The reachability problem for Petri nets

Sławomir Lasota
University of Warsaw

ACPN 2023, Toruń, 2023-09-05

I. Intro

II. Decidability

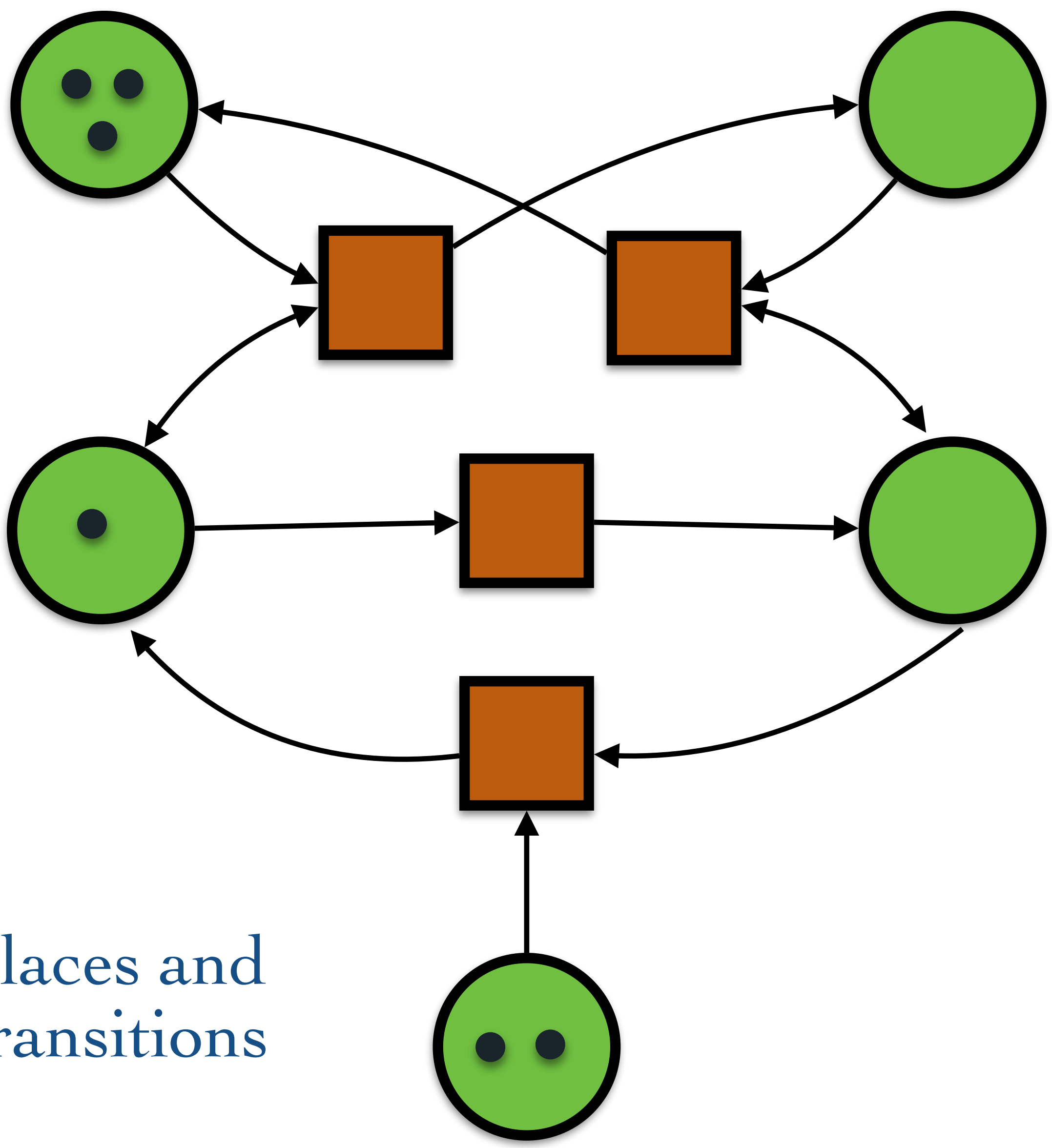
III. F_ω-hardness

I. Intro

- **reachability and coverability**
- equivalent models
- coverability tree
- characteristic equation

~~Reachability~~ Coverability problem in Petri nets

Petri net:



places and
transitions

configuration : places → $\mathbb{N}$ $\mathbb{N}^d$

step relation between configurations

Decision problem:

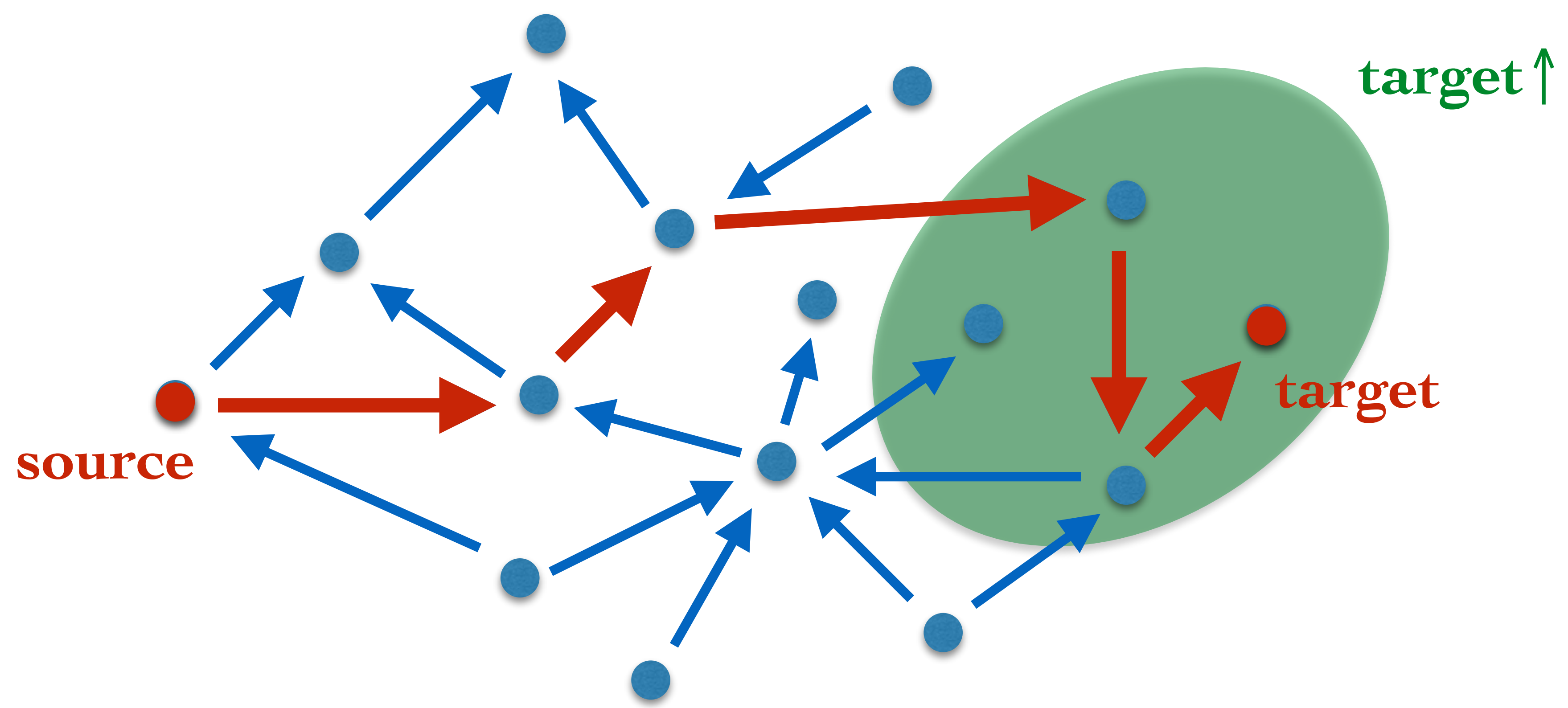
given

- Petri net
- source configuration
- target configuration

check if there is a sequence of steps (**run**) from source to ~~target~~ ≥ target

~~Reachability~~ Coverability problem in Petri nets

configuration graph: configurations and steps



Reachability: is there a path (**run**) from source to target ?

Coverability: is there a path (**run**) from source to target↑ ?

Why is it important?

- core verification problem
- equivalent to many other problems in concurrency, process algebra, logic, language theory, linear algebra, etc

1969 — decidability of coverability **[Karp, Miller JCSS]**

1970

1976 — EXPSPACE lower bound **[Lipton TR Yale U]**

1977 — (incomplete) decidability of reachability **[Sacerdote, Tenney STOC]**

1978 — EXPSPACE algorithm for coverability **[Rackoff TCS]**

1980

1981 — decidability of reachability **[Mayr STOC]**

1982 — decidability of reachability - simplified proof **[Kosaraju STOC]**

Part II

huge complexity gap!

1990

1992 — decidability of reachability - refined data structure **[Lambert TCS]**

2000

2010

2011 — decidability of reachability by Presburger invariants **[Leroux POPL]**

2015 — upper bound F_{ω^3} **[Leroux, Schmitz LICS]**

2019 — Ackermannian upper bound F_{ω} **[Leroux, Schmitz LICS]**

2019 — TOWER lower bound F_3 **[Czerwiński, L., Lazic, Leroux, Mazowiecki STOC]**

} **gap**

2020

2019 — Ackermannian upper bound F_ω [Leroux, Schmitz LICS]

2019 — TOWER lower bound F_3 [Czerwiński, L., Lazic, Leroux, Mazowiecki STOC]

} gap

2020

$$\left.2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}\right\}n$$

2021

2021 — super-TOWER lower bound [Czerwiński, L., Orlikowski ICALP]

$$\left.2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}\right\}2^n$$

gap closed!

