New Lower Bounds for Reachability in Vector Addition Systems

Wojciech Czerwiński
Ismaël Jecker
Sławomir Lasota
Jérôme Leroux
Łukasz Orlikowski

Presentation plan

- History and definition of the problem

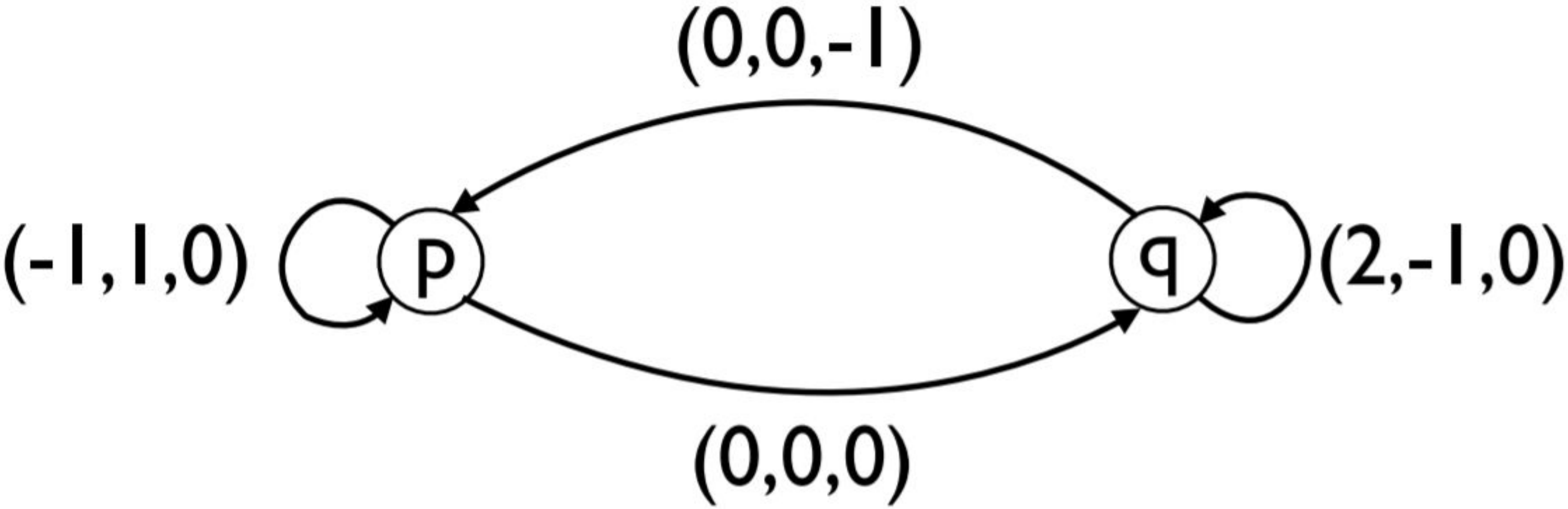
Presentation plan

- History and definition of the problem
- Main result

Presentation plan

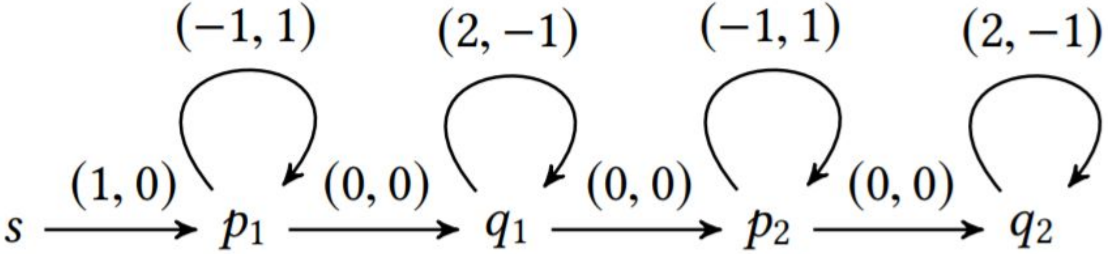
- History and definition of the problem
- Main result
- Techniques

