeated (partial) configurations**

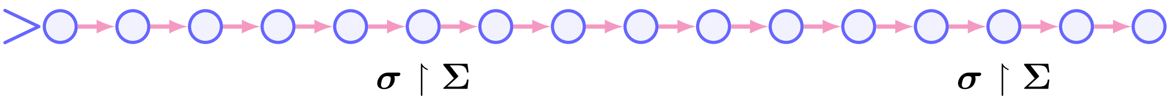
**(we can safely ignore counters not appearing in $\Sigma$)**



# Main Results – Upper bound

**Step 3)** Show by induction the size of $\Sigma \subseteq C$ that **long intervals** containing only $\Sigma$-gates have **repeated (partial) configurations**

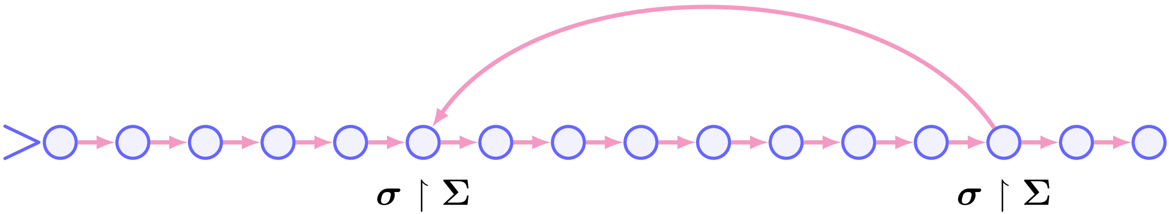
**(we can safely ignore counters not appearing in $\Sigma$)**



# Main Results – Upper bound

**Step 3)** Show by induction the size of $\Sigma \subseteq C$ that **long intervals** containing only $\Sigma$-gates have **repeated (partial) configurations**

**(we can safely ignore counters not appearing in $\Sigma$)**



**Step 4)** We can show that

| | | |
|---|---|---|
| max_length of interval with only $c$-gates, for $c \in \Sigma$ | $<$ | $n^{T(\|\Sigma\|)}$ |

where

$$T(0) = 1 \qquad \text{and} \qquad T(k) = k\,T(k-1) + 2$$

## Main Results – Upper bound

**Step 4)** We can show that

$$\text{max\_length of interval with only } c\text{-gates, for } c \in \Sigma \quad < \quad n^{T(|\Sigma|)}$$

where

$$T(0) = 1 \qquad \text{and} \qquad T(k) = k\,T(k-1) + 2$$

By induction

$$T(k) \;=\; k!\left(\frac{1}{0!} + \frac{2}{1!} + \cdots + \frac{2}{k!}\right) \;<\; 2k!\sum_{t=0}^{\infty}\frac{1}{t!} \;=\; 2ek!$$

**(where $e \approx 2.7182$ is Euler's constant)**

## Main Results – Upper bound

**Step 5)** It then follows that

$$\text{max\_length}(\boldsymbol{r}) \;\leq\; \boldsymbol{n}^{2em!} \;\in\; \boldsymbol{O}\left(2^{2^m}\right)$$

**(since $r$ contains only $C$-gates, and $|C| = m$)**

## Main Results – Upper bound

**Step 5)** It then follows that

$$\text{max\_length}(\boldsymbol{r}) \;\leq\; \boldsymbol{n}^{2em!} \;\in\; \boldsymbol{O}\left(2^{2^m}\right)$$

**(since $r$ contains only $C$-gates, and $|C| = m$)**

$\rightsquigarrow$ **non-deterministic search for a 'long' run requires only *exponential* space**

**Q.E.D**

# Main Results – Lower bound

**Theorem** The ICM termination problem is **PSPACE-hard**

# Main Results – Lower bound

**Theorem** The ICM termination problem is **PSPACE-hard**

**Proof (sketch):**

**Step 1)** Choose your favourite **reliable** counter machine $\mathcal{M}$ and some specified **number of bits** $N > 0$.

## Main Results – Lower bound

> **Theorem** The ICM termination problem is **PSPACE-hard**

**Proof (sketch):**

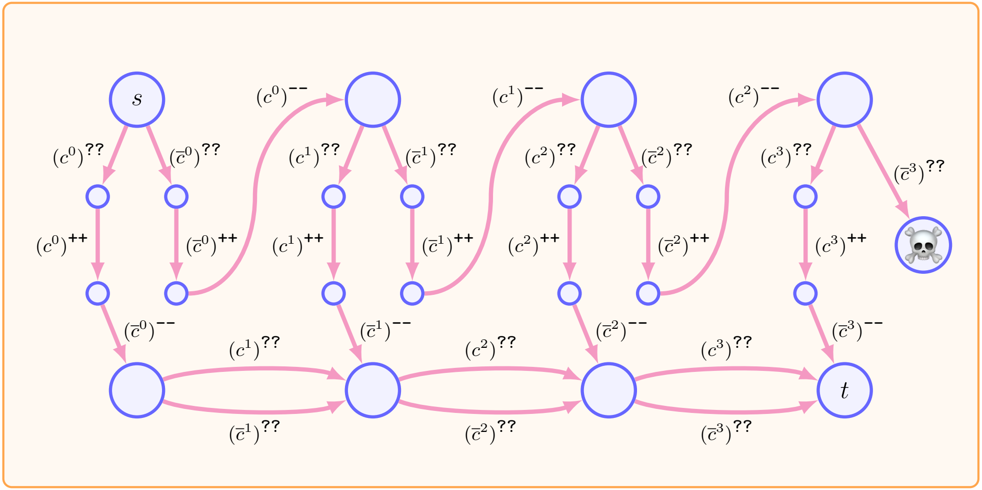
**Step 1)** Choose your favourite **reliable** counter machine $\mathcal{M}$ and some specified **number of bits** $N > 0$.