2021 — Ackermannian lower bound F_ω [Czerwiński, Orlikowski FOCS] [Leroux FOCS]

2021 — improved and simplified Ackermannian lower bound F_ω [L. STACS] **Part III**

2022

Fast growing functions and induced complexity classes

$A_1(n) = 2n$

$A_{i+1}(n) = \underbrace{A_i \circ A_i \circ \ldots \circ A_i}_{n}(1) = A_i^n(1)$

$A_\omega(n) = A_n(n)$ Ackermann function

$A_2(n) = 2^n$

$A_3(n) = \text{tower}(n) = \left.2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}\right\}n$

$A_4(n) = \ldots$

$$\mathcal{F}_i = \bigcup_{j_1 \ldots j_m < i} \text{DTIME}(A_i \circ A_{j_1} \circ \ldots \circ A_{j_m})$$

$\mathcal{F}_2 = \text{DTIME}(2^{O(n)})$

$\mathcal{F}_3 = \text{TOWER}$

$\ldots$

$\mathcal{F}_\omega = \text{ACKERMANN}$

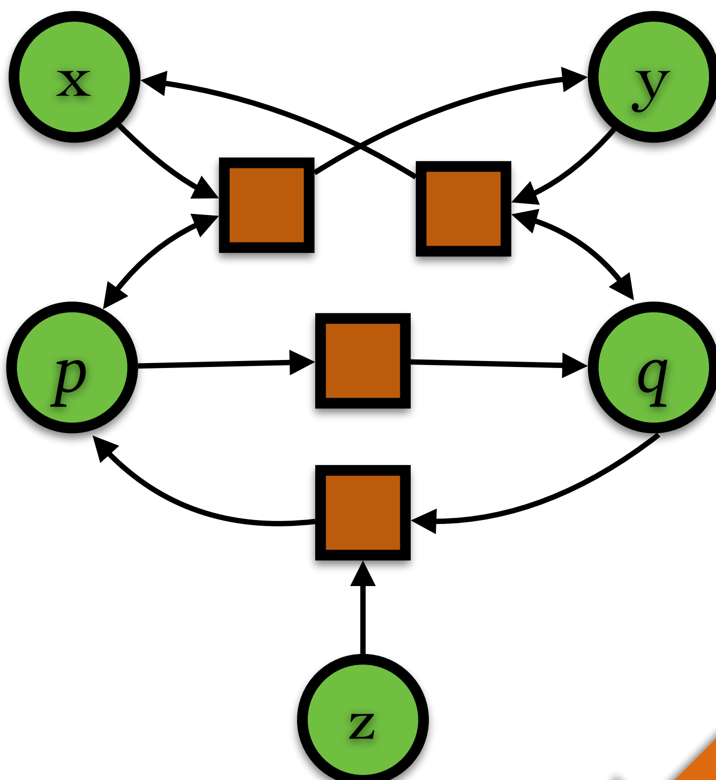
I. Intro

- reachability and coverability
- **equivalent models**
- coverability tree
- characteristic equation

Many faces of Petri nets

Part III

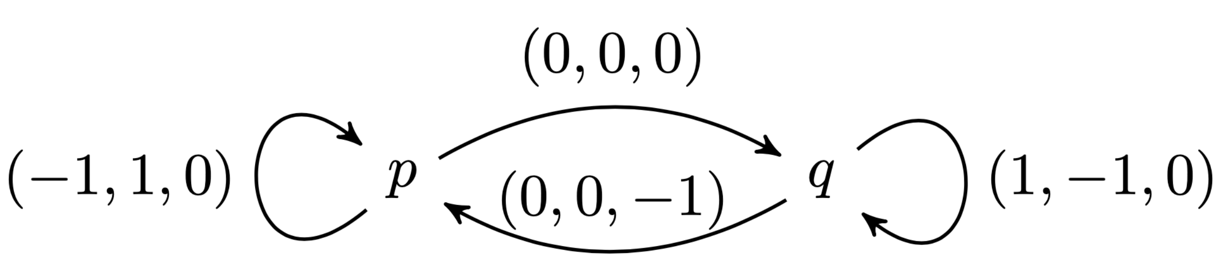
- Petri nets:



- counter programs **without zero-tests**:

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

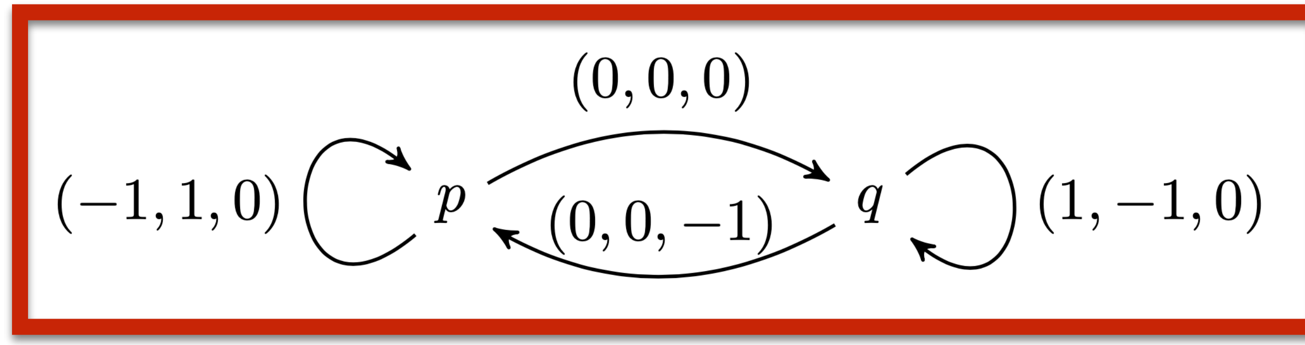
- vector addition systems with states (VASS):



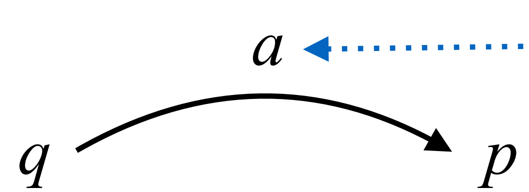
Part II

- vector addition systems
- counter automata **without zero-tests**
- multiset rewriting
- …

VASS



- dimension d
- finite set of control states Q
- finite set of transitions of the form:



effect of transition

$q, p \in Q$

$a \in \mathbb{Z}^d$

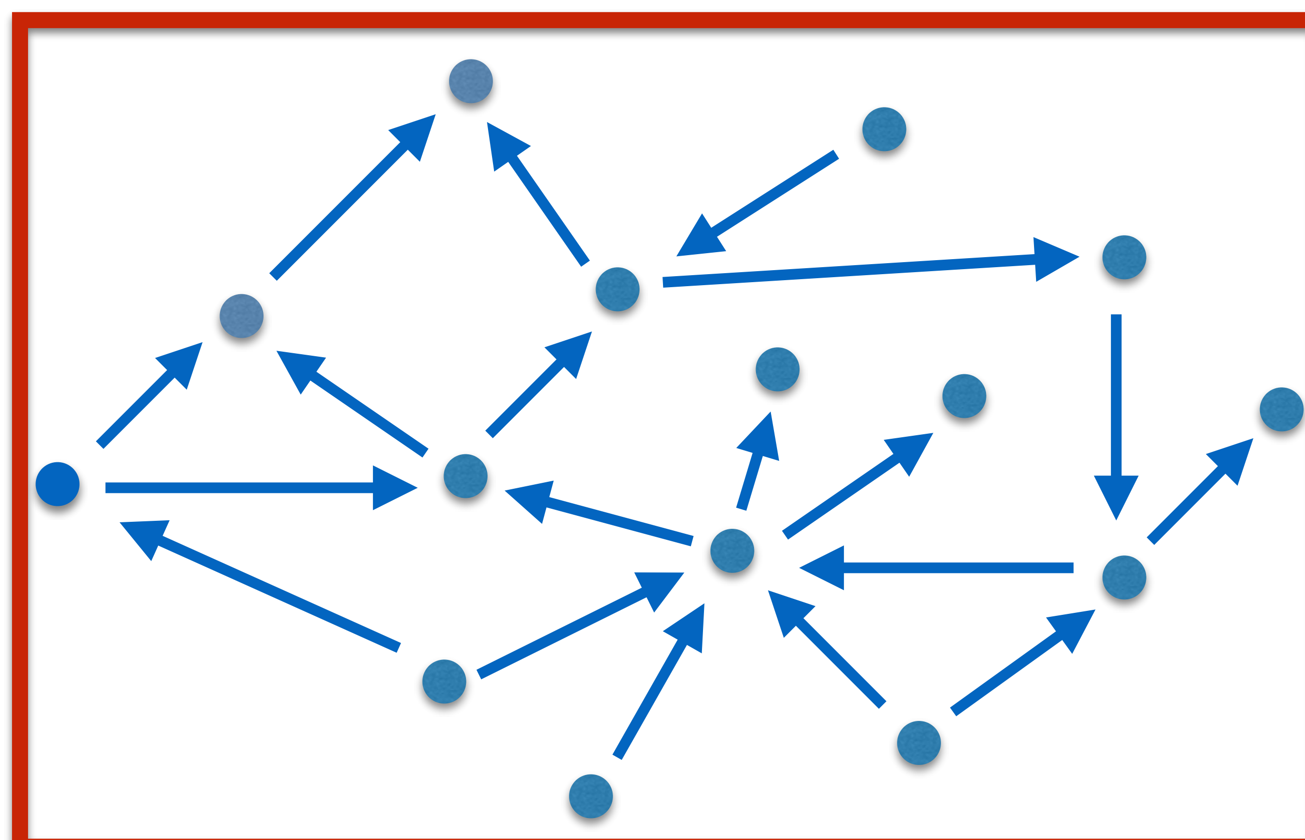
two different graphs!

