FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

for(syte)

The Polynomial Complexity of Vector Addition Systems with States

Florian Zuleger
TU Wien

Alpine Verification Meeting
9.9.2019

Vector Addition Systems (with States)

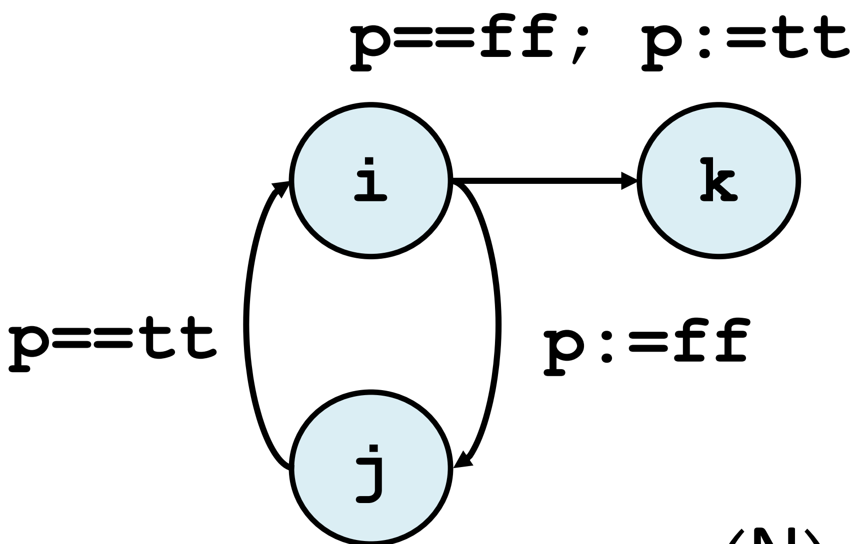
- Vector Addition Systems (VAS) = Petri Nets
- Basic Model for Parellel Processes

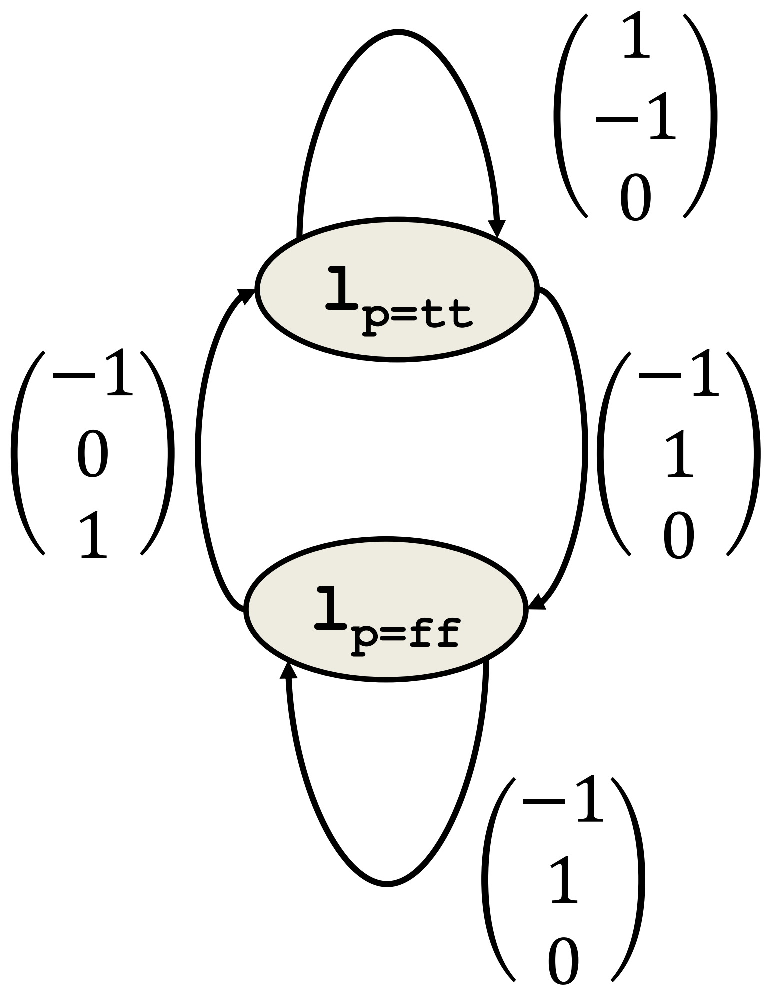
- Vector Addition Systems with States (VASS) =

  VAS + finite control
- Basic model for concurrent systems
- Finite control allows to model communication primitives, such as shared finite memory

Concurrent Systems

Process Template

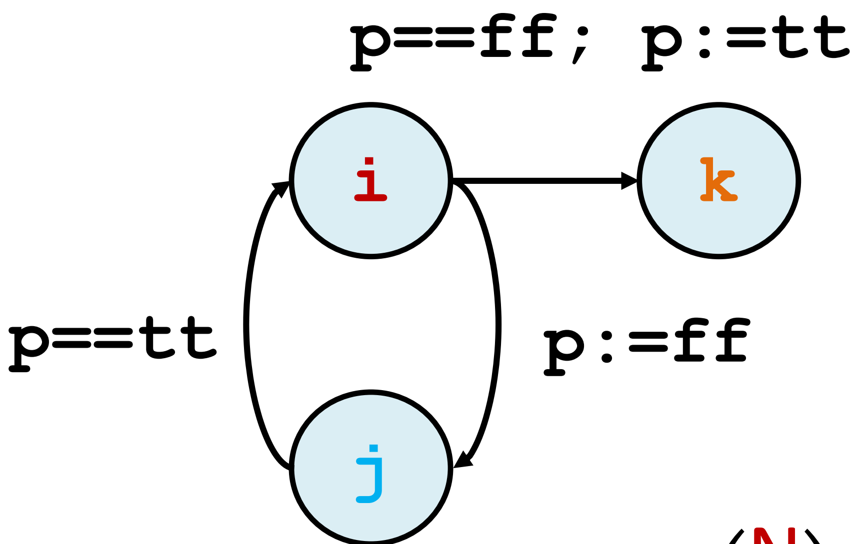


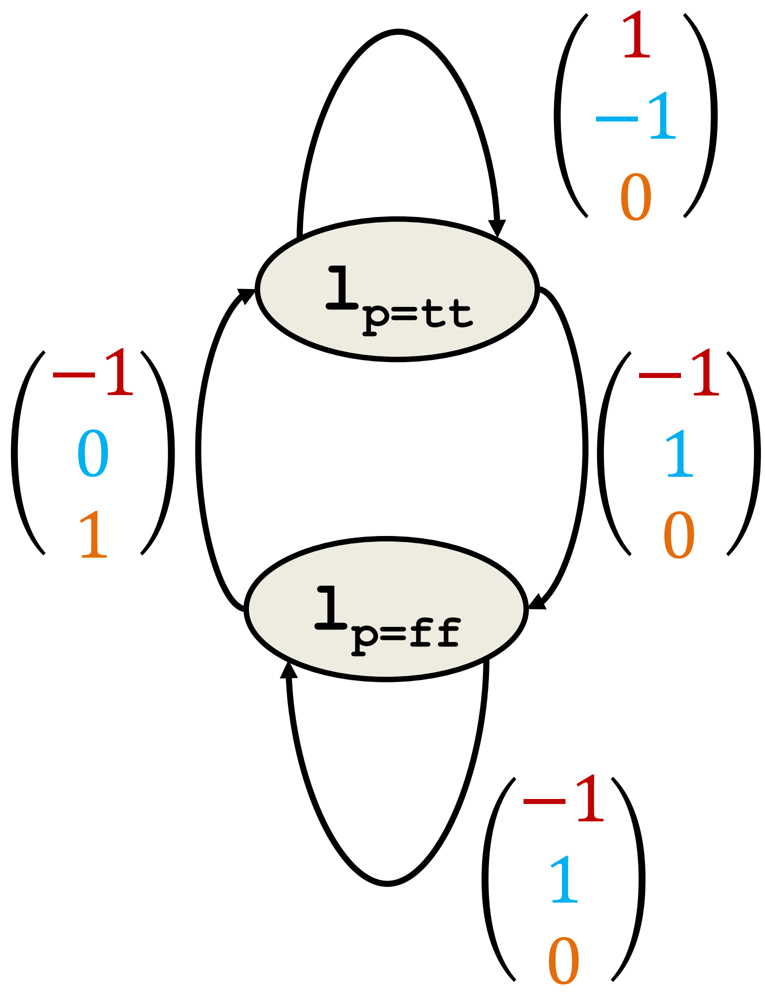


Initial State: $\left(\mathtt{l}_{\mathtt{p=tt}}, \begin{pmatrix} N \\ 0 \\ 0 \end{pmatrix}\right)$

