Coverability in VASS Revisited: Improving Rackoff's Bound to Obtain Conditional Optimality

Henry Sinclair-Banks

University of Warwick

United Kingdom

About joint work with Marvin Künnemann, Filip Mazowiecki, Lia Schütze, and Karol Węgrzycki in ICALP'23.



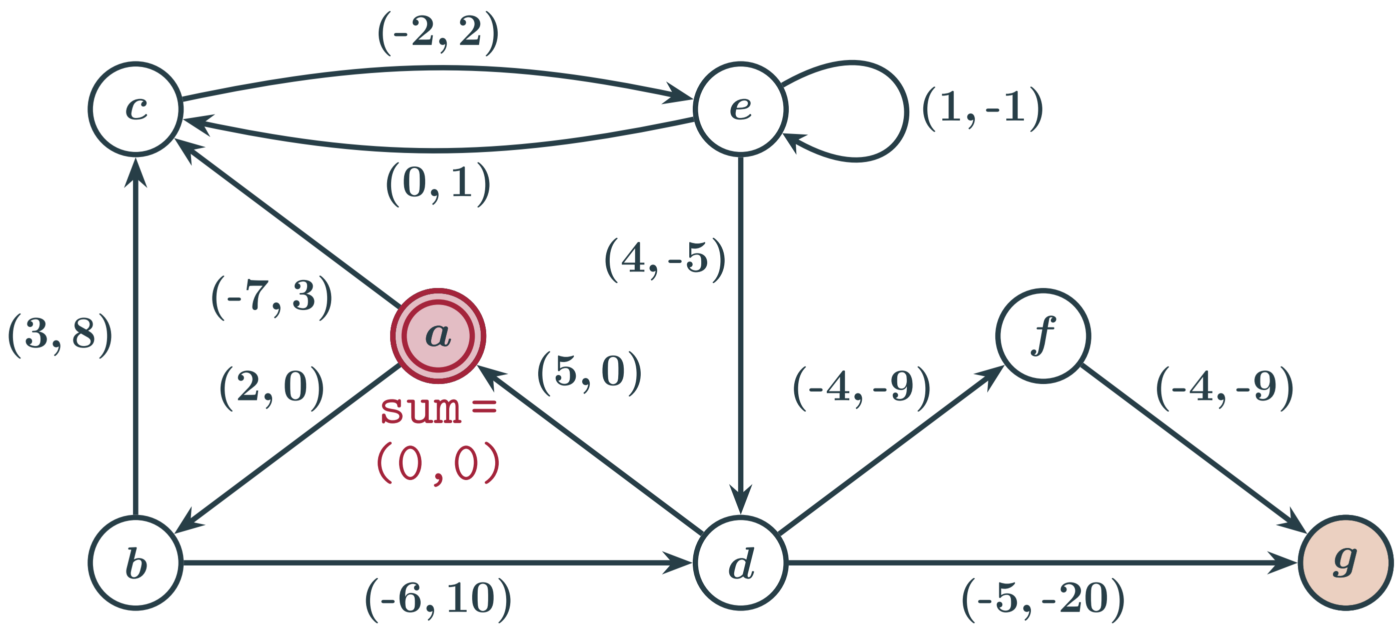


Formal Methods Seminar (M2F)

7th November 2023

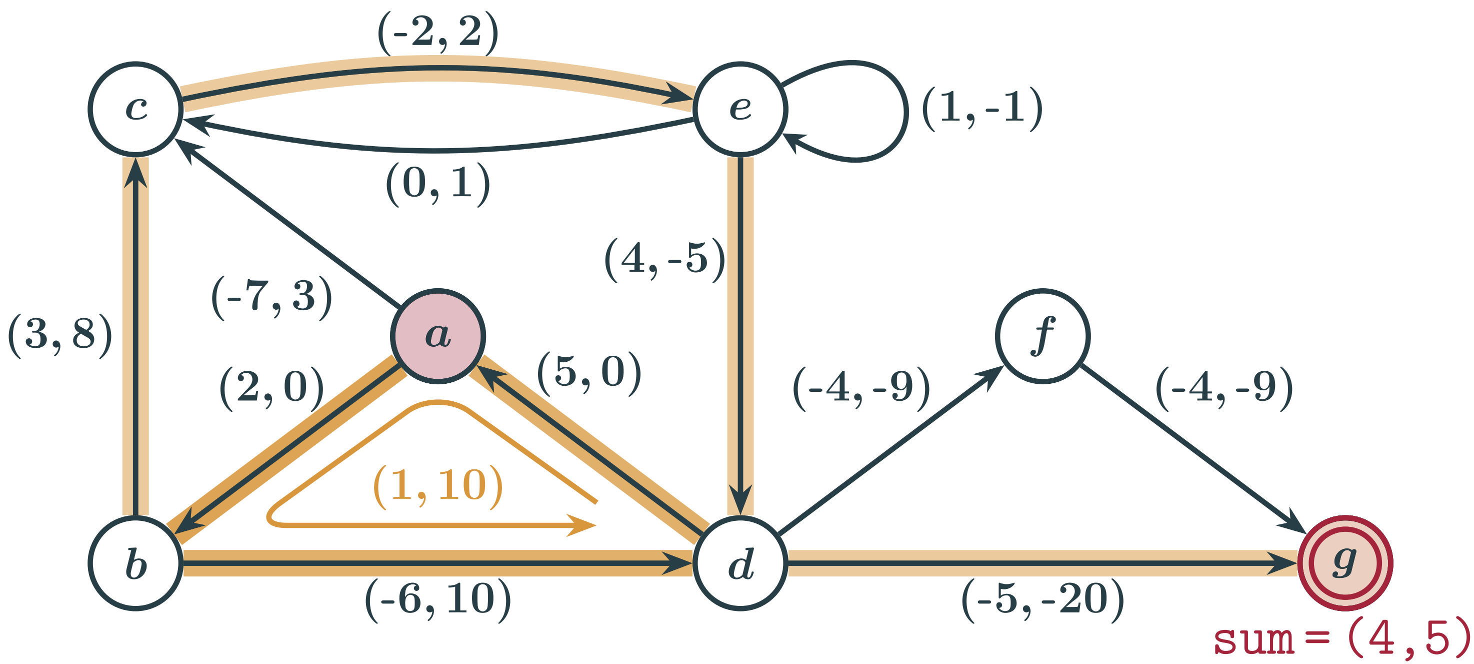
LaBRI, Bordeaux, France

Instance of Coverability in 2-Dimensional VASS



Question: from a can you reach g via a path that is *never negative on any component* ?