- configurations $(q, v) = q(v) \in Q \times \mathbb{N}^d$
- step relation:

$$q(v) \longrightarrow p(v+a)$$

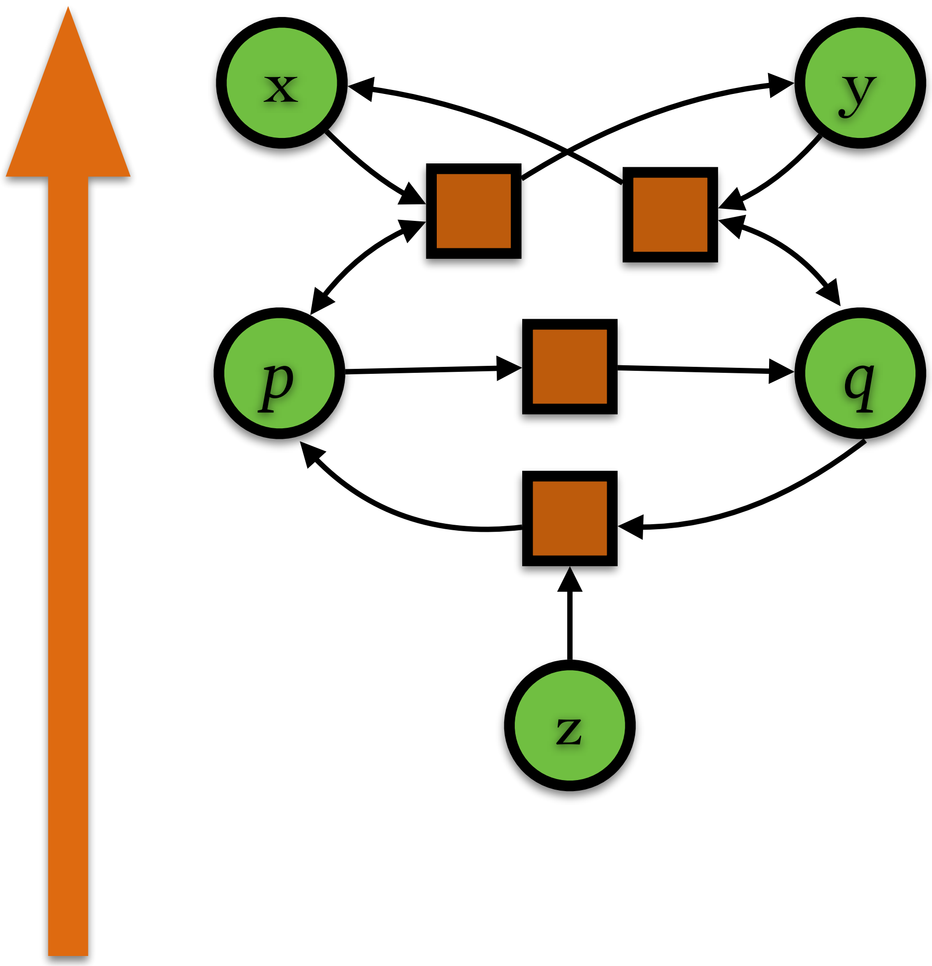
- reachability relation:

$$q(v) \longrightarrow^* p(w)$$

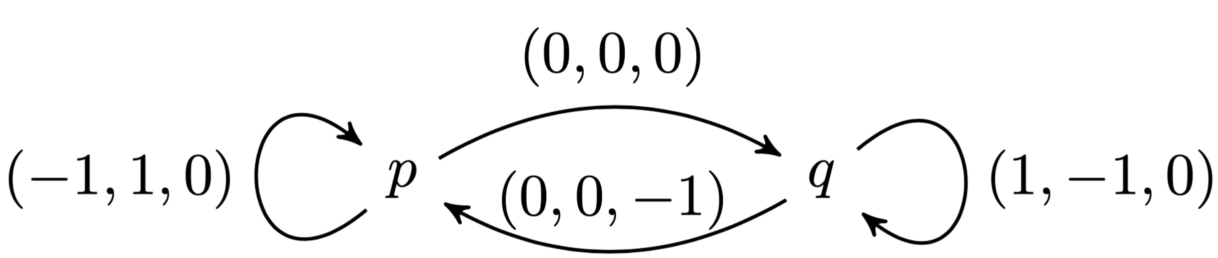


Petri nets ⇆ VASS

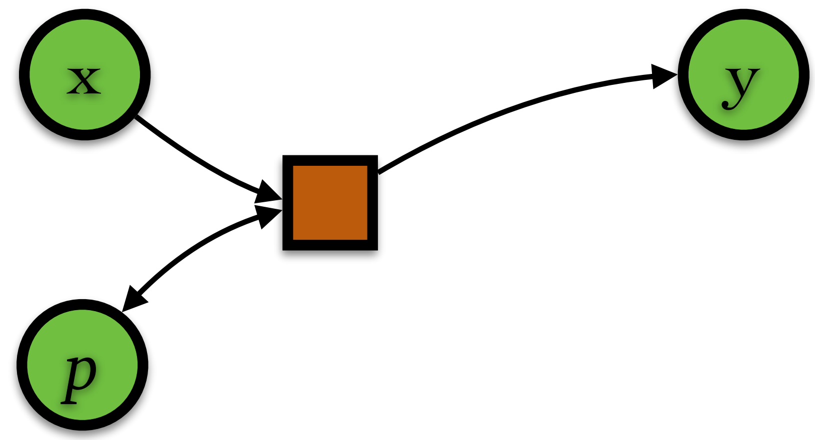
- Petri nets:



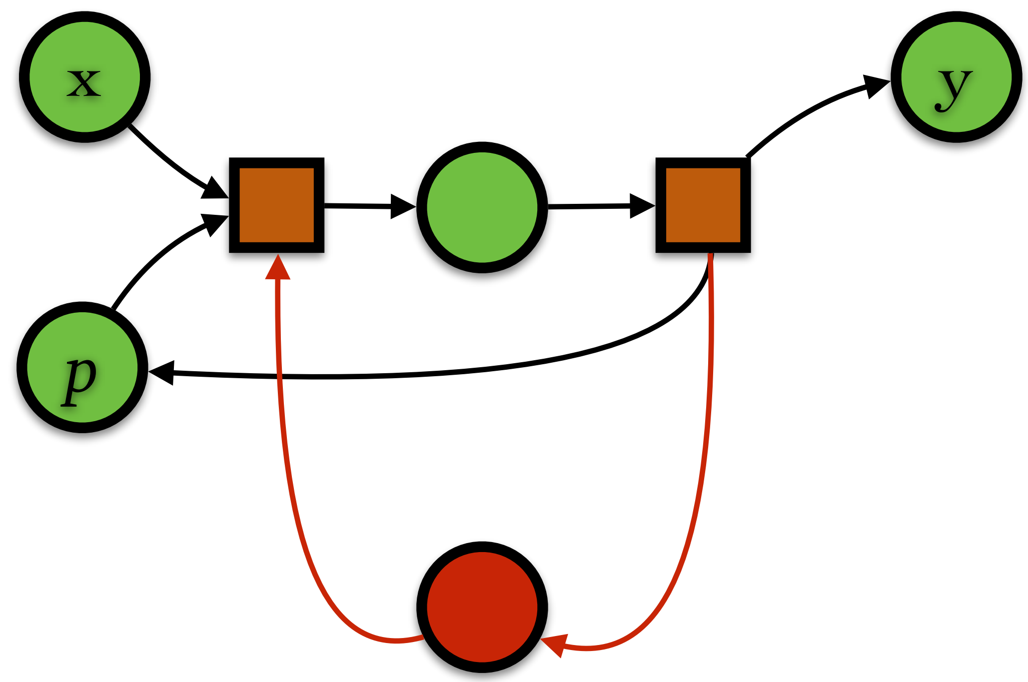
- vector addition systems with states (VASS):



split every transition



into input and output:



then add one more "global" place

Counter programs **without zero-tests**

counters are nonnegative integer variables initially all equal zero

Counter program = a sequence of commands of the form:

$\mathsf{x} \mathrel{+}= n$	(increment counter x by n)	
$\mathsf{x} \mathrel{-}= n$	(decrement counter x by n)	abort if x < *n*
goto L **or** L'	(jump to either line L or line L')	nondeterminism

except for the very last command which is of the form:

halt if $\mathsf{x}_1, \ldots, \mathsf{x}_l = 0$	(terminate provided all the listed counters are zero)	otherwise abort

Example:

1: $\mathsf{x}' \mathrel{+}= 100$
2: **goto** 5 **or** 3
3: $\mathsf{x} \mathrel{+}= 1 \quad \mathsf{x}' \mathrel{-}= 1 \quad \mathsf{y} \mathrel{+}= 2$
4: **goto** 2
5: **halt if** $\mathsf{x}' = 0$.