Concurrent Systems

Process Template

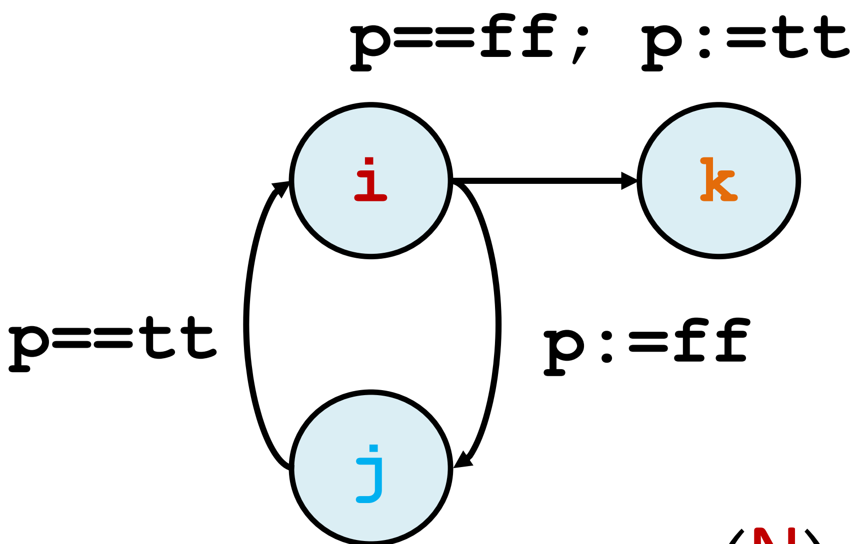


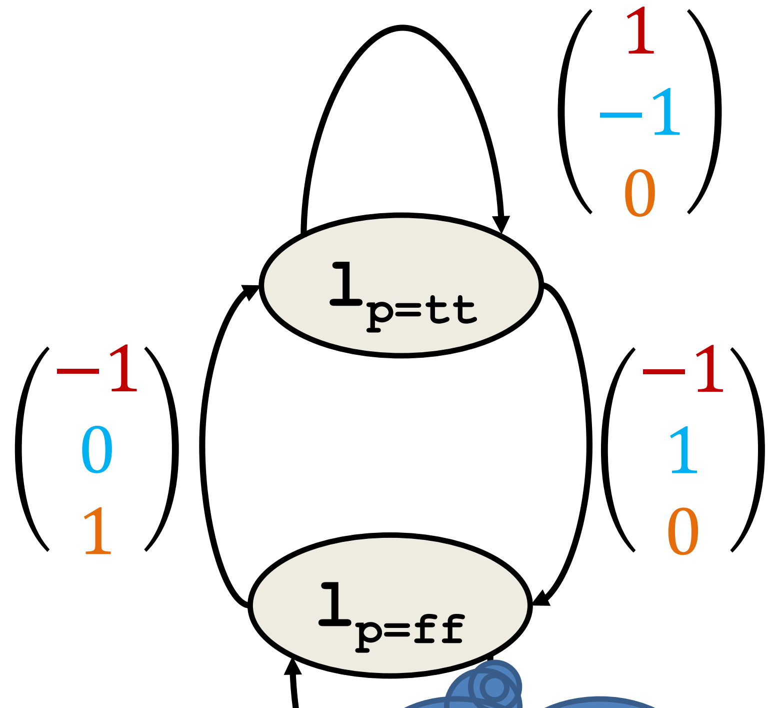


Initial State: $\left(l_{p=tt}, \begin{pmatrix} N \\ 0 \\ 0 \end{pmatrix}\right)$

Concurrent Systems

Process Template



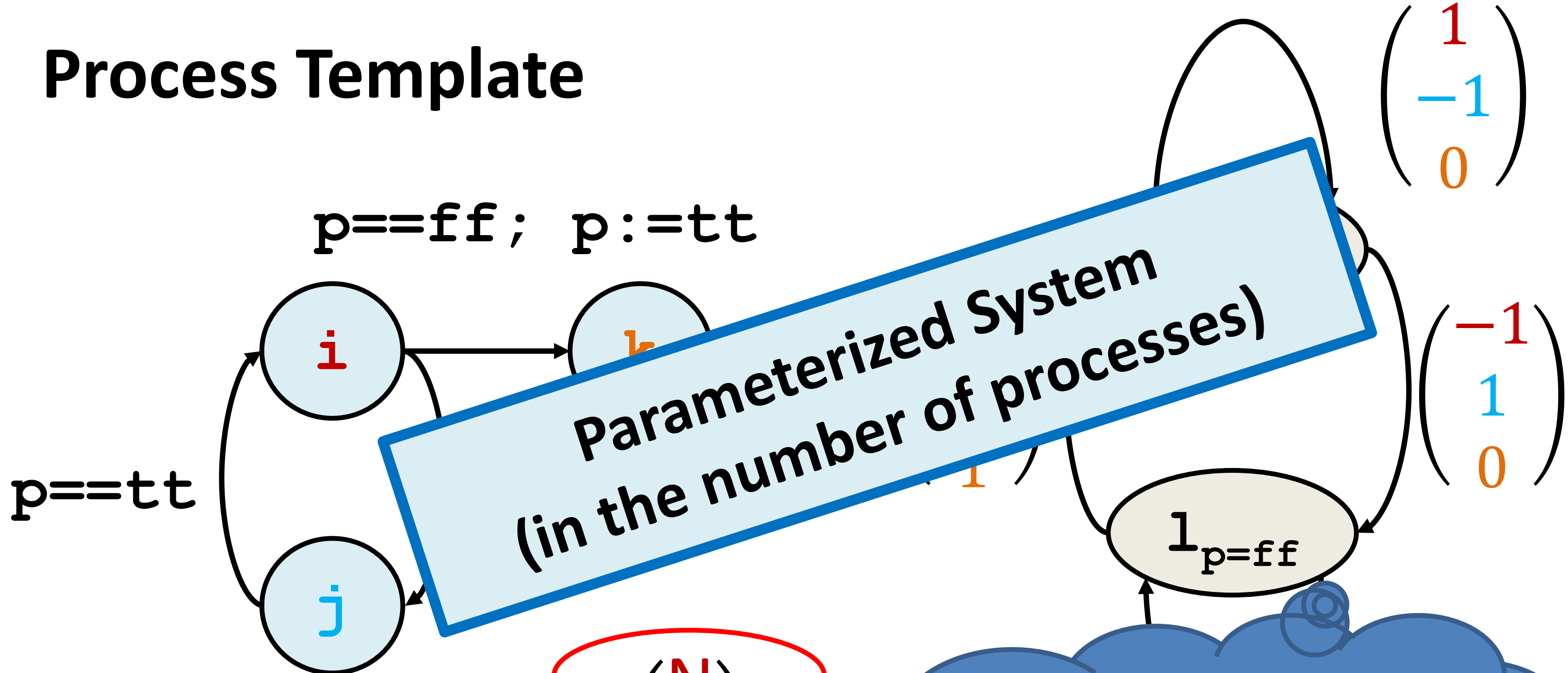


Initial State: $\left(l_{p=tt}, \begin{pmatrix} N \\ 0 \\ 0 \end{pmatrix}\right)$

Boolean p =
Shared Memory =
States of the VASS

Concurrent Systems

Process Template

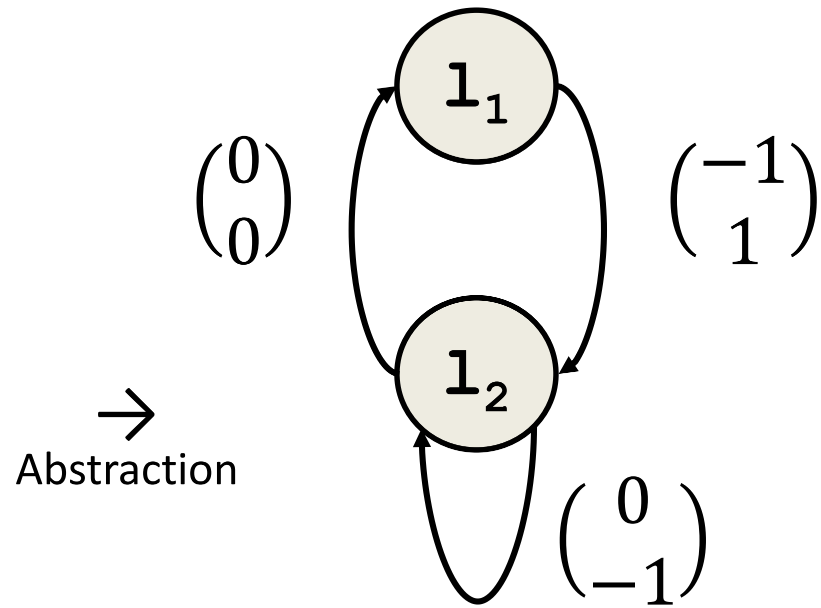


Initial State: $\left(l_{p=tt}, \begin{pmatrix} N \\ 0 \\ 0 \end{pmatrix}\right)$

Boolean p =
Shared Memory =
States of the VASS

Program Analysis

```
void main(uint N){
    uint i=N, j=N;
l1: while(i>0){
     i--;
     j++;
l2:  while(j>0 && *)
        j--;
}
```



Initial State: $(l_1, \binom{N}{N})$

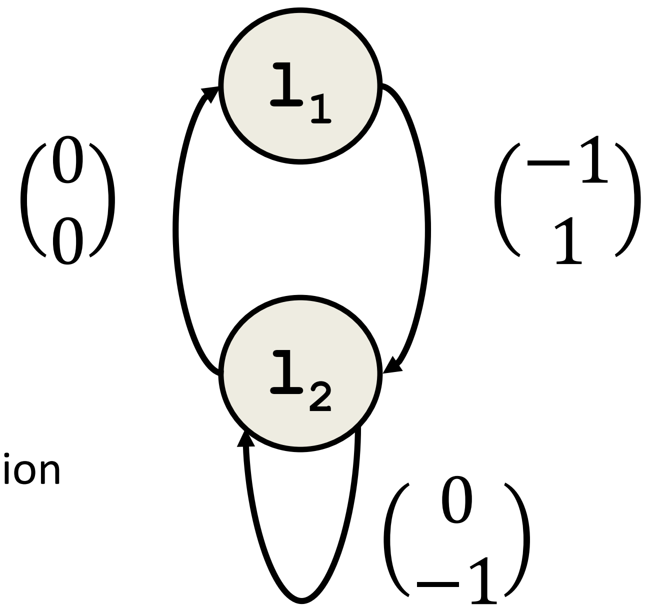
Abstractions to VASSs: CAV'14, FMCAD'15, JAR'17