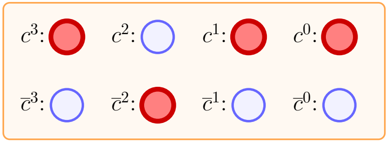
**Step 2)** Introduce counters $c^0, \ldots, c^{N-1}, \overline{c}^0, \ldots, \overline{c}^{N-1}$ for each $c \in C$, and let

$$\theta_{\vec{v}}(c) = \sum_{j=0}^{N-1} 2^j \min\left\{1,\ \vec{v}(c^j)\right\}$$

**($\theta$ is the value represented by the non-emptiness of $c^0, \ldots, c^{N-1}$ in binary)**

**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

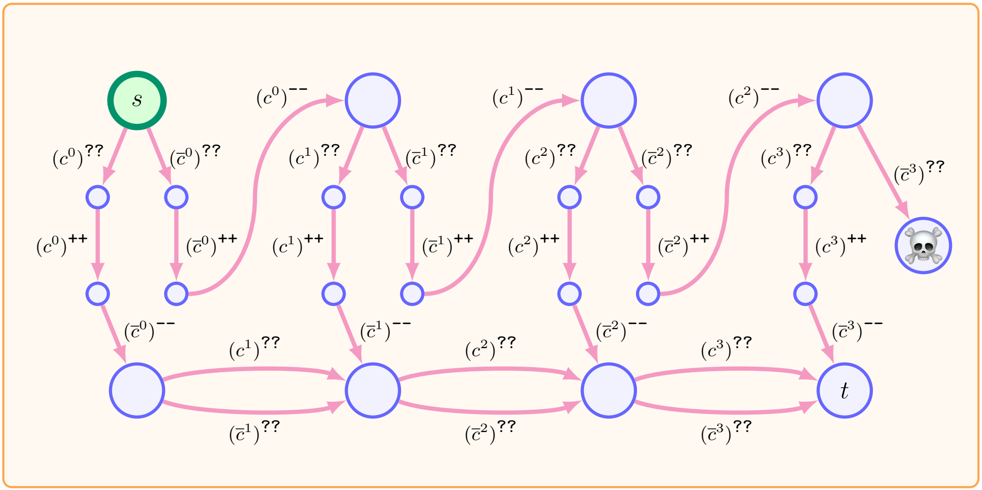


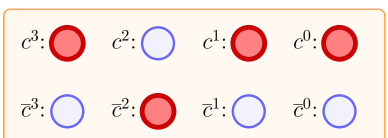




**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

# Main Results – Lower bound

**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

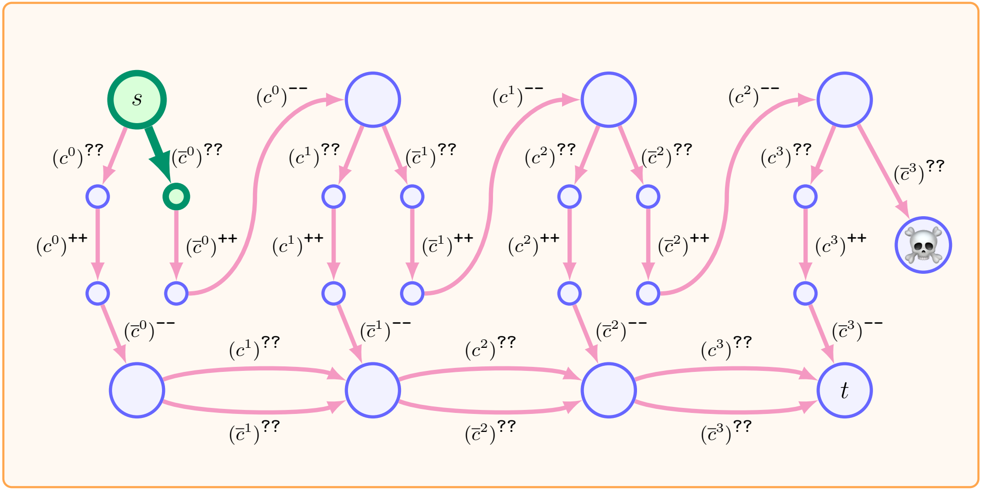


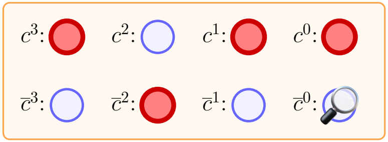