Vector Addition Systems with States



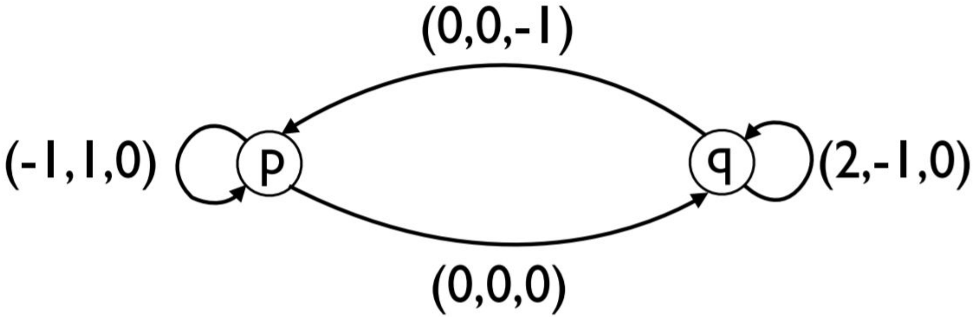
Vector Addition Systems with States

```
x += 1
loop
    x −= 1    y += 1
loop
    x += 2    y −= 1
loop
    x −= 1    y += 1
loop
    x += 2    y −= 1
```



Pushdown Vector Addition Systems with States

VASS extended with stack



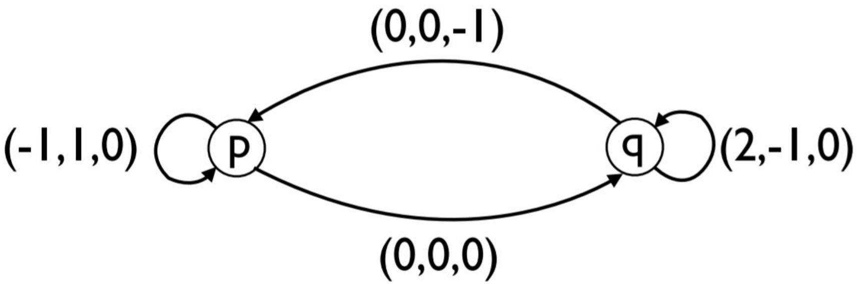
Reachability Problem

Given: a Vector Addition System with States (VASS) V, two configurations s and t

Reachability Problem

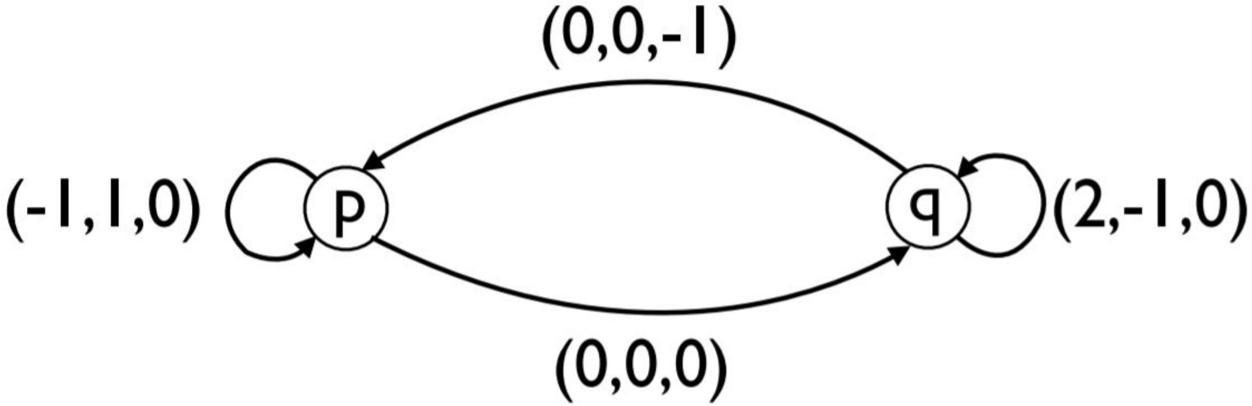
Given: a Vector Addition System with States (VASS) V, two configurations s and t

Question: is there a run from s to t in V?