Program Analysis

```
void main(uint N){
   uint i=N, j=N;
l1: while(i>0){
     i--;
     j++;
l2:  while(j>0 && *)
        j--;
}
```

$\rightarrow$

Abstraction

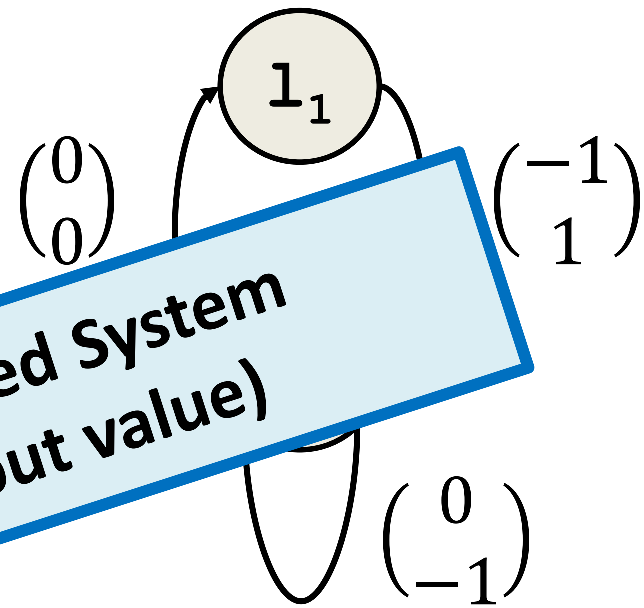


Initial State: $(l_1, \binom{N}{N})$

Abstractions to VASSs: CAV'14, FMCAD'15, JAR'17

Program Analysis

```
void main(uint N){
    uint i=N, j=N;
l1: while(i>0){
     i--;
     j++;
l2:  while(j
        j--;
}
```



Parameterized System (in the input value)

Initial State: $(l_1, \binom{N}{N})$

Abstractions to VASSs: CAV'14, FMCAD'15, JAR'17

Semantics of VASS

Valuation = $\mathbb{N}^{\text{dimension of VASS}}$

Configurations = States x Valuations

Steps = Set of all $(\mathtt{l}_1, v_1) \xrightarrow{u} (\mathtt{l}_2, v_2)$

such that $\mathtt{l}_1 \xrightarrow{u} \mathtt{l}_2$ is a transition,

$v_1 + u \geq 0$ and $v_2 = v_1 + u$.

VASS Termination

Termination (for all initial states):

is there an infinite sequence

$$(\mathtt{l}_1, v_1) \xrightarrow{u_1} (\mathtt{l}_2, v_2) \xrightarrow{u_2} (\mathtt{l}_3, v_3) \ \cdots \ ?$$

Termination (for fixed initial state $(\mathtt{l}_1, v_1)$):

is there an infinite sequence

$$(\mathtt{l}_1, v_1) \xrightarrow{u_1} (\mathtt{l}_2, v_2) \xrightarrow{u_2} (\mathtt{l}_3, v_3) \ \cdots \ ?$$

Classical Results

Termination (for all initial states):

is there an infinite sequence

$$(\mathtt{l}_1, v_1) \xrightarrow{u_1} (\mathtt{l}_2, v_2) \xrightarrow{u_2} (\mathtt{l}_3, v_3) \ \cdots \ ?$$

PTIME (Kosaraju and Sullivan 88')

Termination (for fixed initial state ($\mathtt{l}_1$,v$_1$)):

is there an infinite sequence

$$(\mathtt{l}_1, v_1) \xrightarrow{u_1} (\mathtt{l}_2, v_2) \xrightarrow{u_2} (\mathtt{l}_3, v_3) \ \cdots \ ?$$

EXPSPACE (Lipton 76‘, Rackoff 78‘)

Computational Complexity of VASSs

Termination (for all initial states):

is there an infinite sequence

$$(l_1, v_1) \xrightarrow{u_1} (l_2, v_2) \xrightarrow{u_2} (l_3, v_3) \cdots \ ?$$

Computational Complexity:

Compute a function comp(N) such that the length of the longest sequence

$$(l_1, v_1) \xrightarrow{u_1} (l_2, v_2) \xrightarrow{u_2} (l_3, v_3) \cdots,$$

with $|v_1| = \max_i v_1(i) \leq N$, has comp(N) steps?

Recent Results

Computational Complexity:
comp(N) $\in$ P or comp(N) $\in 2^{\Omega(N)}$

PTIME (Leroux 18)

Computational Complexity:
comp(N) $\in \Theta(N^i)$, for some computable integer $1 \leq i \leq$ dimension of VASS, for some VASS with a "positive normal", i.e., every reachable configuration is linearly bounded in the initial configuration

PTIME (Brázdil, Chatterjee, Kucera, Novotný, Velan, Z 18)

Recent Results

Computational Complexity:
comp(N) ∈ P or comp(N) ∈ $2^{\Omega(N)}$

PTIME (Leroux 18)

Computational Complexity:
comp(N) ∈ $\Theta(N^i)$, for some computable integer $1 \leq i \leq$ dimension of VASS, for some VASS with a "positive normal", i.e., every reachable configuration is linearly bounded in the initial configuration

In this talk we generalize both results

PTIME (Brázdil, Chatterjee, Kucera, Novotný, Velan, Z 18)

Results of this talk

Computational Complexity:

comp(N) $\in \Theta(N^i)$, for some integer $1 \leq i \leq 2^{\text{dimension of VASS}}$, or comp(N) $\in 2^{\Omega(N)}$

PTIME

Results of this talk

Computational Complexity:

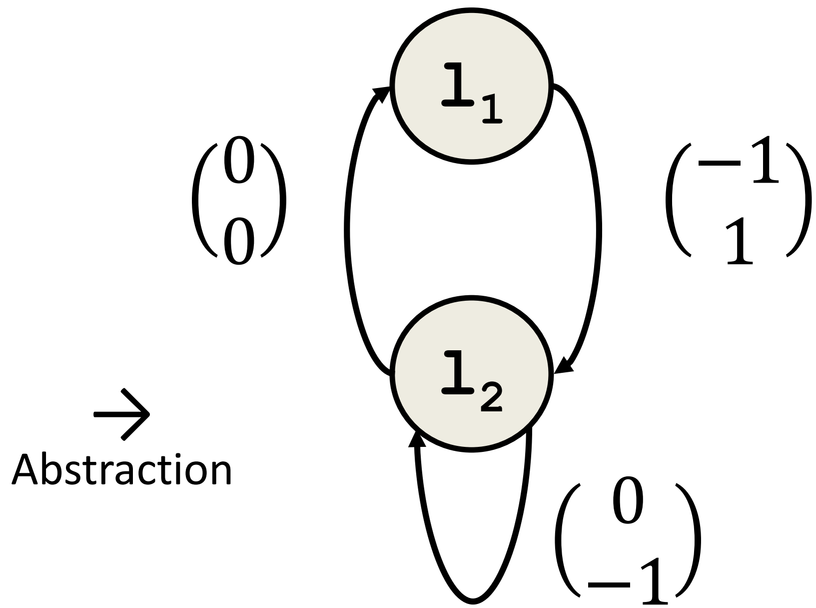
comp(N) ∈ $\Theta(N^i)$, for some integer $1 \leq i \leq 2^{\text{dimension of VASS}}$, or comp(N) ∈ $2^{\Omega(N)}$

PTIME

Full characterization of VASSs with polynomial complexity
(e.g., there is no VASS with complexity N*log(N))

Possible Application: Automated Complexity Analysis

```
void main(uint N){
    uint i=N, j=N;
l1: while(i>0){
     i--;
     j++;
l2:  while(j>0 && *)
        j--;
}
```



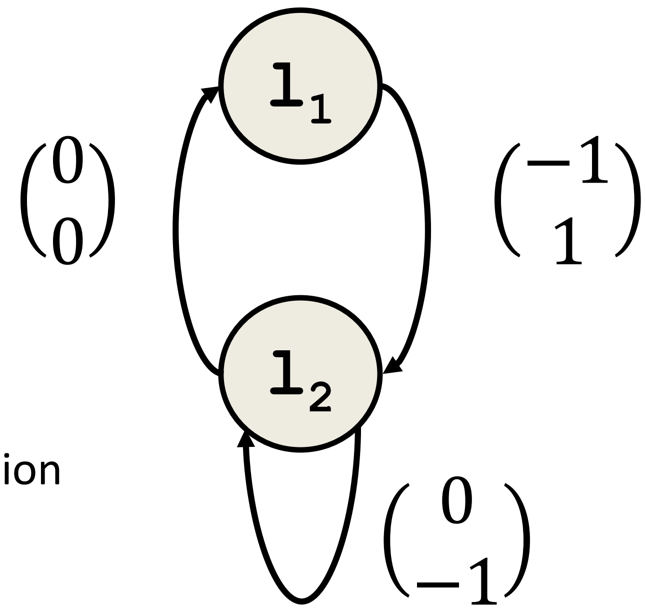
Initial State: $(l_1, \binom{N}{N})$

Complexity: Θ(N)