no zero tests

initially: x' = x = y = 0

1: $\mathsf{x}' \mathrel{+}= 100$
2: **loop**
$\mathsf{x} \mathrel{+}= 1 \quad \mathsf{x}' \mathrel{-}= 1 \quad \mathsf{y} \mathrel{+}= 2$
4: **halt if** $\mathsf{x}' = 0$.

finally: x' = 0 x = 100 y = 200

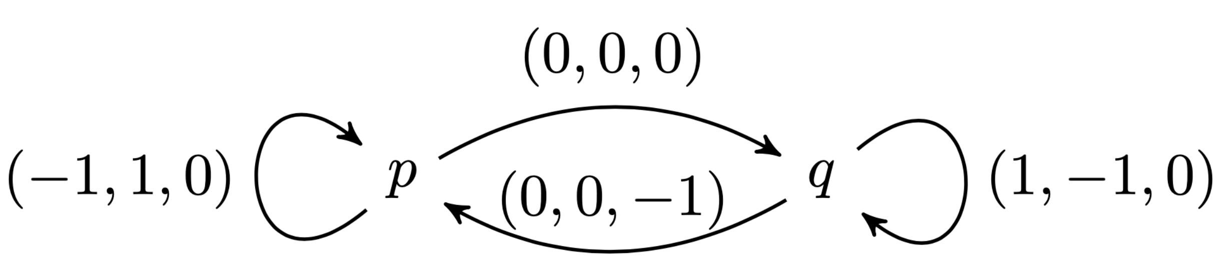
Counter programs → VASS

- counter programs **without zero-tests:**

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

- dimension := number of counters
- control states := control locations
- transitions := commands

- vector addition systems with states (VASS):



Counter programs **with zero-tests**

zero test command:

zero? x	(continue if counter x equals 0)

otherwise abort

Example:

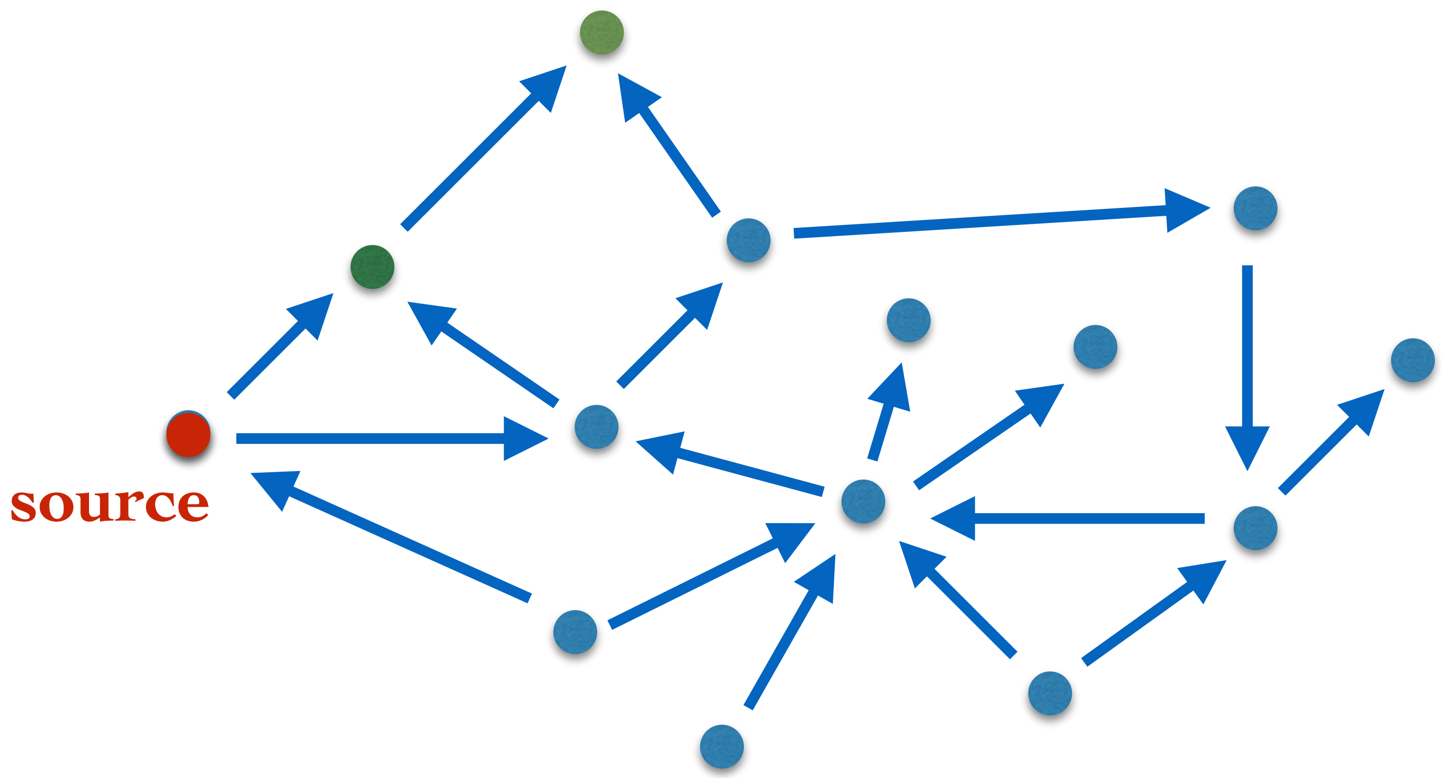
1: x += 100
2: **goto** 3 **or** 5
3: x −= 1
4: **goto** 2
5: **zero?** x
6: x += 1