**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

# Main Results – Lower bound

**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

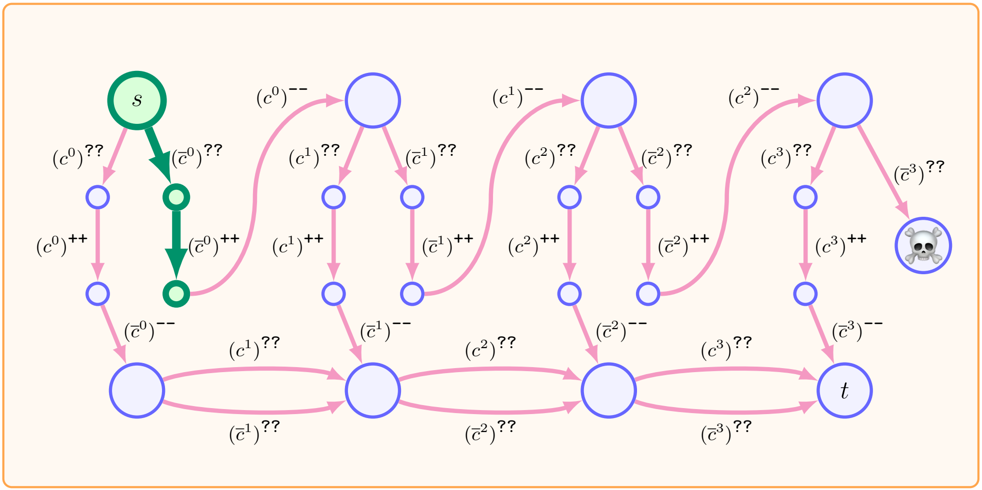


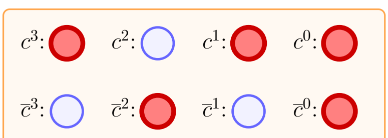




**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

# Main Results – Lower bound

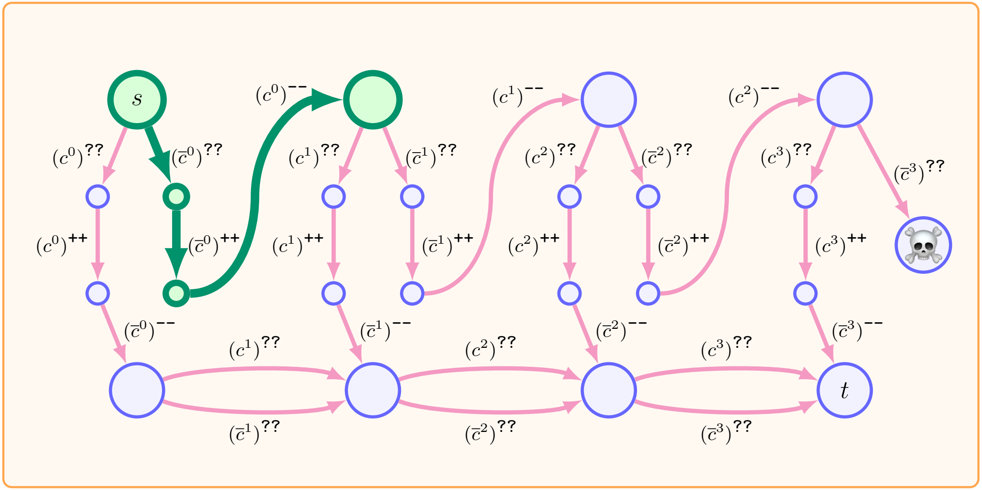
**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

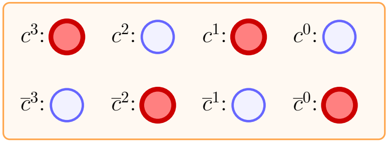



**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

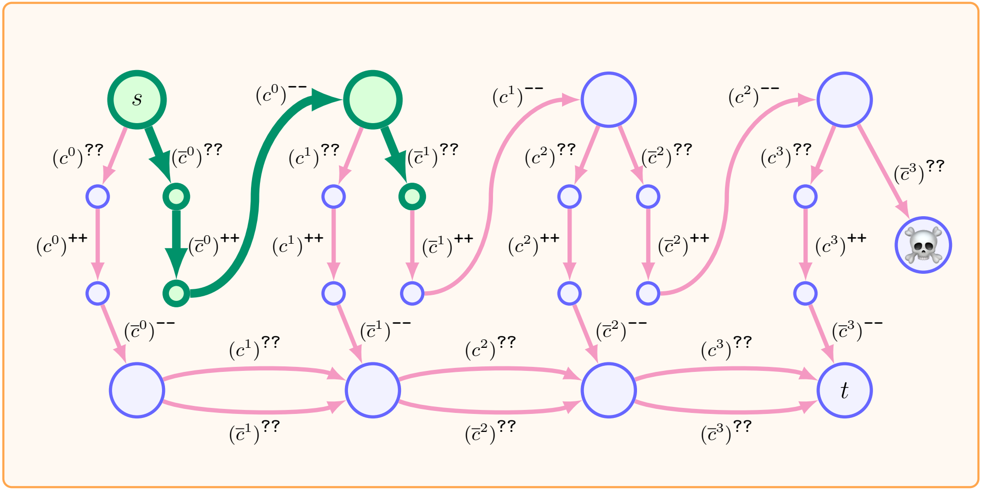
**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

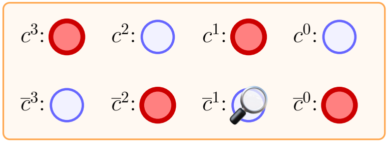



(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)