Possible Application: Automated Complexity Analysis

```
void main(uint N){
    uint i=N, j=N;
l1: while(i>0){
     i--;
     j++;
l2:  while(j>0 && *)
        j--;
}
```

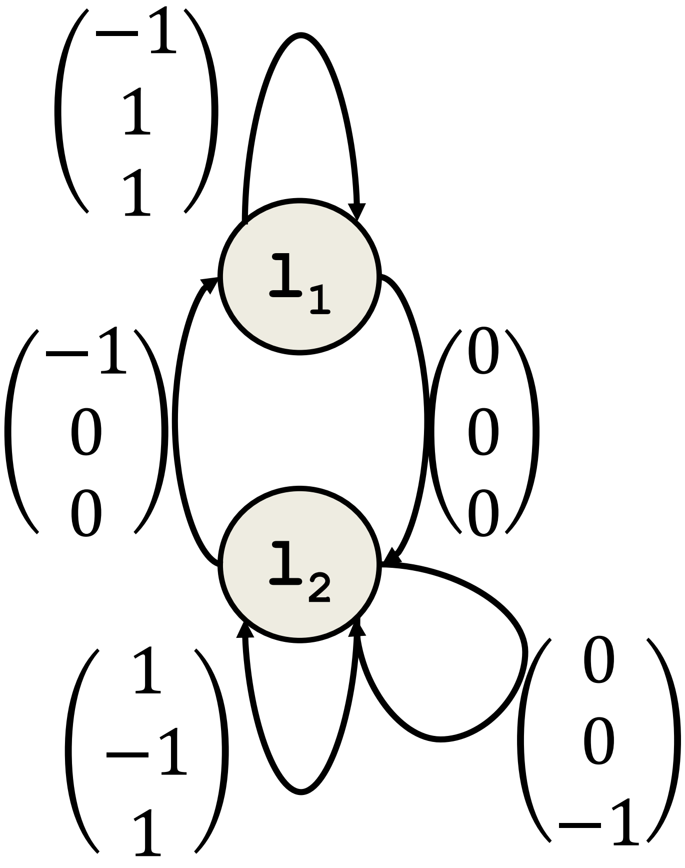
→ Abstraction



Initial State: $(l_1, \binom{N}{N})$

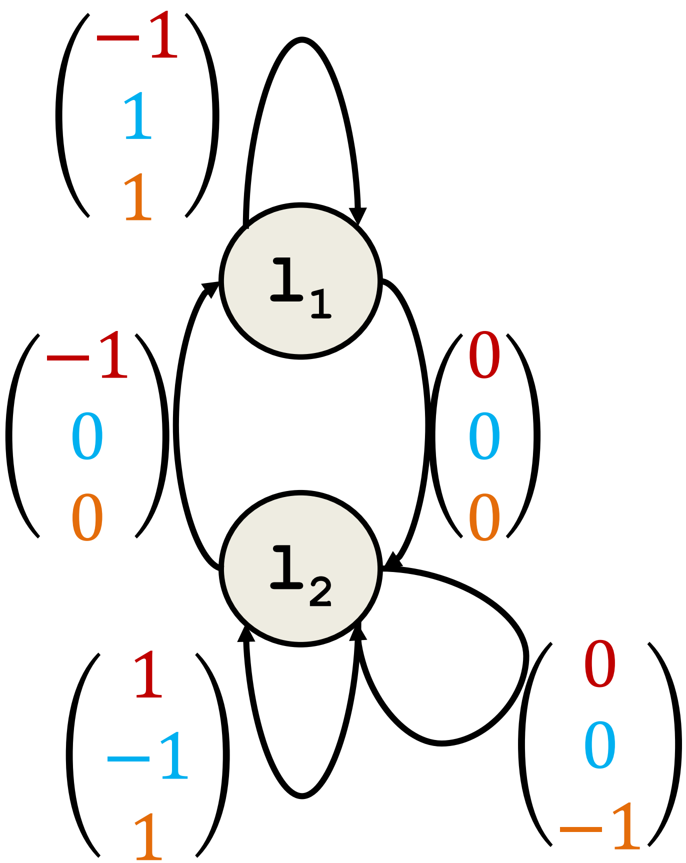
Complexity: O(N) ← **Complexity: Θ(N)**

Example



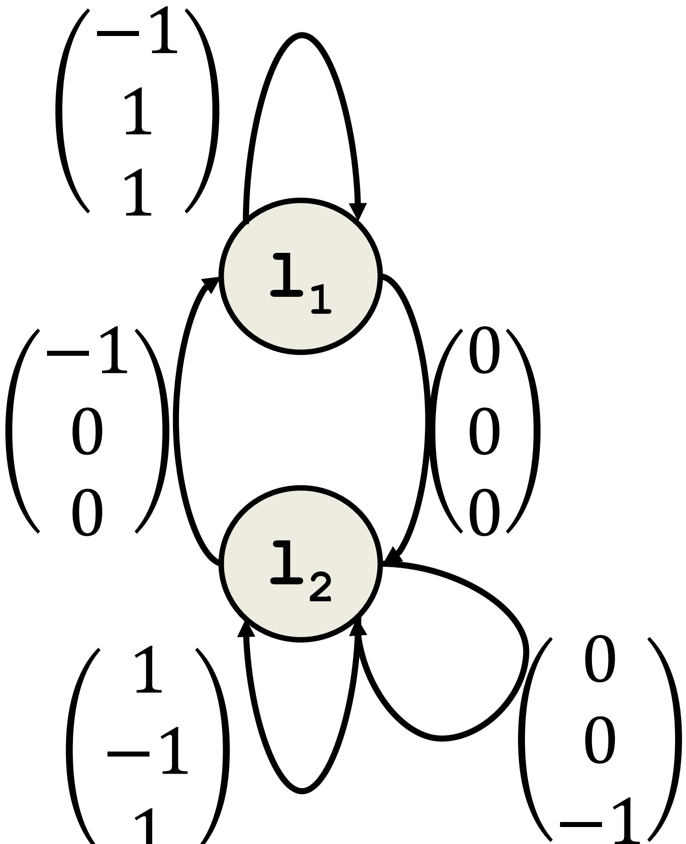
$\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$
l1
$\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$
l2
$\begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$
$\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$

Notation: Variables for Vector Components



$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Step 1a: Find linearly bounded variables



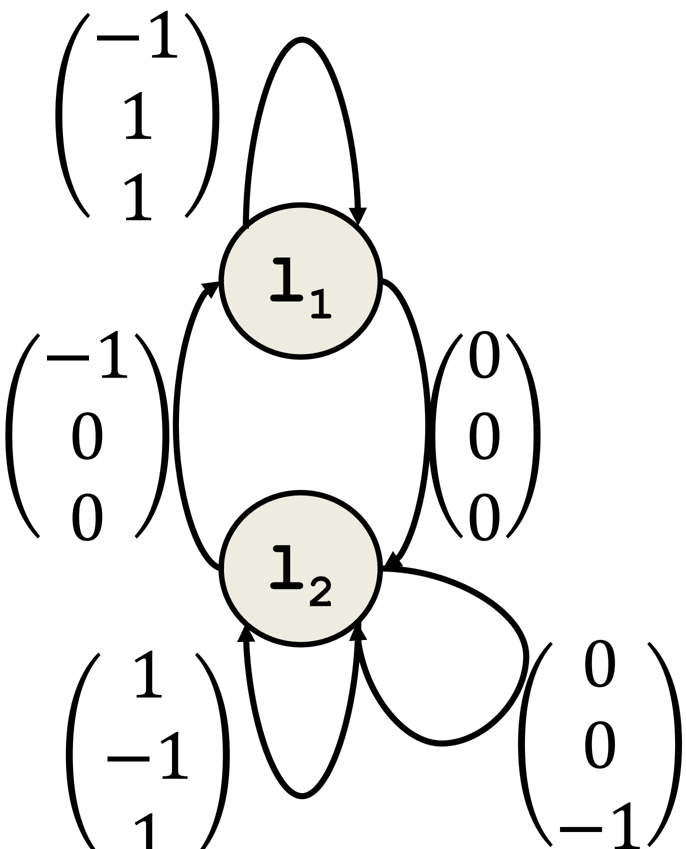
Find coefficients
a,b,c, $d(l_1)$, $d(l_2) \in \mathbb{N}$
such that

$$ax + by + cz + d(l_i) \geq$$
$$ax' + by' + cz' + d(l_j)$$

for all steps

$$(\mathtt{l}_i, \begin{pmatrix}x\\y\\z\end{pmatrix}) \longrightarrow (\mathtt{l}_j, \begin{pmatrix}x'\\y'\\z'\end{pmatrix})$$