Example of run

$$p(1,0,2) \to p(0,1,2) \to q(0,1,2) \to q(2,0,2) \to p(2,0,1)$$



Fast growing functions

$$F_1(n) = 2n$$

Fast growing functions

$F_1(n) = 2n$ $F_2(n) = 2^{n-1}$

Fast growing functions

$F_1(n) = 2n$ $\qquad$ $F_2(n) = 2^{n-1}$

$F_d(n) = F_{d-1}(F_{d-1}(\ldots(F_{d-1}(1))\ldots))$ composed n-1 times

Short History

- Lipton `76: ExpSpace-hardness

Short History

- Lipton `76: ExpSpace-hardness
- Mayr `81: decidability

Short History

- Lipton `76: ExpSpace-hardness
- Mayr `81: decidability
- Kosaraju `82, Lambert `92: simplifications

Short History

- Lipton `76: ExpSpace-hardness
- Mayr `81: decidability
- Kosaraju `82, Lambert `92: simplifications
- Leroux, Schmitz `15: cubic-Ackermann upper bound

Short History

- Lipton `76: ExpSpace-hardness
- Mayr `81: decidability
- Kosaraju `82, Lambert `92: simplifications
- Leroux, Schmitz `15: cubic-Ackermann upper bound
- Leroux, Schmitz `19: Ackermann upper bound

Short History

- Lipton `76: ExpSpace-hardness
- Mayr `81: decidability
- Kosaraju `82, Lambert `92: simplifications
- Leroux, Schmitz `15: cubic-Ackermann upper bound
- Leroux, Schmitz `19: Ackermann upper bound
- Czerwiński, Lasota, Lazic, Leroux, Mazowiecki `19: Tower-hardness

Short History

- Leroux and Czerwiński, Orlikowski `21: Two independent proofs of Ackermann-hardness (F_d-hardness in dimension 4d+5 and 6d)

Short History

- Leroux and Czerwiński, Orlikowski `21: Two independent proofs of Ackermann-hardness (F_d-hardness in dimension 4d+5 and 6d)
- Lasota follow-up with simpler proof (3d + 2)

Short History

- Leroux and Czerwiński, Orlikowski `21: Two independent proofs of Ackermann-hardness (F_d-hardness in dimension 4d+5 and 6d)
- Lasota follow-up with simpler proof (3d + 2)
- For pushdown decidability not known

Our contribution

Theorem:

The Reachability Problem for Vector Addition Systems is F_d-hard in dimension 2d+3

Our contribution

Theorem:

The Reachability Problem for Vector Addition Systems is F_d-hard in dimension 2d+3

Theorem:

The Reachability Problem for Pushdown VASSes is F_d-hard in dimension $\frac{d}{2} + 6$.

Our contribution

Theorem:

The Reachability Problem for Vector Addition Systems is F_d-hard in dimension 2d+3

Theorem:

The Reachability Problem for Pushdown VASSes is F_d-hard in dimension $\frac{d}{2} + 6$. First lower bound not inherited from VASS!

Techniques

Minsky machine

- counter program with zero-tests and two counters

Minsky machine

- counter program with zero-tests and two counters
- in general reachability problem is undecidable

F_d-zero test reachability problem

Given: a Minsky machine M and two configurations s and t

F_d-zero test reachability problem

Given: a Minsky machine M and two configurations s and t

Question: is there a run from s to t in M which does exactly $F_d(|M|)$ zero tests?