Instance of Coverability in 2-Dimensional VASS

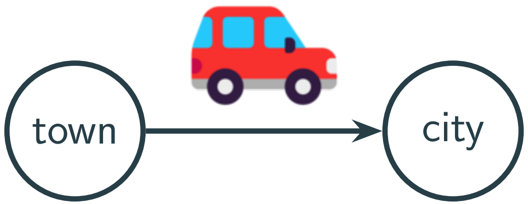


YES!

Question: from a can you reach g via a path that is *never negative on any component* ?

Motivation

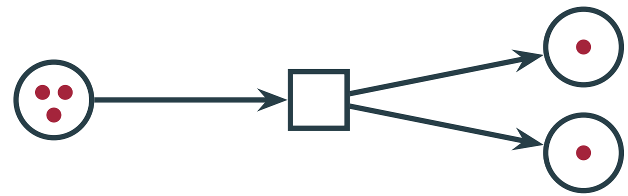
Resource Management



Road cost: (−1L fuel, +2kWh battery)

Model of Concurrency

VASS are equivalent to Petri nets



Testing Safety

Positive instance of coverability

⇓

Some action sequence reaches a ‘bad’ state

⇓

System is unsafe!

Related Problems

Unboundedness

Reachability

Word problems for (commutative) semi-groups

Overview of this Presentation

1. The history and complexity of coverability.

2. Our improvement over Rackoff's upper bound.

   Main concepts: introducing 'thin configurations' and using Rackoff's bounding technique.

3. Obtaining an optimal space algorithm and a conditionally optimal time algorithm.

4. Our Exponential Time Hypothesis conditional lower bound.

   Main concepts: reducing clique detection to coverability and simulating bounded counter machines.

History and Complexity

$\boldsymbol{d}$ is the dimension: number of components.

$\boldsymbol{n}$ is the size: number of states plus the absolute values of all updates.
(unary encoding)

History and Complexity

Richard Lipton

Theorem: Coverability in VASS is EXPSPACE-hard. [Lipton '76]

Charles Rackoff

Theorem: Coverability in VASS is in EXPSPACE. [Rackoff '78]

d is the dimension: number of components.

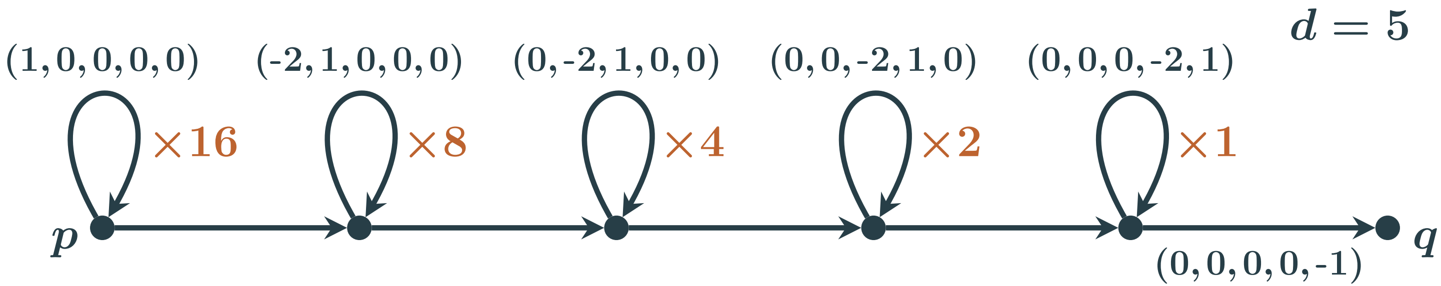
n is the size: number of states plus the absolute values of all updates (unary encoding).

History and Complexity

Richard Lip

[Lipton '76]

Example of Long Coverability Runs



Any coverability run from p to q has length $2^{\Omega(d)}$.

d is the dimension: number of components.

n is the size: number of states plus the absolute values of all updates (unary encoding).

History and Complexity

Richard Lipton

Theorem: Coverability in VASS requires $2^{\Omega(d)} \cdot \log(n)$ space. [Lipton '76]

Idea: find instances only admitting $n^{2^{\Omega(d)}}$ length runs. *"Lipton's construction"*

Charles Rackoff

Theorem: Coverability in VASS can be decided in $2^{\mathcal{O}(d \log d)} \cdot \log(n)$ space. [Rackoff '78]

Idea: argue that there are always $n^{2^{\mathcal{O}(d \log d)}}$ length runs. *"Rackoff's bound"*

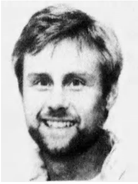
Ernst Mayr

Albert Meyer

Open Problem

Improve these bounds.

[Mayr and Meyer '82]



Louis Rosier

Hsu-Chun Yen

Refined via a multiparameter analysis.

[Rosier and Yen '85]

d is the dimension: number of components.
n is the size: number of states plus the absolute values of all updates (unary encoding).

Vector Addition Systems with(out) States

d-VASS

$$(\, \boldsymbol{Q},\ \boldsymbol{T}\,)$$

$\boldsymbol{Q}$ is a finite set of states.

$\boldsymbol{T} \subseteq \boldsymbol{Q} \times \mathbb{Z}^{\boldsymbol{d}} \times \boldsymbol{Q}$ are the transitions.

Configurations are in $\boldsymbol{Q} \times \mathbb{N}^{\boldsymbol{d}}$.

d-VAS

$$(\, \boldsymbol{V}\,)$$

$\boldsymbol{V} \subseteq \mathbb{Z}^{\boldsymbol{d}}$ is just a set of vectors.