Step 1a: Find linearly bounded variables



Find coefficients
a,b,c, $d(l_1)$, $d(l_2) \in \mathbb{N}$
such that

$$ax + by + cz + d(l_i) \geq ax' + by' + cz' + d(l_j)$$

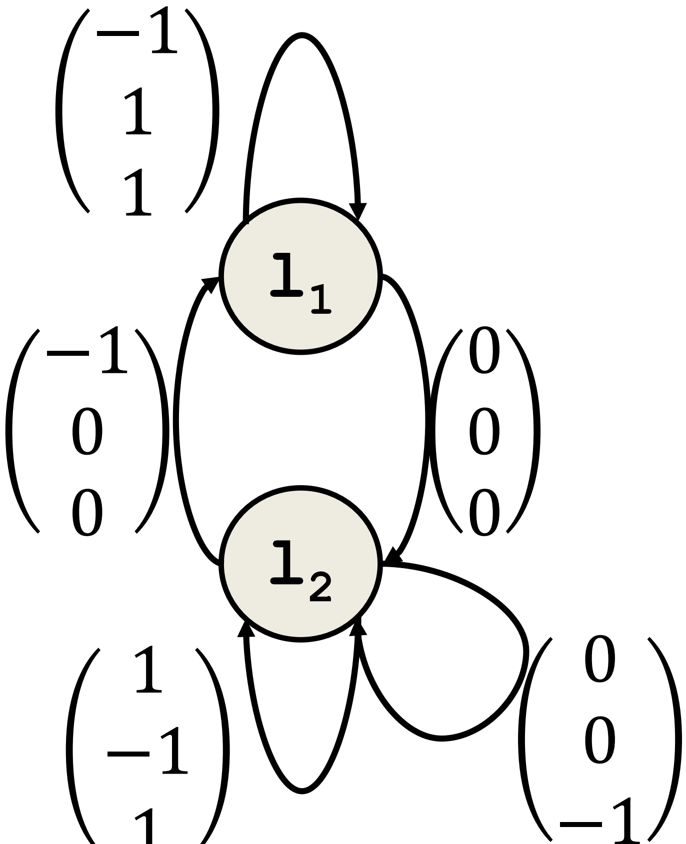
for all steps

$$(l_i, \begin{pmatrix} x \\ y \\ z \end{pmatrix}) \longrightarrow (l_j, \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix})$$

Solution:
a=2
b=2
c=0
$d(l_1)=1$
$d(l_2)=0$

Step 1a: Find linearly bounded variables

„>0“ → x,y are linearly bounded: O(N)



Find coefficients
a,b,c, $d(l_1)$, $d(l_2) \in \mathbb{N}$
such that

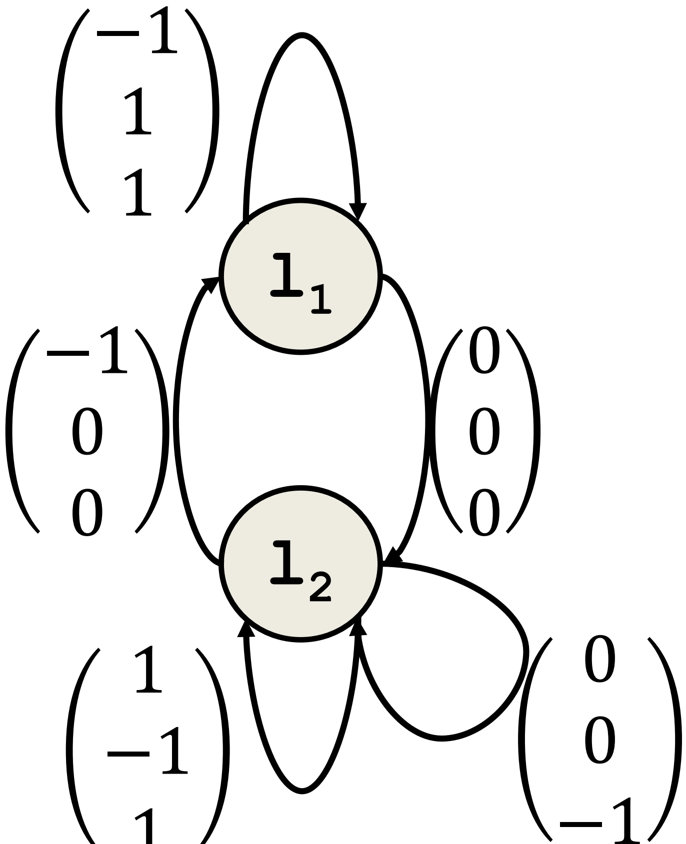
$$ax + by + cz + d(l_i) \geq ax' + by' + cz' + d(l_j)$$

for all steps

$$(l_i, \begin{pmatrix}x\\y\\z\end{pmatrix}) \longrightarrow (l_j, \begin{pmatrix}x'\\y'\\z'\end{pmatrix})$$

Solution:
a=2
b=2
c=0
$d(l_1)=1$
$d(l_2)=0$

Step 1a: Find linearly bounded variables



„>0“ → x,y are linearly bounded: O(N)

„=0“ → z is not linearly bounded

Find coefficients $a, b, c, d(l_1), d(l_2) \in \mathbb{N}$ such that

$$ax + by + cz + d(l_i) \geq ax' + by' + cz' + d(l_j)$$

for all steps

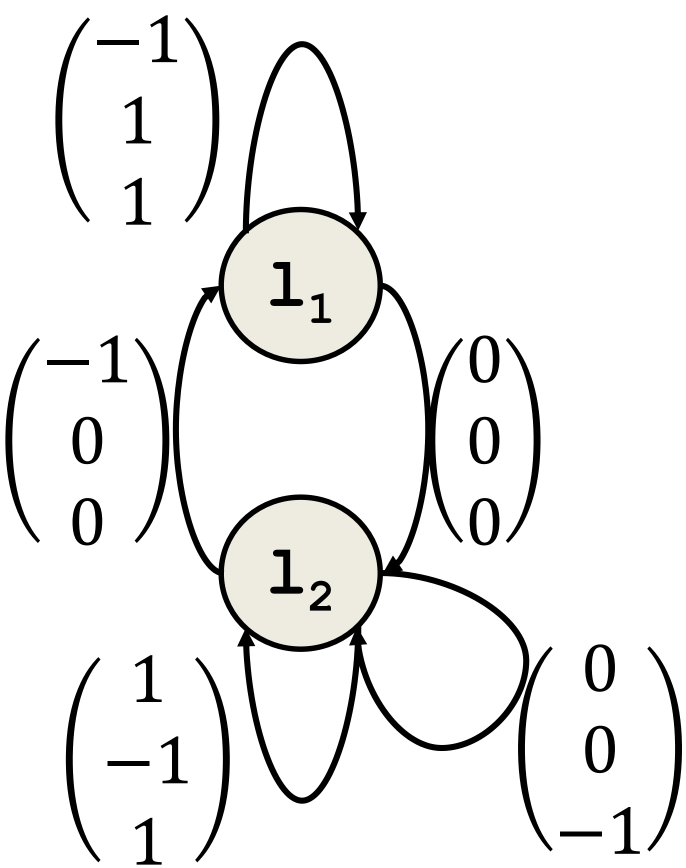
$$(l_i, \begin{pmatrix}x\\y\\z\end{pmatrix}) \longrightarrow (l_j, \begin{pmatrix}x'\\y'\\z'\end{pmatrix})$$

Solution:
- a=2
- b=2
- c=0
- $d(l_1)=1$
- $d(l_2)=0$

Step 1a: Find linearly bounded variables

„>0“ → x,y are linearly bounded: O(N)

„=0“ → z is not linearly bounded



Find coefficients $a, b, c, d(l_1), d(l_2) \in \mathbb{N}$ such that

$$ax + by + cz + d(l_i) \geq ax' + by' + cz' + d(l_j)$$

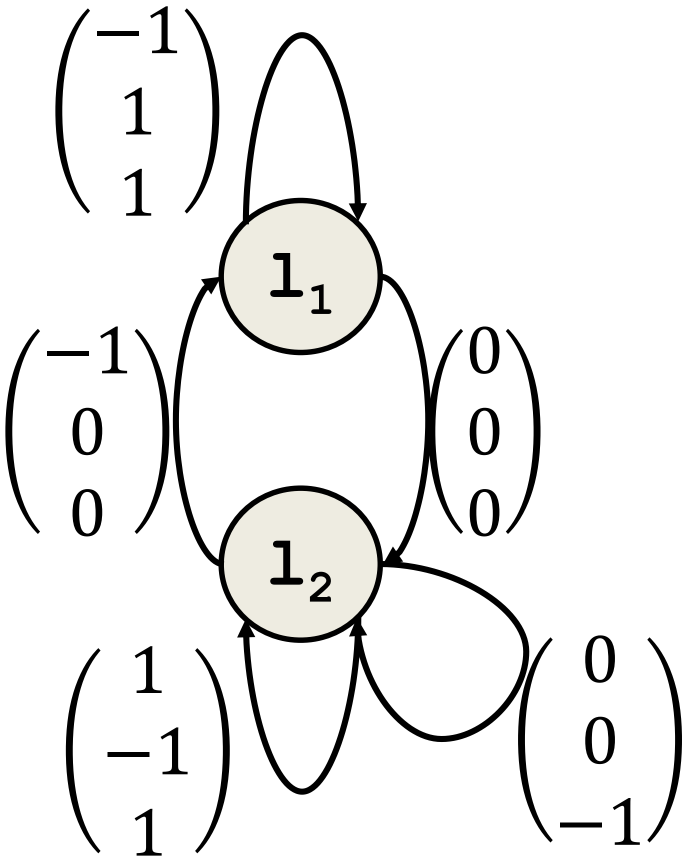
for all steps

$$(l_i, \begin{pmatrix} x \\ y \\ z \end{pmatrix}) \longrightarrow (l_j, \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix})$$