Multiplication triples technique

- Let’s have a triple (b,2c, 2bc)

Multiplication triples technique

- Let’s have a triple (b,2c, 2bc)
- Allows for c zero-tests on b-bounded counters

Multiplication triples technique

flush(x,y,z):

loop

$$x \mathrel{-}= 1 \qquad y \mathrel{+}= 1 \qquad z \mathrel{-}= 1$$

Multiplication triples technique

Let’s have triple (b, y, z) = (B, 2C, 2BC)

Zero-test(x):

flush(b, x, z)
flush(x, b, z)
$y \mathrel{-}= 2$

Multiplication triples technique

Let's have triple (b, y, z) = (B, 2C, 2BC)

Zero-test(x):

1: **flush**(b, x, z)
2: **flush**(x, b, z)
3: $y \mathrel{-}= 2$

Invariant (b+x)y=z kept only if x was indeed zero!

New Multiplication triples technique

- Let's have a triple $(a,b,(4^b-1)a)$

New Multiplication triples technique

- Let’s have a triple $(a,b, (4^b-1)a)$
- Allows for b zero-tests on a-bounded counters

New Multiplication triples technique

Let’s have triple (a, b, c) = (a,b, $(4^b-1)a$)

Program $\mathcal{Z}ero$(x):

```
loop  a ⟶ t   c ⟶ t
loop  y ⟶ x   c ⟶ t
loop  t ⟶ a   c ⟶ a
loop  x ⟶ y   c ⟶ a
b −= 1
```

New Multiplication triples technique

Let's have triple (a, b, c) = (a,b, $(4^b$-1)a)

Program $\mathit{Zero}(x)$**:**

loop $a \longrightarrow t \quad c \longrightarrow t$
loop $y \longrightarrow x \quad c \longrightarrow t$
loop $t \longrightarrow a \quad c \longrightarrow a$
loop $x \longrightarrow y \quad c \longrightarrow a$
$b \mathrel{-}= 1$

Invariant $(a+x+y+t)(4^b-1)=c$ kept only if x was indeed zero!

New Multiplication triples technique

Let's have triple (a, b, c) = (a,b, (4^b-1)a)

Program $\mathcal{Z}ero(\mathrm{x})$:

```
loop a ⟶ t   c ⟶ t
loop y ⟶ x   c ⟶ t
loop t ⟶ a   c ⟶ a
loop x ⟶ y   c ⟶ a
b −= 1
```

Invariant (a+x+y+t)(4^b-1)=c kept only if x was indeed zero!

You can see this invariant as (a+x+y+t)4^b=c+a+x+y+t

- Main challenge: producing such triples

- Main challenge: producing such triples
- Can be done by producing bigger triples from smaller ones

- Main challenge: producing such triples
- Can be done by producing bigger triples from smaller ones
- We can produce $(M, F_d(M), (4^{F_d(M)} - 1)M)$ triple using 2d+4 counters in VASS and $\frac{d}{2} + 4$ in PVASS

F_d-hard VASS/PVASS

- d nested levels of counters

b_1	c_1
b_2	c_2
$\vdots$	$\vdots$
b_d	c_d

F_d-hard VASS/PVASS

- d nested levels of counters
- new triple technique allows for one common a for all levels

b_1	c_1
b_2	c_2
. . .	. . .
b_d	c_d

F_d-hard VASS/PVASS

- d nested levels of counters
- new triple technique allows for one common a for all levels
- $b_1, b_2, \ldots, b_d$ have stack structure

b_1	c_1
b_2	c_2
. . .	. . .
b_d	c_d

F_d-hard VASS/PVASS

- d nested levels of counters
- new triple technique allows for one common a for all levels
- $b_1, b_2, \ldots, b_d$ have stack structure
- odd/even c counters have also stack structure

b_1	c_1
b_2	c_2
. . .	. . .
b_d	c_d

F_d-hard VASS/PVASS

- d nested levels of counters
- new triple technique allows for one common a for all levels
- $b_1, b_2, \ldots, b_d$ have stack structure
- odd/even c counters have also stack structure
- we can store all b counters and half c counters on stack

b_1	c_1
b_2	c_2
$\vdots$	$\vdots$
b_d	c_d

Thank you!