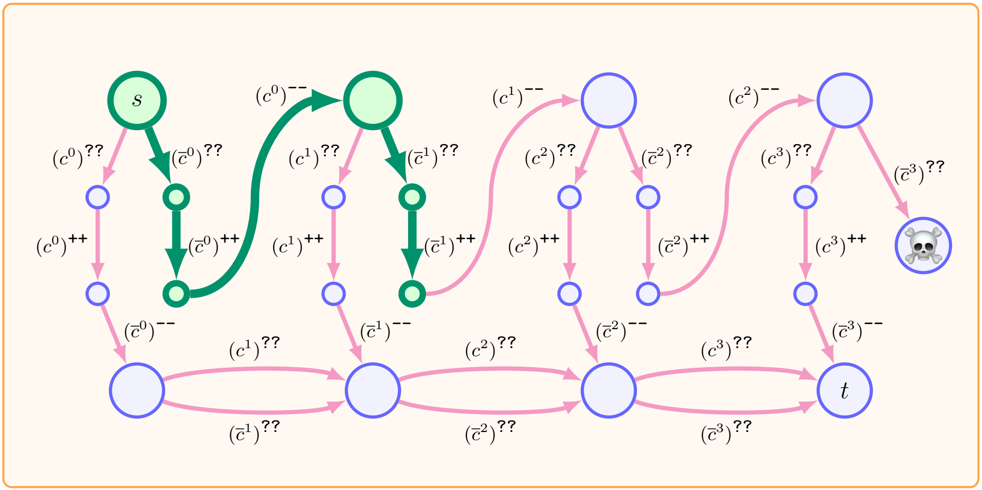
**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

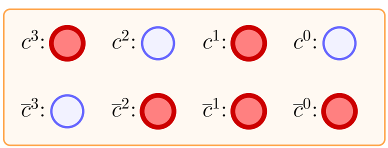



**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

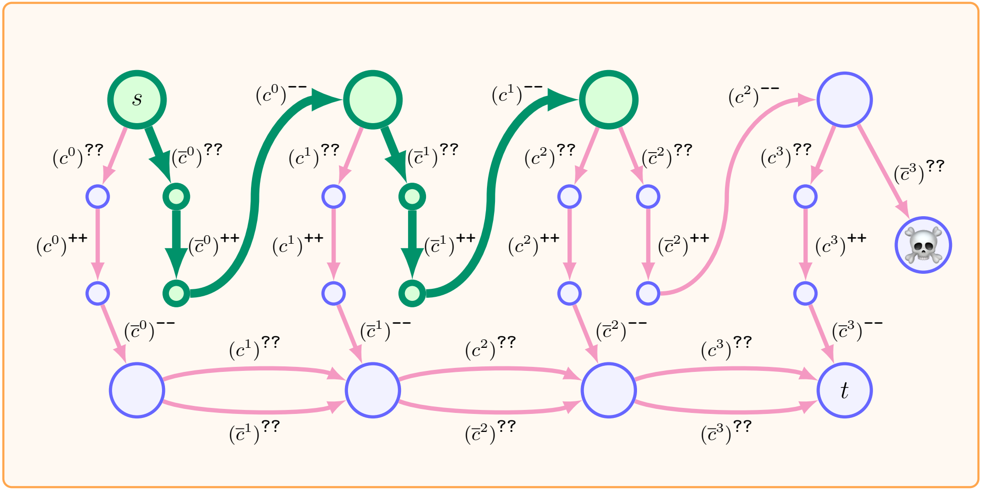
**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