Solution:
a=2
b=2
c=0
$d(l_1)=1$
$d(l_2)=0$

Linear Programming

Step 1b: Remove decreasing transitions



We are actually interested in coefficients such that

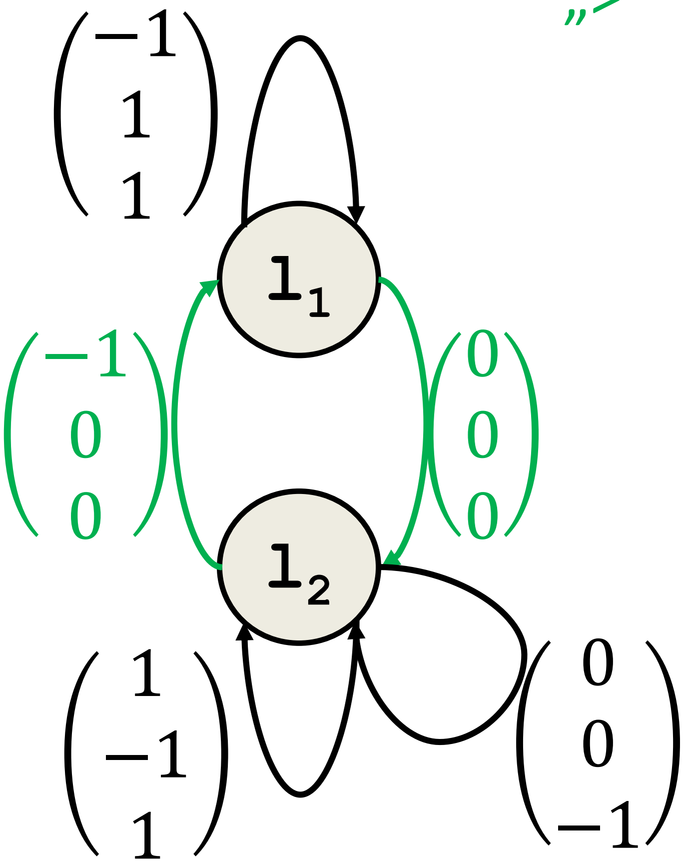
$$ax + by + cz + d(l_i) > ax' + by' + cz' + d(l_j)$$

for as many steps as possible

$$(\mathtt{l}_\mathtt{i}, \begin{pmatrix} x \\ y \\ z \end{pmatrix}) \longrightarrow (\mathtt{l}_\mathtt{j}, \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix})$$

Solution:
a=2
b=2
c=0
$d(l_1)=1$
$d(l_2)=0$

Step 1b: Remove decreasing transitions



„>“ → linearly bounded: O(N)

We are actually interested in coefficients such that

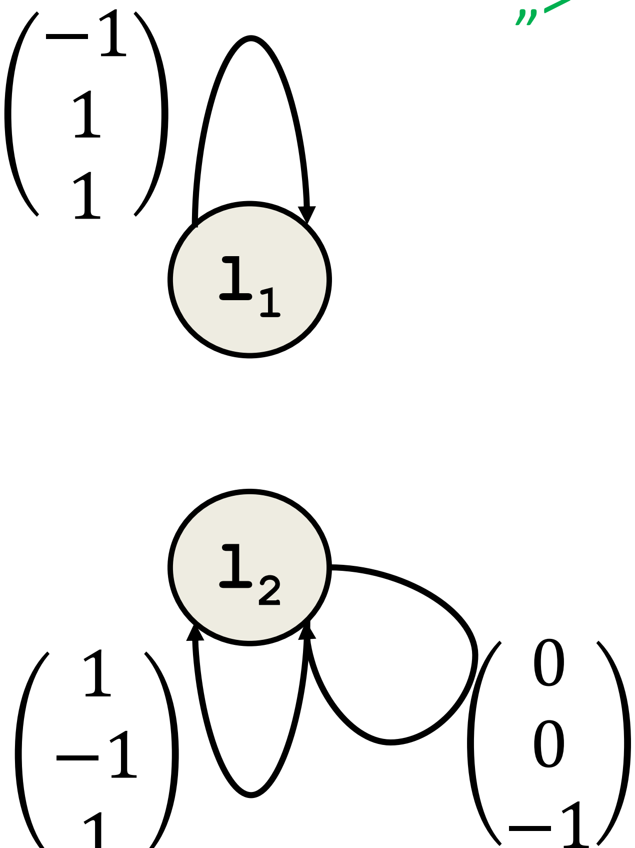
$$ax + by + cz + d(l_i) >$$
$$ax' + by' + cz' + d(l_j)$$

for as many steps as possible

$$(\mathtt{l}_i, \begin{pmatrix} x \\ y \\ z \end{pmatrix}) \longrightarrow (\mathtt{l}_j, \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix})$$

Solution:
$a=2$
$b=2$
$c=0$
$d(l_1)=1$
$d(l_2)=0$

Step 1b: Remove decreasing transitions



„>“ → linearly bounded: O(N)

We are actually interested in coefficients such that

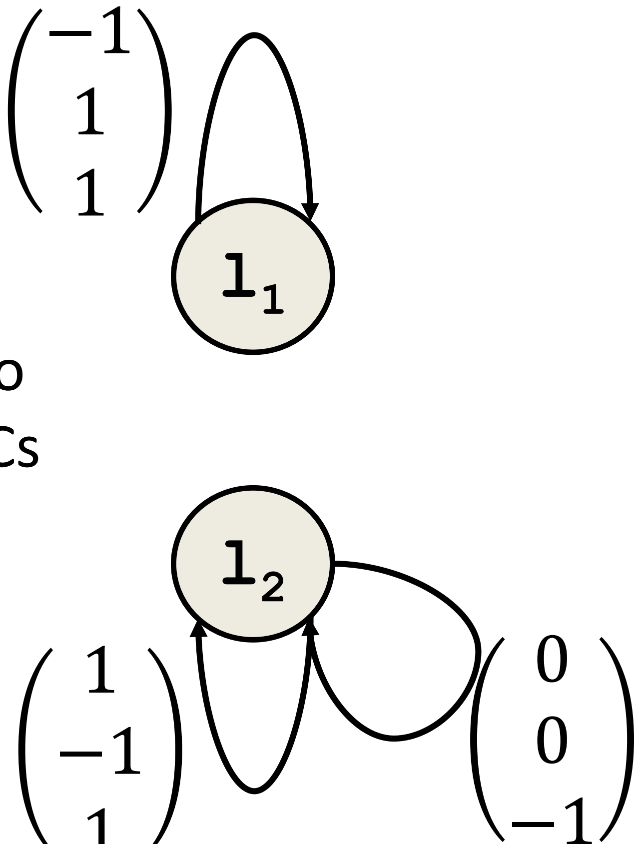
$$ax + by + cz + d(l_i) >$$
$$ax' + by' + cz' + d(l_j)$$

for as many steps as possible

$$\left(\mathtt{l}_i, \begin{pmatrix} x \\ y \\ z \end{pmatrix}\right) \longrightarrow \left(\mathtt{l}_j, \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}\right)$$

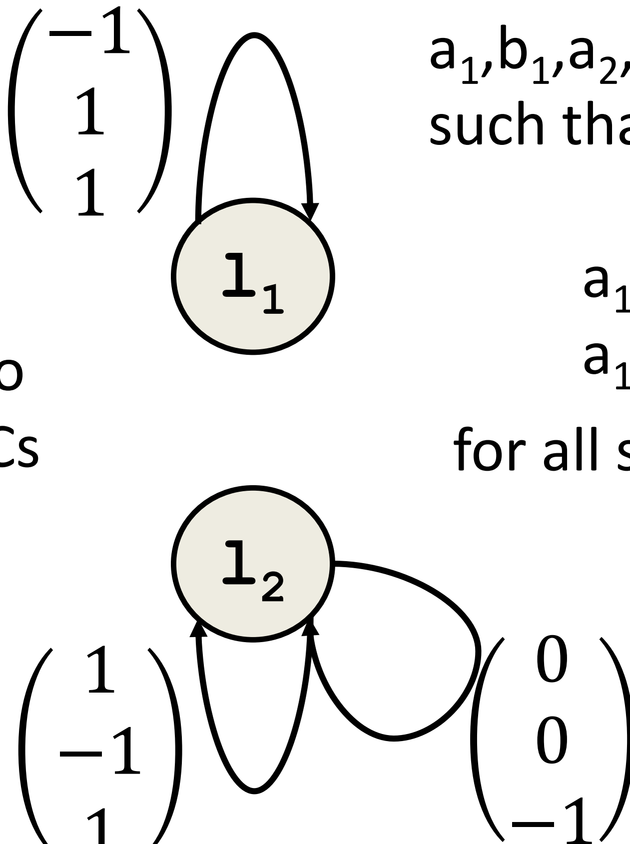
Solution:
a=2
b=2
c=0
$d(l_1)=1$
$d(l_2)=0$