Configurations are in $\mathbb{N}^{\boldsymbol{d}}$.

John Hopcroft

Jean-Jacques Pansiot

Lemma: A $\boldsymbol{d}$-VASS can be *simulated* by a $(\boldsymbol{d}+\boldsymbol{3})$-VAS. [Hopcroft and Pansiot '79]

Idea: maintain invariants containing information about the number of states and the current state on three dedicated additional counters.

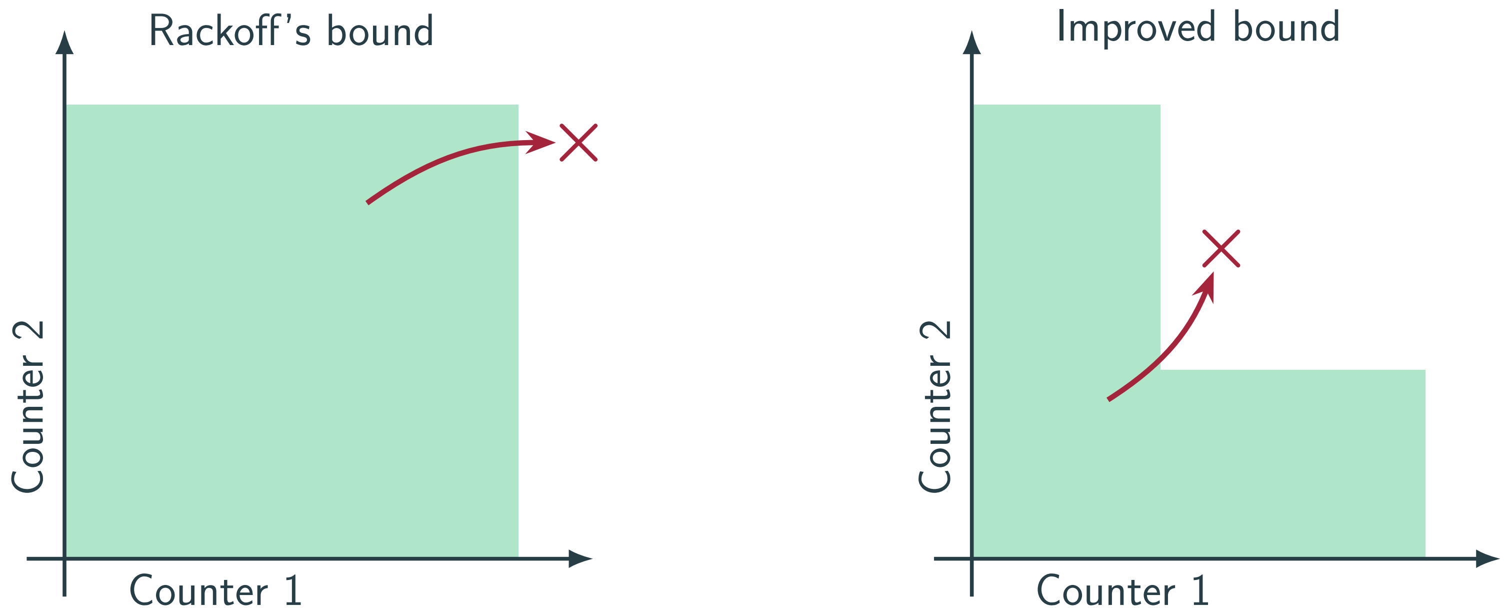
Takeaway: we will work with VAS because we do not fix the dimension.

Improving Rackoff's Upper Bound

Theorem: Coverability in VASS is always witnessed by $n^{2^{\mathcal{O}(d)}}$ length runs.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Idea: Carefully use Rackoff's bounding technique with sharper counter value bounds.