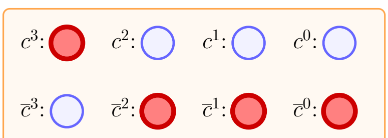



**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

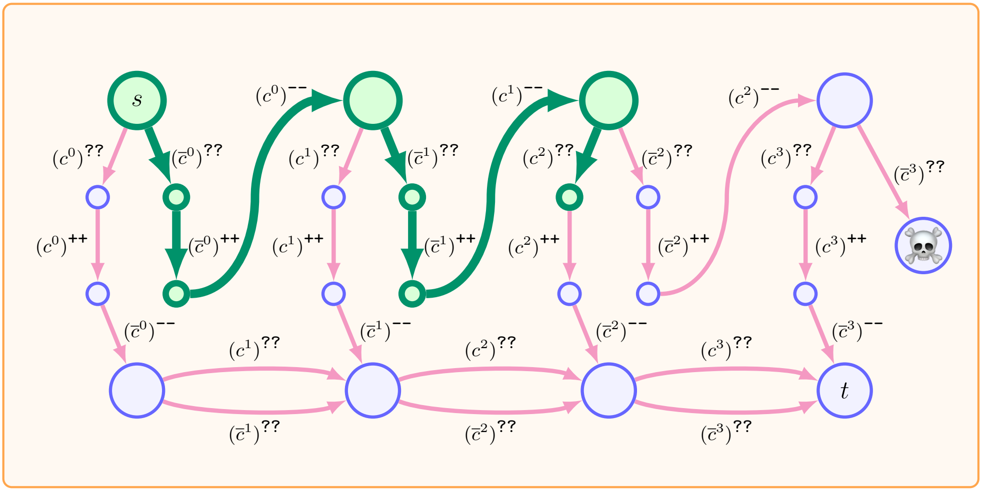


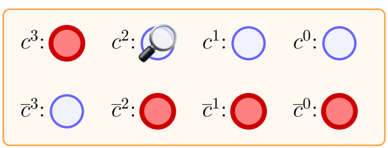




**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

# Main Results – Lower bound

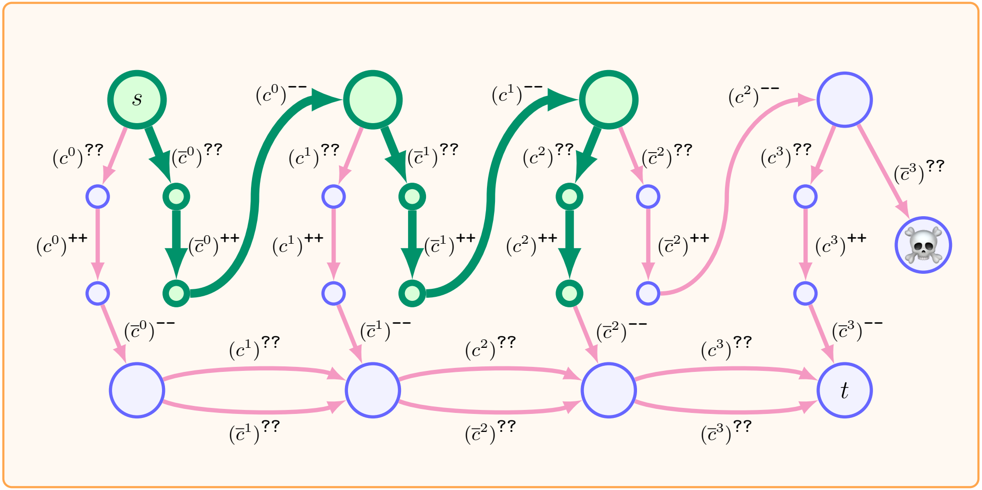
**Step 3a)** Replace each increment transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

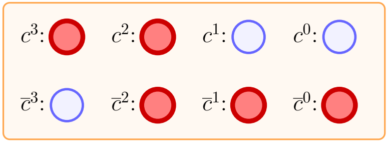



**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

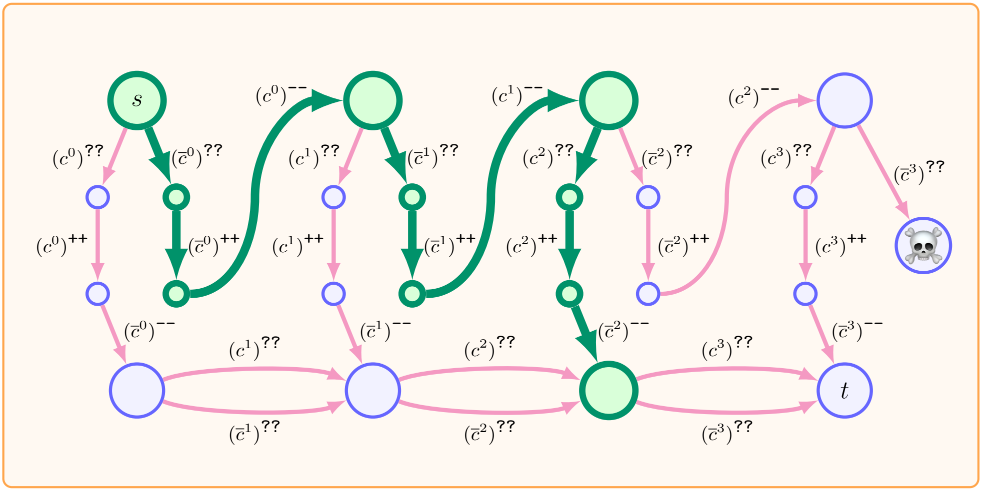
**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

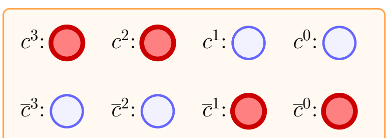



**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

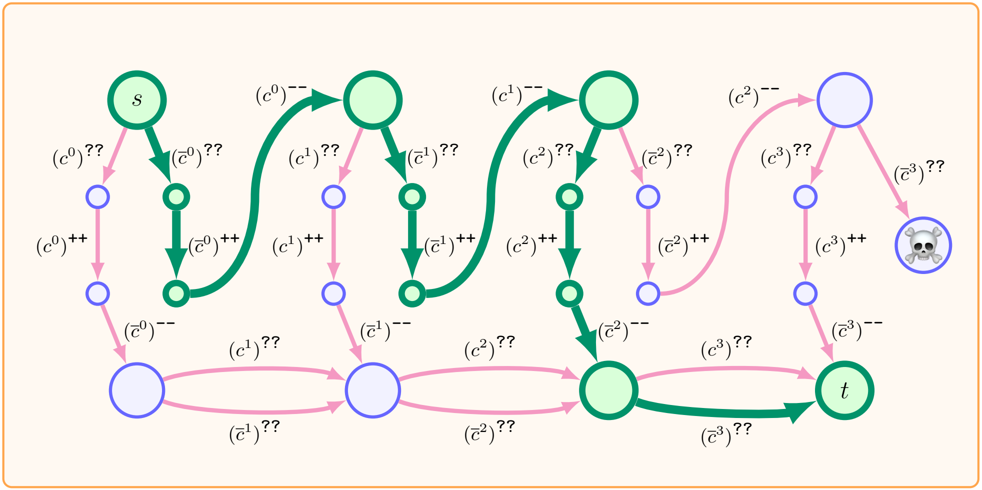


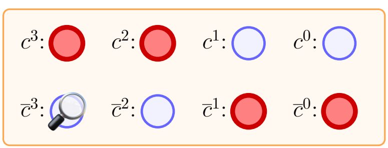




(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)