Step 2a: Find quadratically bounded variables



Two SCCs

Step 2a: Find quadratically bounded variables



Two SCCs

Find coefficients
$a_1, b_1, a_2, b_2, c, d(l_1), d(l_2) \in \mathbb{N}$
such that

$$a_1 x + b_1 y + cz + d(l_1) \geq$$
$$a_1 x' + b_1 y' + cz' + d(l_1)$$

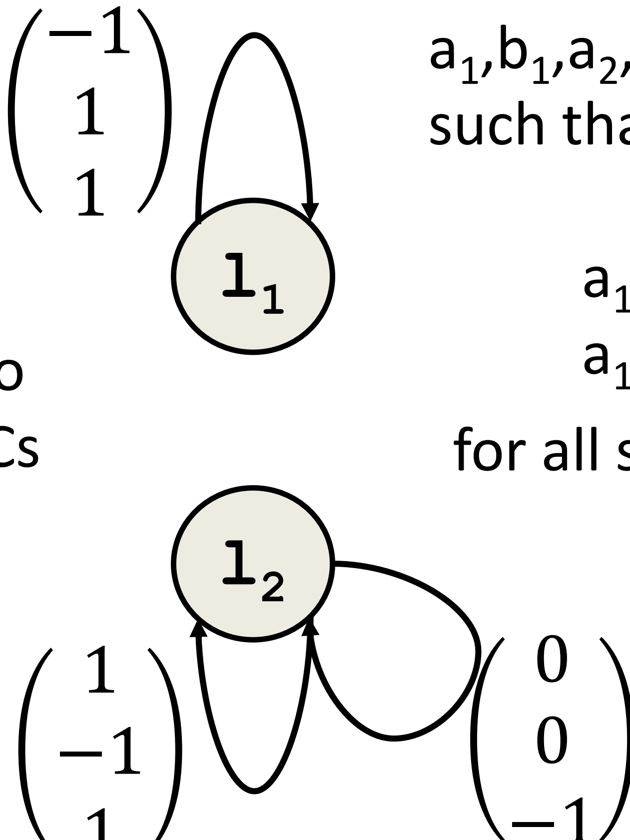
for all steps of SCC l_1, and

$$a_2 x + b_2 y + cz + d(l_2) \geq$$
$$a_2 x' + b_2 y' + cz' + d(l_2)$$

for all steps of SCC l_2.

Step 2a: Find quadratically bounded variables

Two SCCs



Find coefficients
$a_1, b_1, a_2, b_2, c, d(l_1), d(l_2) \in \mathbb{N}$
such that

$$a_1x + b_1y + cz + d(l_1) \geq a_1x' + b_1y' + cz' + d(l_1)$$

for all steps of SCC l_1, and

$$a_2x + b_2y + cz + d(l_2) \geq a_2x' + b_2y' + cz' + d(l_2)$$

for all steps of SCC l_2.

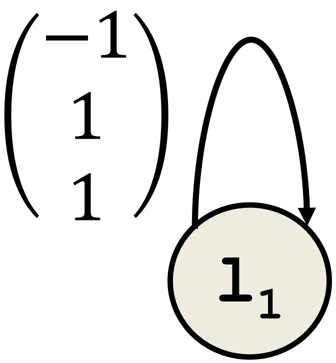
Solution:
$a_1=3$
$b_1=1$
$a_2=1$
$b_2=3$
$c=1$
$d(l_1)=0$
$d(l_2)=0$

Linear Programming

Step 2a: Find quadratically bounded variables

„>0“ → z is quadratically bounded: $O(N^2)$

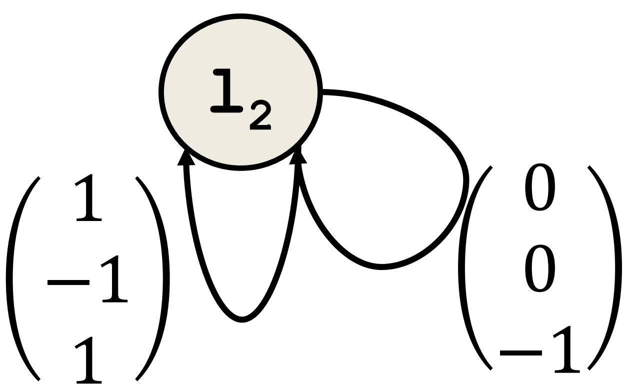
Find coefficients
$a_1, b_1, a_2, b_2, c, d(l_1), d(l_2) \in \mathbb{N}$
such that



$$a_1 x + b_1 y + cz + d(l_1) \geq a_1 x' + b_1 y' + cz' + d(l_1)$$

Two SCCs

for all steps of SCC l_1, and



$$a_2 x + b_2 y + cz + d(l_2) \geq a_2 x' + b_2 y' + cz' + d(l_2)$$

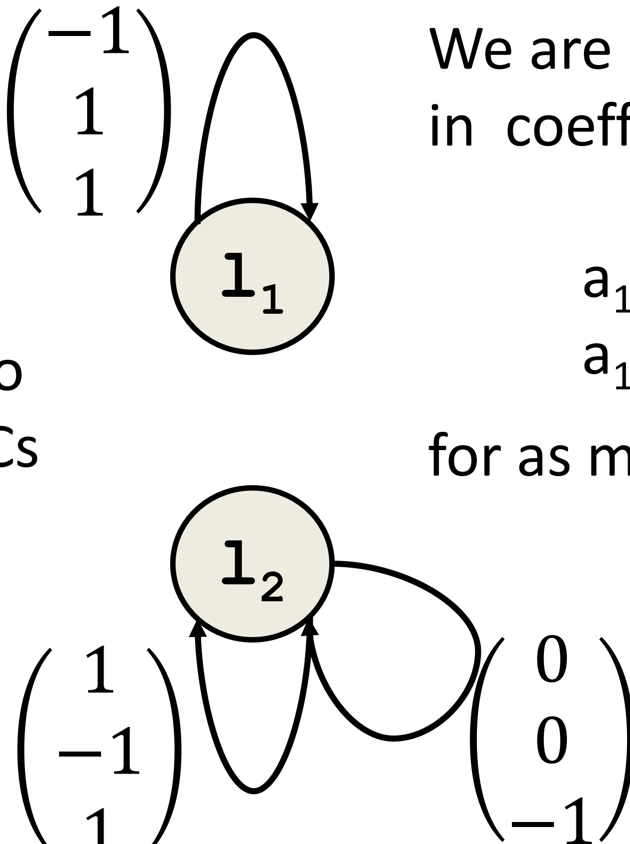
for all steps of SCC l_2.

Solution:
$a_1=3$
$b_1=1$
$a_2=1$
$b_2=3$
$c=1$
$d(l_1)=0$
$d(l_2)=0$

Linear Programming

Step 2b: Remove decreasing transitions

Two SCCs



We are actually interested in coefficients such that

$$a_1x + b_1y + cz + d(l_1) > a_1x' + b_1y' + cz' + d(l_1)$$

for as many steps of SCC l_1, and

$$a_2x + b_2y + cz + d(l_2) > a_2x' + b_2y' + cz' + d(l_2)$$

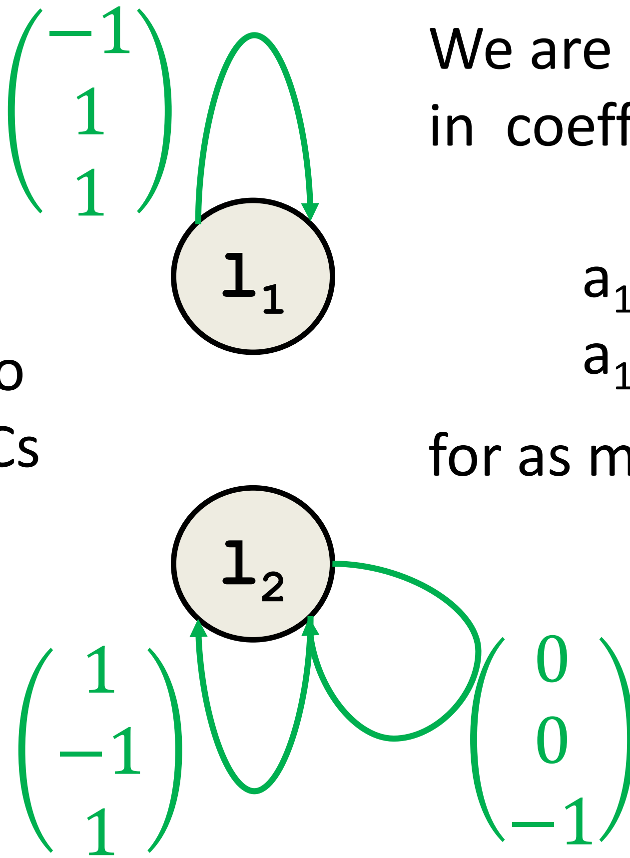
for as many steps of SCC l_2 as possible.

Solution:
$a_1=3$
$b_1=1$
$a_2=1$
$b_2=3$
$c=1$
$d(l_1)=0$
$d(l_2)=0$

Step 2b: Remove decreasing transitions

„>“ → quadratically bounded: $O(N^2)$

Two SCCs



We are actually interested in coefficients such that

$$a_1x + b_1y + cz + d(l_1) > a_1x' + b_1y' + cz' + d(l_1)$$

for as many steps of SCC l_1, and

$$a_2x + b_2y + cz + d(l_2) > a_2x' + b_2y' + cz' + d(l_2)$$

for as many steps of SCC l_2 as possible.