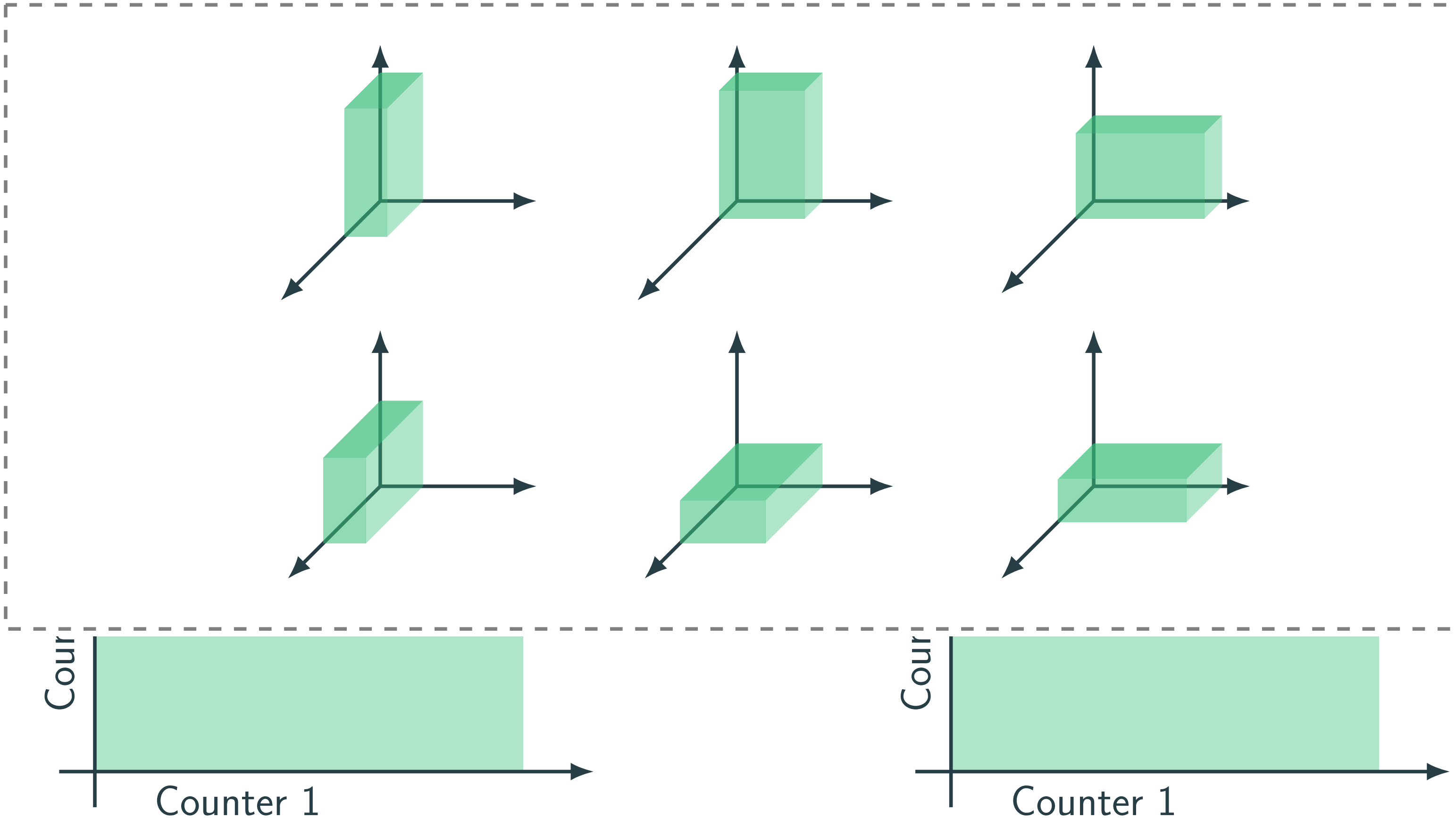


Improving Rackoff's Upper Bound

Theor

grzycki '23]

Idea:

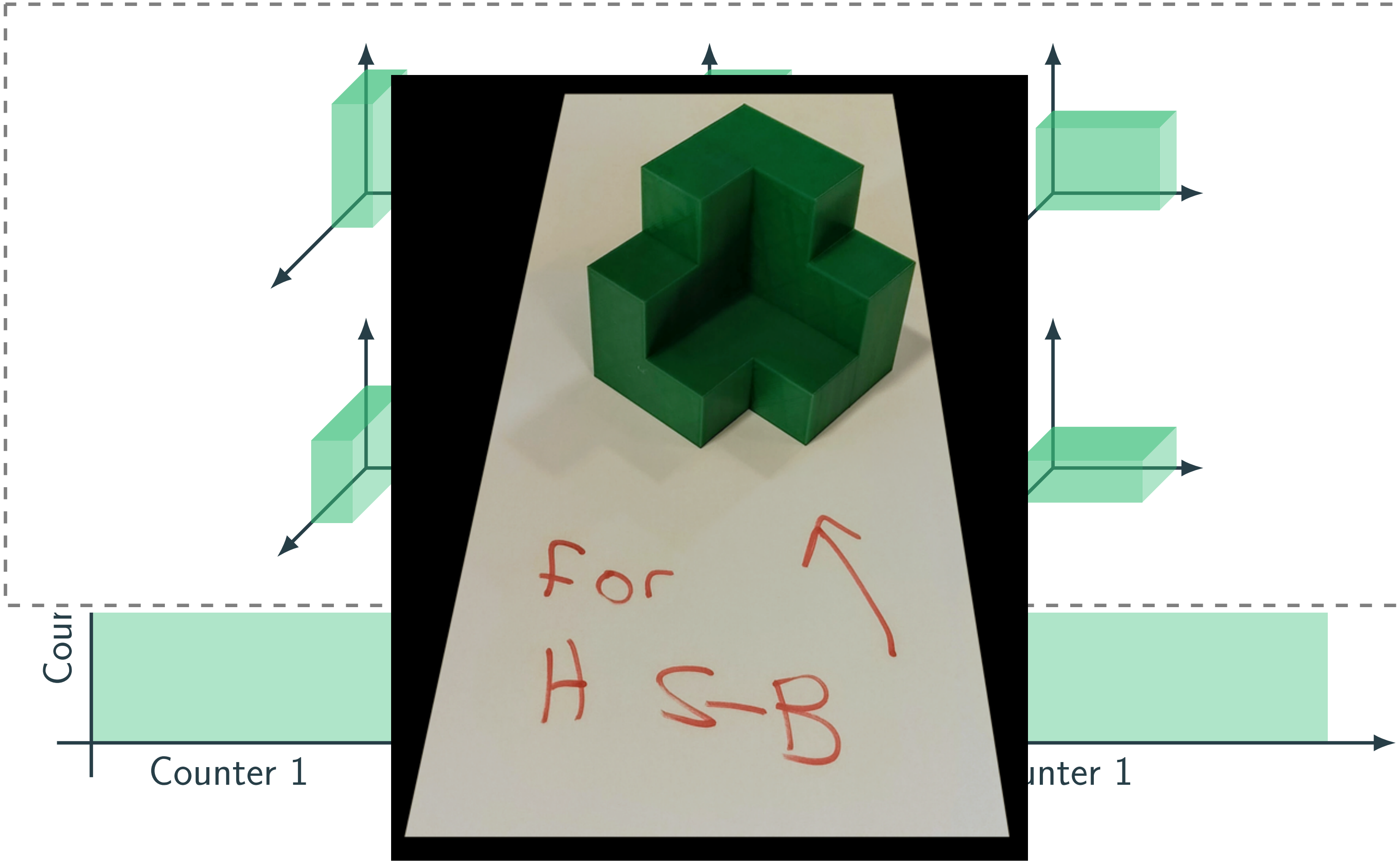


Improving Rackoff's Upper Bound

Theor

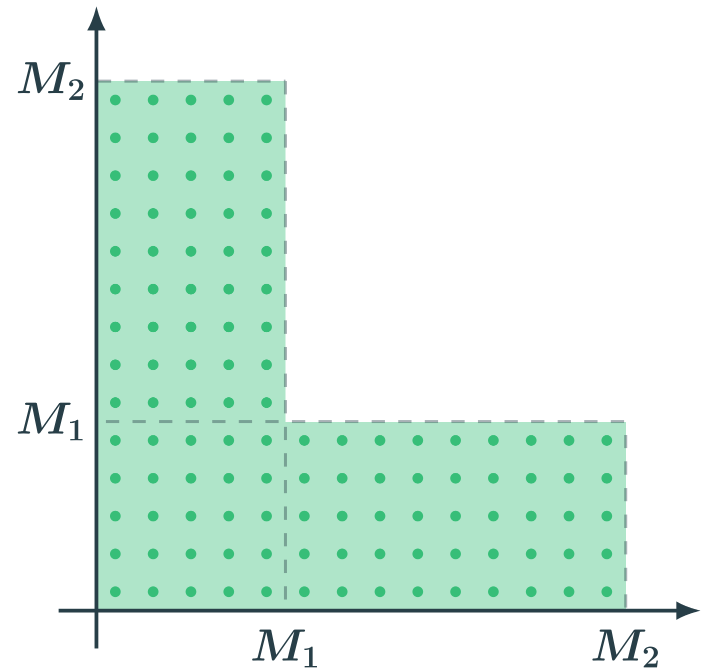
grzycki '23]

Idea: (



Thin Configurations

Definition: A configuration $\vec{v} \in \mathbb{N}^d$ is *thin* if, after sorting the components, $\vec{v}[1] < M_1$, $\vec{v}[2] < M_2$, ..., $\vec{v}[d] < M_d$.



Importantly, to get an improvement over Rackoff's bound:

$M_1 << M_2 << \ldots << M_d$.

Precisely,

$$M_1 = n \cdot n^{4^0}, M_2 = n \cdot n^{4^1}, \ldots, M_d = n \cdot n^{4^{d-1}}.$$

How many thin configurations exist?

$$\leq d! \cdot M_1 \cdot M_2 \cdot \ldots \cdot M_d = d! \cdot (n \cdot n^{4^0}) \cdot (n \cdot n^{4^1}) \cdot \ldots \cdot (n \cdot n^{4^{d-1}}).$$

$$= d! \cdot n^d \cdot n^{\sum_{i=0}^{d-1} 4^i}.$$

Bounding the Length of Coverability Runs

Consider the shortest coverability run $\vec{u} \xrightarrow{\pi} \vec{w}$, where $\vec{w} \geq \vec{v}$.

Split π at first "non-thin" configuration: $\vec{u} \xrightarrow{\rho} \vec{x} \xrightarrow{\tau} \vec{w}$.

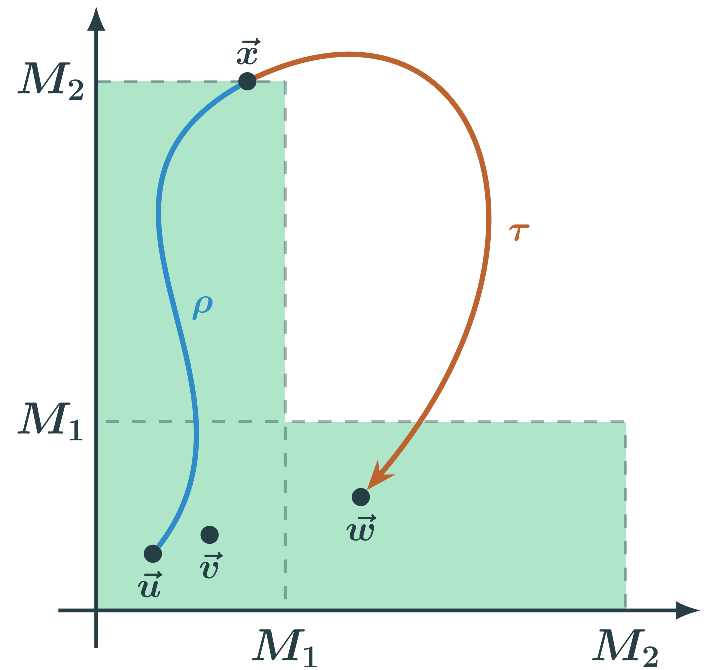
ρ is the *thin part* of the run, its length is bounded by the number of thin configurations.

Claim 1: $\mathit{len}(\rho) \leq d! \cdot n^d \cdot n^{\sum_{i=0}^{d-1} 4^i}$.

Proof idea: there cannot be any zero effect cycles in π.

τ is the *tail* of the run, at least one component had a large value at $\vec{x}$, so can then be 'ignored'.

Claim 2: $\mathit{len}(\tau) \leq n^{4^{d-1}}$.



Using Rackoff's Inductive Technique

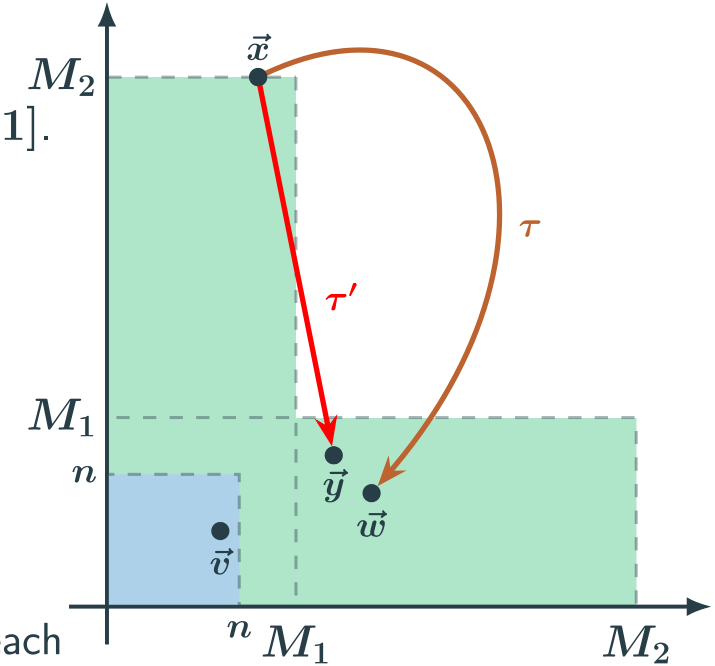
Claim 2: $\mathit{len}(\tau) \leq n^{4^{d-1}}$. **(Proof by induction on d)**

Sort the components $\vec{x}[1] \leq \vec{x}[2] \leq \ldots \leq \vec{x}[d]$.