# Main Results – Lower bound

**Step 3a)** Replace each **increment** transition $(s, (c)^{++}, t) \in \Delta$ with the following widget:

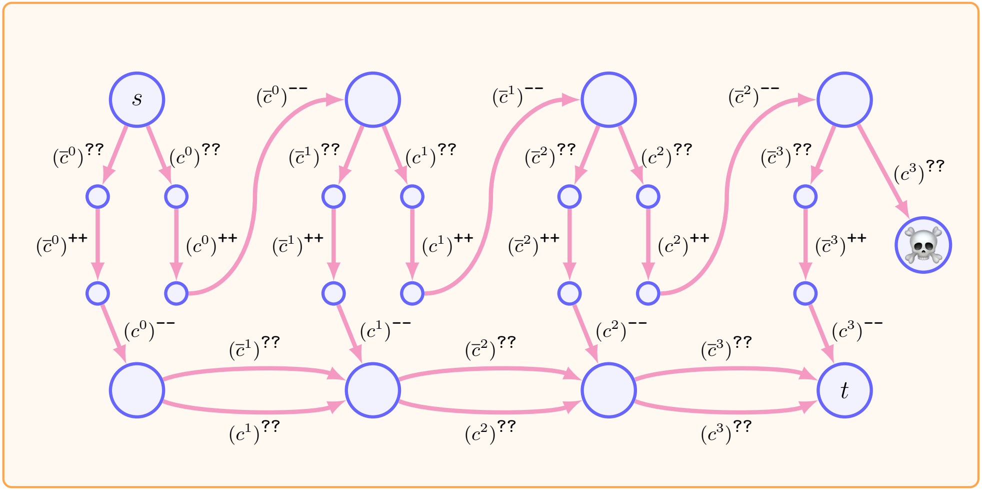



**(traversable iff $\theta_t(c) \geq \theta_s(c) + 1$)**

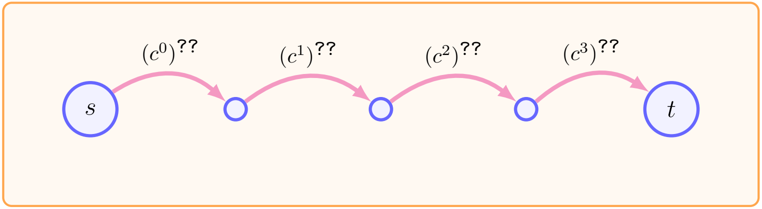
# Main Results – Lower bound

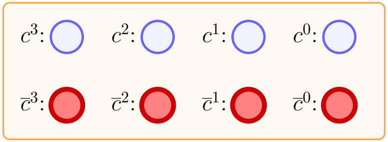
**Step 3b)** Replace each **decrement** transition $(s, (c)^{--}, t) \in \Delta$ with the following widget:




**(traversable iff $\theta_t(c) \geq \theta_s(c) - 1$)**

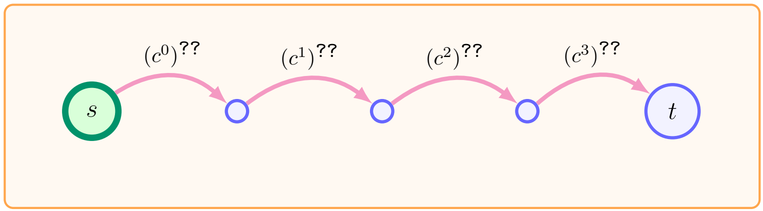
**Step 3c)** Replace each **emptiness-check** transition $(s, (c)^{??}, t) \in \Delta$ with the following widget:

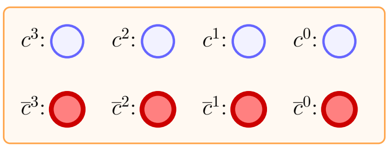



**(traversable iff $\theta_s(c) = 0$)**

**Step 3c)** Replace each **emptiness-check** transition $(\boldsymbol{s}, (\boldsymbol{c})^{??}, \boldsymbol{t}) \in \boldsymbol{\Delta}$ with the following widget:

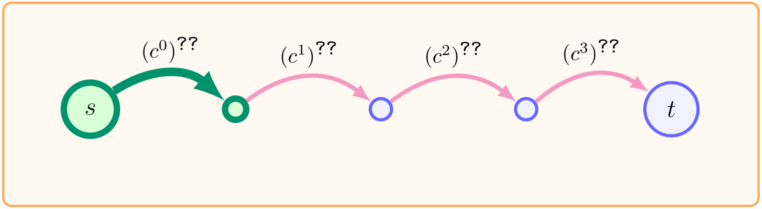


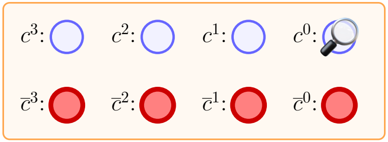




**(traversable iff $\theta_s(c) = 0$)**

# Main Results – Lower bound

**Step 3c)** Replace each **emptiness-check** transition $(s, (c)^{??}, t) \in \Delta$ with the following widget:

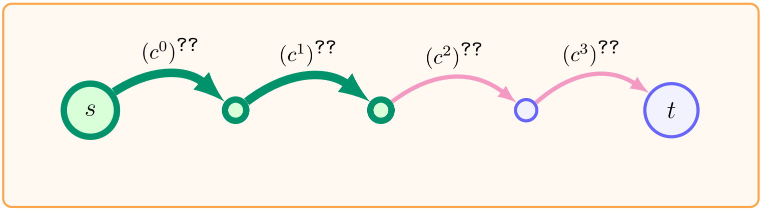


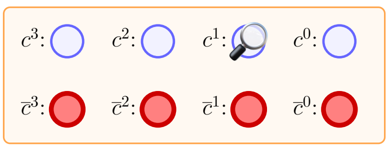




**(traversable iff $\theta_s(c) = 0$)**

## Main Results – Lower bound

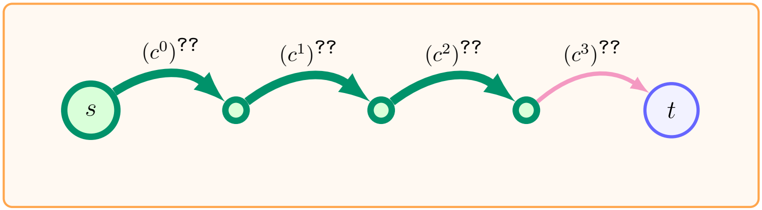
**Step 3c)** Replace each **emptiness-check** transition $(s,(c)^{??},t) \in \Delta$ with the following widget:

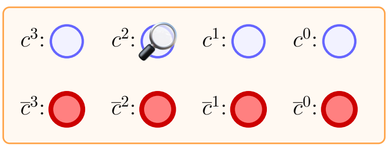



**(traversable iff $\theta_s(c) = 0$)**

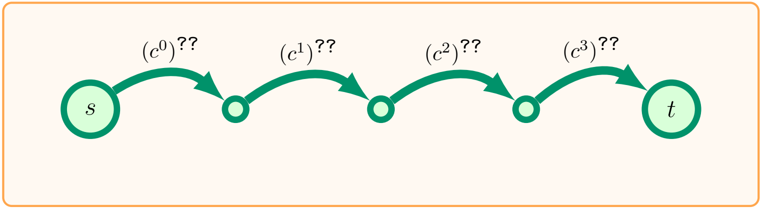
**Step 3c)** Replace each **emptiness-check** transition $(s, (c)^{??}, t) \in \Delta$ with the following widget:

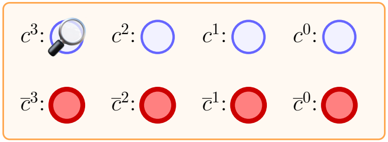



**(traversable iff $\theta_s(c) = 0$)**

**Step 3c)** Replace each **emptiness-check** transition $(s, (c)^{??}, t) \in \Delta$ with the following widget:




**(traversable iff $\theta_s(c) = 0$)**

# Main Results – Lower bound

**Step 4)** It follows that:

| | | |
|---|---|---|
| $\mathcal{M}'$ has a non-terminating **incrementing** run | $\iff$ | $\mathcal{M}$ has a non-terminating **reliable** run whose counters do not exceed $2^N - 1$ |

## Main Results – Lower bound

**Step 4)** It follows that:

| $\mathcal{M}'$ has a non-terminating **incrementing** run | $\Longleftrightarrow$ | $\mathcal{M}$ has a non-terminating **reliable** run whose counters do not exceed $2^N - 1$ |
|---|---|---|

**Step 5)** PSPACE-hardness follows from the following claim

**Claim**

The termination problem for relaible counter machines with "exponentially bounded counters" is PSPACE-hard.

**Q.E.D**

## Main Results – Fixed number of counters

**Theorem** The $k$-counter ICM termination problem is **NLogSpace-complete**

## Main Results – Fixed number of counters

**Theorem** The $k$-counter ICM termination problem is **NLOGSPACE-complete**

**Proof:**

- **Lower Bound)** Trivial **(via a reduction from Graph Reachability)**

## Main Results – Fixed number of counters

**Theorem** The $k$-counter ICM termination problem is **NLogSpace-complete**

**Proof:**

- **Lower Bound)** Trivial **(via a reduction from Graph Reachability)**
- **Upper Bound)** With $|C| = k$ fixed

$$\text{max\_length} \leq n^{2ek!} \in O(n)$$

## Main Results – Fixed number of counters

**Theorem** The $k$-counter ICM termination problem is **NLogSpace-complete**

**Proof:**

- **Lower Bound)** Trivial **(via a reduction from Graph Reachability)**
- **Upper Bound)** With $|C| = k$ fixed

$$\text{max\_length} \leq n^{2ek!} \in O(n)$$

$\rightsquigarrow$ **non-deterministic search requires only logarithmic space**

**Q.E.D**

## Summary of Known Results

| | *Lossy* | *Incrementing* |
|---|---|---|
| Channel Systems (with emptiness testing) | HYPERACKERMANN-complete<br>*Chambart–Schnoebelen 2008* | TOWER-complete<br>*Bouyer et al. 2012* |
| Counter Machines | ACKERMANN-complete<br>*Schnoebelen 2010* | PSPACE-hard in EXPSPACE<br>*H 2019* |
| Counter Machines (with $k$ counters) | non-ELEMENTARY for $k > 5$<br>*Schnoebelen 2010* | NLOGSPACE-complete<br>*H 2019* |

# Recent Developments . . .

# Some recent (unpublished) developments

**Theorem** The ICM termination problem is **EXPSPACE-complete**

# Some recent (unpublished) developments

**Theorem** The ICM termination problem is **EXPSPACE-complete**

**Proof (sketch):**

**Step 1)** Using the decrement gadget above, we can construct a **controlled loop** that visits a given state *exactly* $2^n$ times (or terminates prematurely).