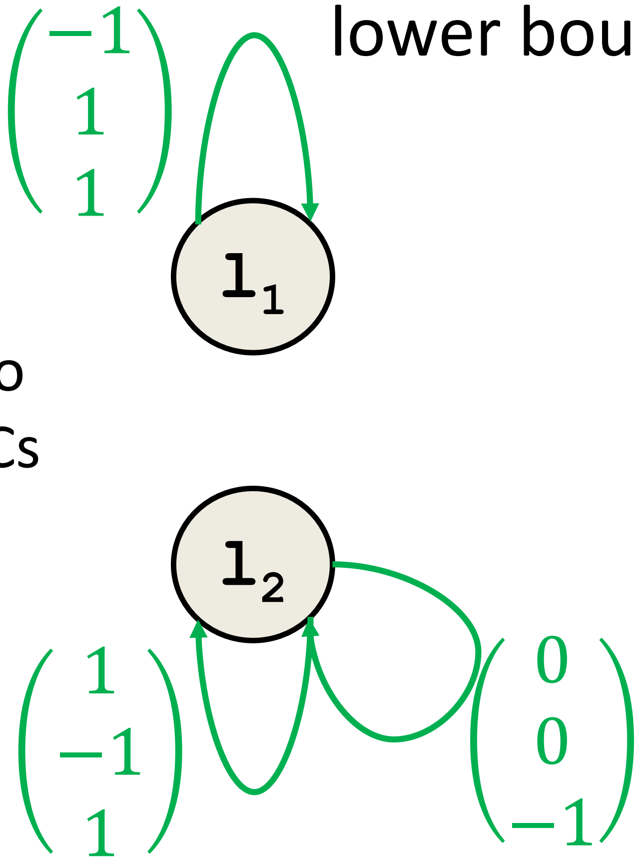
Solution:
$a_1=3$
$b_1=1$
$a_2=1$
$b_2=3$
$c=1$
$d(l_1)=0$
$d(l_2)=0$

Precision?

Do we also have the lower bound $\Omega(N^2)$?

quadratically bounded: $O(N^2)$

Two SCCs

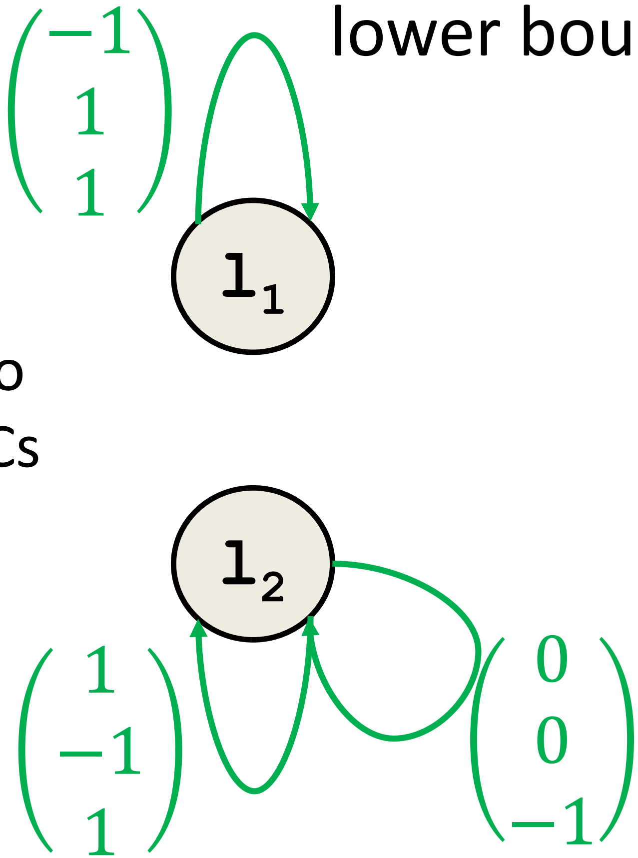


Precision?

Do we also have the lower bound $\Omega(N^2)$?

Yes!

quadratically bounded: $O(N^2)$



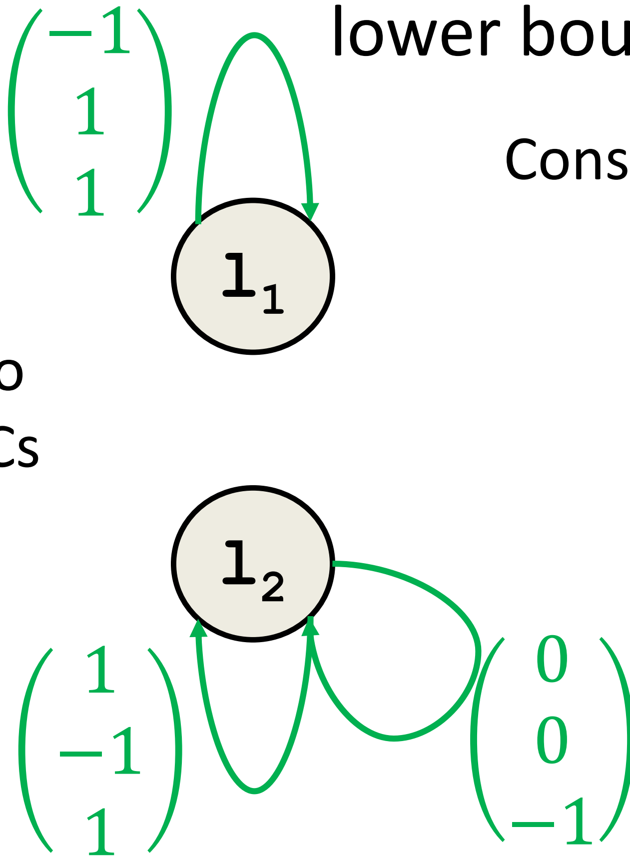
Two SCCs

Precision?

Do we also have the lower bound $\Omega(N^2)$? Yes!

quadratically bounded: $O(N^2)$

Two SCCs



Consider the two cycles

$$l_1 \xrightarrow{\begin{pmatrix}-1\\1\\1\end{pmatrix}} l_1 \xrightarrow{\begin{pmatrix}-1\\1\\1\end{pmatrix}} l_1$$

$$l_2 \xrightarrow{\begin{pmatrix}1\\-1\\1\end{pmatrix}} l_2 \xrightarrow{\begin{pmatrix}1\\-1\\1\end{pmatrix}} l_2 \xrightarrow{\begin{pmatrix}0\\0\\-1\end{pmatrix}} l_2$$

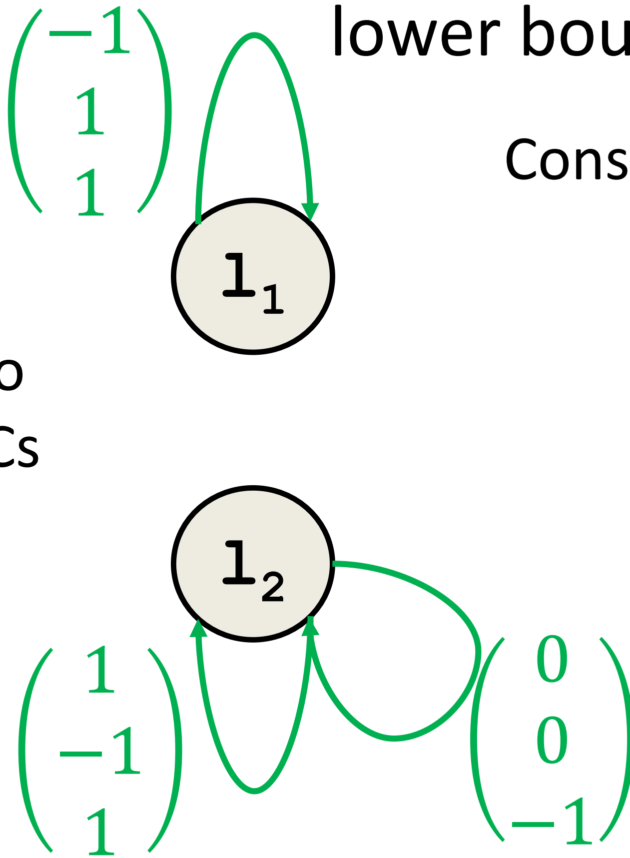
Sum of updates: $\begin{pmatrix}0\\0\\1\end{pmatrix}$

<u>Iteration scheme:</u> Execute some cycle $\Omega(N)$ times, then the other cycle the same number of times. Do this $\Omega(N)$ times.

Precision?

Do we also have the lower bound $\Omega(N^2)$? Yes!

quadratically bounded: $O(N^2)$



Two SCCs

Consider the two cycles

$$l_1 \xrightarrow{\begin{pmatrix}-1\\1\\1\end{pmatrix}} l_1 \xrightarrow{\begin{pmatrix}-1\\1\\1\end{pmatrix}} l_1$$

$$l_2 \xrightarrow{\begin{pmatrix}1\\-1\\1\end{pmatrix}} l_2 \xrightarrow{\begin{pmatrix}1\\-1\\1\end{pmatrix}} l_2 \xrightarrow{\begin{pmatrix}0\\0\\-1\end{pmatrix}} l_2$$

Sum of updates: $\begin{pmatrix}0\\0\\1\end{pmatrix}$

- Cycles are extracted during Step 1 by a **dual linear program**
- Iterations schemes always exists

Contributions

- Full characterization of VASSs with polynomial complexity
- PTIME algorithms promise to be of practical use
- Interesting application of duality in linear programming: during each step we find a ranking function or a set of cycles that prove an upper resp. lower complexity bound