There exists $i \in \{0, \ldots, d-1\}$ such that $M_{i+1} \leq \vec{x}[i+1]$.

Moreover, $M_{i+1} = n \cdot n^{4^i} \leq \vec{x}[i+1] \leq \ldots \leq \vec{x}[d]$.

Example: $\vec{x}[1] < M_1$ but $\vec{x}[2] \geq M_2$.



Use induction, focussing just on the first i components.

There is an alternative suffix τ' with $\mathit{len}(\tau') \leq n^{4^i}$ and

$(x[1], \ldots, x[i]) \xrightarrow{\tau'} (\vec{y}[1], \ldots, \vec{y}[i]) \geq (\vec{v}[1], \ldots, \vec{v}[i])$.

We know that τ' has at least $-n \cdot (\mathit{len}(\tau') - 1)$ effect on each of the remaining components. Fortunately, $(n \cdot n^{4^i}, \ldots, n \cdot n^{4^i}) \leq (\vec{x}[i+1], \ldots, \vec{x}[d])$.

So, $(\vec{x}[i+1], \ldots, \vec{x}[d]) \xrightarrow{\tau'} (\vec{y}[i+1], \ldots, \vec{y}[d]) \geq (n, \ldots, n) \geq (\vec{v}[i+1], \ldots, \vec{v}[d])$. □

Proof of Main Theorem

Theorem: Coverability in VASS is always witnessed by $n^{2^{\mathcal{O}(d)}}$ length runs.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Proof: Let π be the shortest run witnessing coverability.

$$
\begin{aligned}
\mathit{len}(\pi) &= \mathit{len}(\rho) + \mathit{len}(\tau) \\
&\leq d! \cdot n^d \cdot n^{\sum_{i=0}^{d-1} 4^i} + n^{4^{d-1}} && \text{(By **Claim 1** and **Claim 2**)} \\
&\leq 2 \cdot d! \cdot n^d \cdot n^{\sum_{i=0}^{d-1} 4^i} \\
&\leq n^{2^d} \cdot n^{\sum_{i=0}^{d-1} 4^i} && \text{(when } n \geq 2, \quad 2 \cdot d! \cdot n^d \leq n^{2^d} \text{)} \\
&\leq n^{4^d} && \text{(when } d \geq 1, \quad 2^d + \textstyle\sum_{i=0}^{d-1} 4^i \leq 4^d \text{)} \\
&= n^{2^{2d}} = n^{2^{\mathcal{O}(d)}}.
\end{aligned}
$$

□

Algorithms for Coverability

Theorem: Coverability in VASS is always witnessed by $n^{2^{\mathcal{O}(d)}}$ length runs.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Corollary 1: Coverability in VASS can be decided in $2^{\mathcal{O}(d)} \cdot \log(n)$ space. **OPTIMAL!**

Proof idea: Nondeterministically search through the configuration space, each configuration can be expressed with $2^{\mathcal{O}(d)} \cdot \log(n)$ bits.

Corollary 2: Coverability in VASS can be decided in $n^{2^{\mathcal{O}(d)}}$ time. **CONDITIONALLY OPTIMAL!**

Proof idea: Deterministically search through the configuration space.

Conditionally Optimal Time Bound

Corollary 2: Coverability in VASS can be decided in $n^{2^{\mathcal{O}(d)}}$ time.

Theorem: Assuming the Exponential Time Hypothesis, coverability in VASS requires $n^{2^{\Omega(d)}}$ time.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Idea: Reduce detecting a 2^d-clique in a 2^d-partite n-vertex directed graph to coverability.

Conjecture (Exponential Time Hypothesis): 3-SAT with k-variables requires $2^{\Omega(k)}$ time.

$$\Downarrow$$

Detecting whether there is a k-clique in a k-partite n-vertex graph requires $n^{\Omega(k)}$ time.

[Chen, Chor, Fellows, Huang, Juedes, Kanj, and Xia '05]

[Chen, Huang, Kanj, and Xia '06]

[Cygan, Fomin, Kowalik, Lokshtanov, Marx, Ma. Pilipczuk, and Mi. Pilipczuk '15]

Bounded Two-Counter Machines

Idea: Reduce detecting a 2^d-clique in a 2^d-partite n-vertex directed graph to coverability.

First, reduce to coverability in a $n^{2^{\mathcal{O}(d)}}$-bounded two-counter machine.

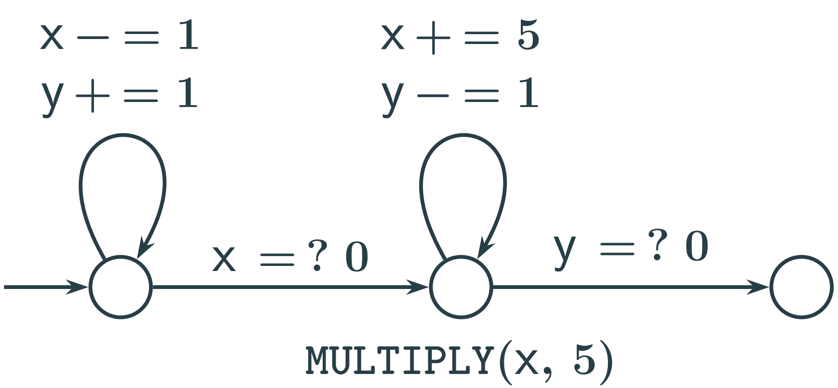
Then, simulate a $n^{2^{\mathcal{O}(d)}}$-bounded two-counter machine using an $\mathcal{O}(n)$-state $\mathcal{O}(d)$-VASS.