**(set all bits to 1 then decrement repeatedly until all bits are 0)**

## Some recent (unpublished) developments

**Theorem** The ICM termination problem is **EXPSPACE-complete**

**Proof (sketch):**

**Step 1)** Using the decrement gadget above, we can construct a **controlled loop** that visits a given state *exactly* $2^n$ times (or terminates prematurely).

**(set all bits to 1 then decrement repeatedly until all bits are 0)**

**Claim**

It is possible to construct a **controlled loop** that visits a given state *exactly* $2^{2^n}$ times (or terminates prematurely).

## Some recent (unpublished) developments

**Step 2)** Take a **reliable** machine $\mathcal{M}$ and construct an ICM $\mathcal{M}'$ by replacing every **increment** with the following subroutine:

**Increment $c_i$ subroutine**

- Increment $\boldsymbol{c_i}$ and decrement $\boldsymbol{d_i}$,
- Use a **controlled loop** to transfer $(\boldsymbol{c_i}, \boldsymbol{d_i}) \mapsto (\mathbf{temp}, \overline{\mathbf{temp}})$
- Check that $(\boldsymbol{c_i}, \boldsymbol{d_i}) = (0, 0)$, else terminate.
- Use a **controlled loop** to transfer $(\mathbf{temp}, \overline{\mathbf{temp}}) \mapsto (\boldsymbol{c_i}, \boldsymbol{d_i})$
- Check that $(\mathbf{temp}, \overline{\mathbf{temp}}) = (0, 0)$, else terminate.

**(the sum of $c$ and $d$ remains constant and any errors result in termination)**

## Some recent (unpublished) developments

**Step 2)** Take a **reliable** machine $\mathcal{M}$ and construct an ICM $\mathcal{M}'$ by replacing every **increment** with the following subroutine:

**Increment $c_i$ subroutine**

- Increment $\boldsymbol{c_i}$ and decrement $\boldsymbol{d_i}$,
- Use a **controlled loop** to transfer $(\boldsymbol{c_i}, \boldsymbol{d_i}) \mapsto (\textbf{temp}, \overline{\textbf{temp}})$
- Check that $(\boldsymbol{c_i}, \boldsymbol{d_i}) = (0, 0)$, else terminate.
- Use a **controlled loop** to transfer $(\textbf{temp}, \overline{\textbf{temp}}) \mapsto (\boldsymbol{c_i}, \boldsymbol{d_i})$
- Check that $(\textbf{temp}, \overline{\textbf{temp}}) = (0, 0)$, else terminate.

**(the sum of $c$ and $d$ remains constant and any errors result in termination)**

**Step 3)** Next replace every **decrement** with the analogous subroutine.

**Step 4)** It follows that:

| | | |
|---|---|---|
| $\mathcal{M}'$ has a non-terminating **incrementing** run | $\iff$ | $\mathcal{M}$ has a non-terminating **reliable** run whose counters do not exceed $2^{2^n}$ |

**Q.E.D**

## Summary of Known Results

| | *Lossy* | *Incrementing* |
|---|---|---|
| Channel Systems (with emptiness testing) | HyperAckermann-complete<br>*Chambart–Schnoebelen 2008* | Tower-complete<br>*Bouyer et al. 2012* |
| Counter Machines | Ackermann-complete<br>*Schnoebelen 2010* | ExpSpace-complete<br>**NEW** |
| Counter Machines (with $k$ counters) | non-Elementary for $k > 5$<br>*Schnoebelen 2010* | NLogSpace-complete<br>*H 2019* |

# End of Slides!

🎉

## Some references


- P. Bouyer, N. Markey, J.Ouaknine, P. Schnoebelen and J. Worrell. On termination and invariance for faulty channel machines. **Formal Aspects of Computing**, 24(1):595–607, 2012.
- M. L. Minsky. Computation. Prentice-Hall Englewood Cliffs, 1967.
- R. Mayr. Undecidable problems in unreliable computations. **Theoretical Computer Science**, 297(1-3):337–354, 2003.
- J. Ouaknine and J. Worrell. On metric temporal logic and faulty Turing machines. In **International Conference on Foundations of Software Science and Computation Structures**, pages 217–230. Springer, 2006.
- J. Ouaknine and J. Worrell. On the Decidability and Complexity of Metric temporal logic over finite words. **Logical Methods in Computer Science**, 3(1), 2007.
- P. Chambart and P. Schnoebelen. The ordinal recursive complexity of lossy channel systems. In **2008 23rd Annual IEEE Symposium on Logic in Computer Science**, pages 205–216. IEEE, 2008.
- S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. **ACM Transactions on Computational Logic (TOCL)**, 10(3):16, 2009.
- P. Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In **International Symposium on Mathematical Foundations of Computer Science**, pages 616–628. Springer, 2010.