An **$n^{2^{\mathcal{O}(d)}}$-bounded two-counter machine** has two counters $\mathsf{x}, \mathsf{y} \in \{0, 1, \ldots, n^{2^{\mathcal{O}(d)}}\}$ that can be added to ($\mathsf{x} \mathrel{+}= 2$), subtracted from ($\mathsf{y} \mathrel{-}= 3$), and zero-tested ($\mathsf{x} =?\ 0$).

Pre: $\mathsf{x} = x,\ \mathsf{y} = 0$

1. $\texttt{LOOP}\,(\mathsf{x} \mathrel{-}= 1, \mathsf{y} \mathrel{+}= 1)$
2. $\mathsf{x} =?\ 0$
3. $\texttt{LOOP}\,(\mathsf{x} \mathrel{+}= 5, \mathsf{y} \mathrel{-}= 1)$
4. $\mathsf{y} =?\ 0$

Post: $\mathsf{x} = x \cdot 5,\ \mathsf{y} = 0$



Bounded Two-Counter Machines

Idea: Reduce detecting a 2^d-clique in a 2^d-partite n-vertex directed graph to coverability.

First, reduce to coverability in a $n^{2^{\mathcal{O}(d)}}$-bounded two-counter machine.

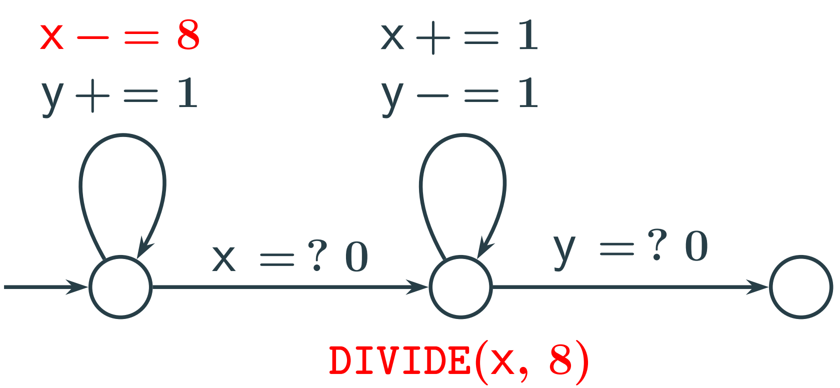
Then, simulate a $n^{2^{\mathcal{O}(d)}}$-bounded two-counter machine using an $\mathcal{O}(n)$-state $\mathcal{O}(d)$-VASS.

An **$n^{2^{\mathcal{O}(d)}}$-bounded two-counter machine** has two counters $\mathsf{x}, \mathsf{y} \in \{0, 1, \ldots, n^{2^{\mathcal{O}(d)}}\}$ that can be added to ($\mathsf{x} \mathrel{+}= 2$), subtracted from ($\mathsf{y} \mathrel{-}= 3$), and zero-tested ($\mathsf{x} =?\, 0$).

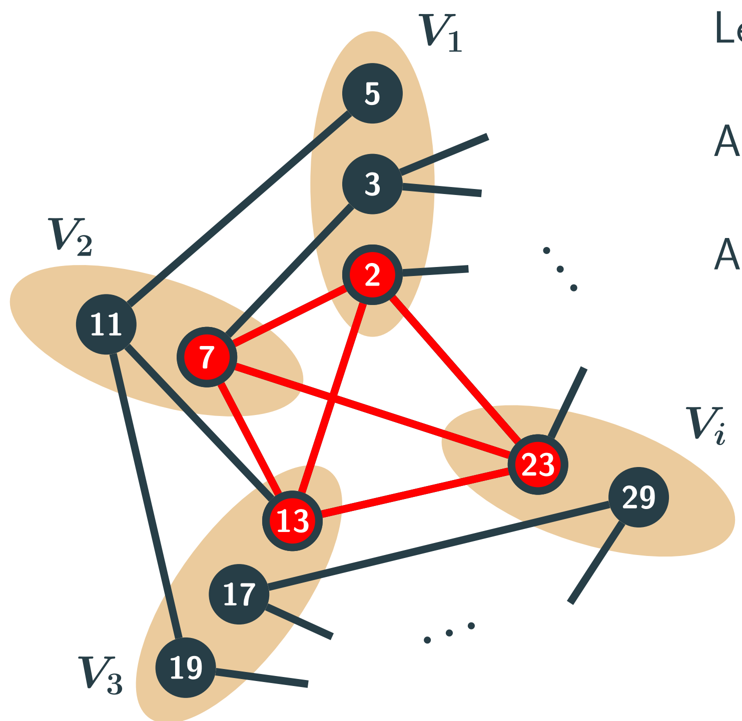
Pre: $\mathsf{x} = x$, $\mathsf{y} = 0$

1. $\mathtt{LOOP}\,(\mathsf{x} \mathrel{-}= 8, \mathsf{y} \mathrel{+}= 1)$
2. $\mathsf{x} =?\, 0$
3. $\mathtt{LOOP}\,(\mathsf{x} \mathrel{+}= 1, \mathsf{y} \mathrel{-}= 1)$
4. $\mathsf{y} =?\, 0$

Post: $\mathsf{x} = x \div 8$, $\mathsf{y} = 0$



Detecting Cliques using Divisibility Tests



Let $(V_1 \cup V_2 \cup \cdots \cup V_k, E)$ be a k-partite n-vertex graph.

Associate the first n primes with the verticies.

A candidate k-clique is represented by a product of k primes.

Example: $c = 2 \cdot 7 \cdot 13 \cdot \ldots \cdot 23$.

To check if v represents a clique, use divisibility tests to verify all nodes are adjacent.

Example: $(2 \cdot 7) | c\,?$ $(2 \cdot 13) | c\,?$ $(7 \cdot 13) | c\,?$ $\ldots$ $(2 \cdot 23) | c\,?$ $(7 \cdot 23) | c\,?$ $(13 \cdot 23) | c\,?$

There exist $p_1 \in \text{Primes}(V_1), \ldots, p_k \in \text{Primes}(V_k)$ such that for every pair $1 \leq i < j \leq k$, there is an edge $\{p, q\} \in (V_i \times V_j) \cap E$ such that $(p \cdot q) | \, p_1 \cdot \ldots \cdot p_k$ $\iff$ there exists a k-clique.

Bounded Two-Counter Machine Implementation

There e… k, there

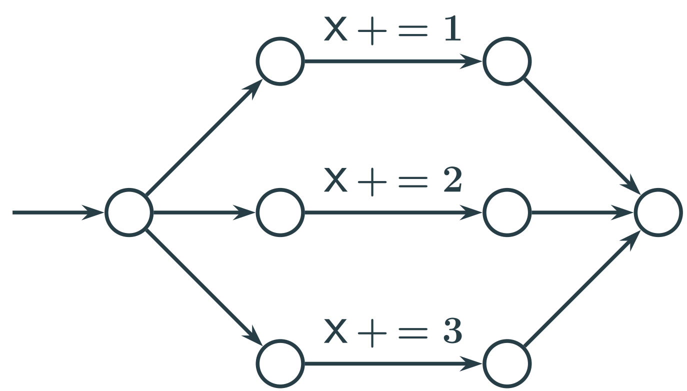
is an ec… k-clique.

Guessing with Nondeterministic Branching

Pre: $\mathsf{x} = x$

1. $\texttt{GUESS:}$ $c \in \{1, 2, 3\}$
2. $\mathsf{x} \mathrel{+}= c$

Post: $\mathsf{x} = x + 1$, or
$\mathsf{x} = x + 2$, or
$\mathsf{x} = x + 3$.



Bounded Two-Counter Machine Implementation

There exist $p_1 \in \text{Primes}(V_1), \ldots, p_k \in \text{Primes}(V_k)$ such that for every pair $1 \leq i < j \leq k$, there is an edge $\{p, q\} \in (V_i \times V_j) \cap E$ such that $(p \cdot q) \mid p_1 \cdot \ldots \cdot p_k \iff$ there exists a k-clique.

Part one: Guess a candidate clique.

Pre: $\mathsf{x} = 1$, $\mathsf{y} = 0$.

1. **GUESS**: $p_1 \in \text{Primes}(V_1)$
2. **MULTIPLY**(x, p_1)

$\vdots$

2k−1. **GUESS**: $p_k \in \text{Primes}(V_k)$

2k. **MULTIPLY**(x, p_k)

Post: $\mathsf{x} = p_1 \cdot \ldots \cdot p_k$, $\mathsf{y} = 0$.

This two-counter program terminates

$\iff$ there exists a k-clique.

Part two: Check the candidate is a clique.

Pre: $\mathsf{x} = p_1 \cdot \ldots \cdot p_k$, $\mathsf{y} = 0$.

1. **GUESS**: $\{p_1, p_2\} \in (V_1 \times V_2) \cap E$
2. **DIVIDE**(x, $p_1 \cdot p_2$)
3. **MULTIPLY**(x, $p_1 \cdot p_2$)

$\vdots$

$<3k^2$. **GUESS**: $\{p_{k-1}, p_k\} \in (V_{k-1} \times V_k) \cap E$

$<3k^2$. **DIVIDE**(x, $p_{k-1} \cdot p_k$)

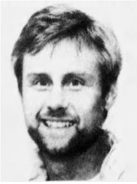
$<3k^2$. **MULTIPLY**(x, $p_{k-1} \cdot p_k$)

Post: $\mathsf{x} = p_1 \cdot \ldots \cdot p_k$, $\mathsf{y} = 0$.

VASS can Simulate Bounded Two-Counter Machines

Counter bound of k-clique detecting two-counter machine: $\mathcal{O}(p_{\max}^k) \le \mathcal{O}(n^k \log(n)^k) \le \mathcal{O}(n^{2k})$.

Size of k-clique detecting two-counter machine: $\mathcal{O}(n^{11}) \le poly(n)$.



Louis Rosier

Hsu-Chun Yen

Lemma: In *poly*(n) time, one can construct a $\mathcal{O}(\log(k))$-VASS that can *simulate* an $\mathcal{O}(n^k)$-bounded $\mathcal{O}(1)$-counter machine of *poly*(n) size.

[Rosier and Yen '85]

If we set $k = 2^d$, the *poly*(n)-size two-counter machine for detecting 2^d-cliques is $\mathcal{O}(n^{2^d})$-bounded.

$\implies$ In *poly*(n) time, one can construct an $\mathcal{O}(d)$-VASS for detecting 2^d-cliques.

Remark: Here, termination is coverability.

"Can I get to the end of the program with any (at least zero) value on each of the counters?"

Reducing to Coverability in VASS

Detecting 2^d-cliques in an n-vertex graph requires $n^{\Omega(2^d)}$ time under the Exponential Time Hypothesis.

↓ Via divisibilty tests of a product of primes encoding.

First, construct an instance of termination in a *poly*(n)-size $\mathcal{O}(n^{2^d})$-bounded two-counter machine.

↓ Using Rosier and Yen's simulation lemma.

Then, in *poly*(n) time, construct an instance of coverability in an $\mathcal{O}(d)$-VASS.

Theorem: Assuming the Exponential Time Hypothesis, coverability in VASS requires $n^{2^{\Omega(d)}}$ time.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Coverability in VASS Revisited: Improving Rackoff's Bound to Obtain Conditional Optimality

Theorem: Coverability in VASS is always witnessed by $n^{2^{\mathcal{O}(d)}}$ length runs.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Corollary 1: Coverability in VASS can be decided in $2^{\mathcal{O}(d)} \cdot \log(n)$ space. **OPTIMAL!**

Corollary 2: Coverability in VASS can be decided in $n^{2^{\mathcal{O}(d)}}$ time. **CONDITIONALLY OPTIMAL!**

Theorem: Assuming the Exponential Time Hypothesis, coverability in VASS requires $n^{2^{\Omega(d)}}$ time.

[Künnemann, Mazowiecki, Schütze, S-B, and Węgrzycki '23]

Thank You!

Presented by Henry Sinclair-Banks, University of Warwick, UK

Formal Methods Seminar (M2F) in LaBRI, Bordeaux, France

European Research Council
Established by the European Commission