counter programs **with zero-tests** are Turing complete

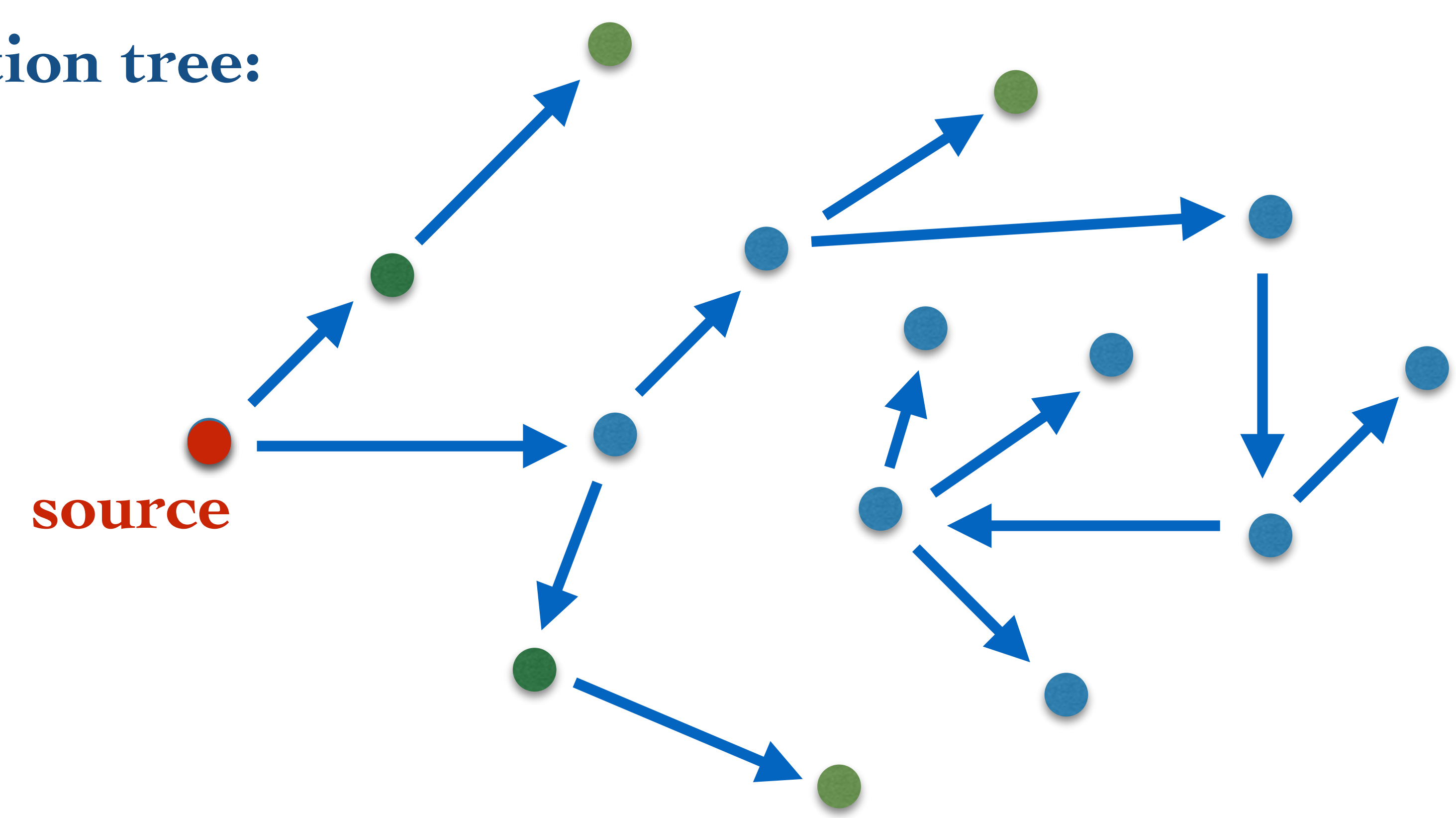
I. Intro

- reachability and coverability
- equivalent models
- **coverability tree**
- characteristic equation

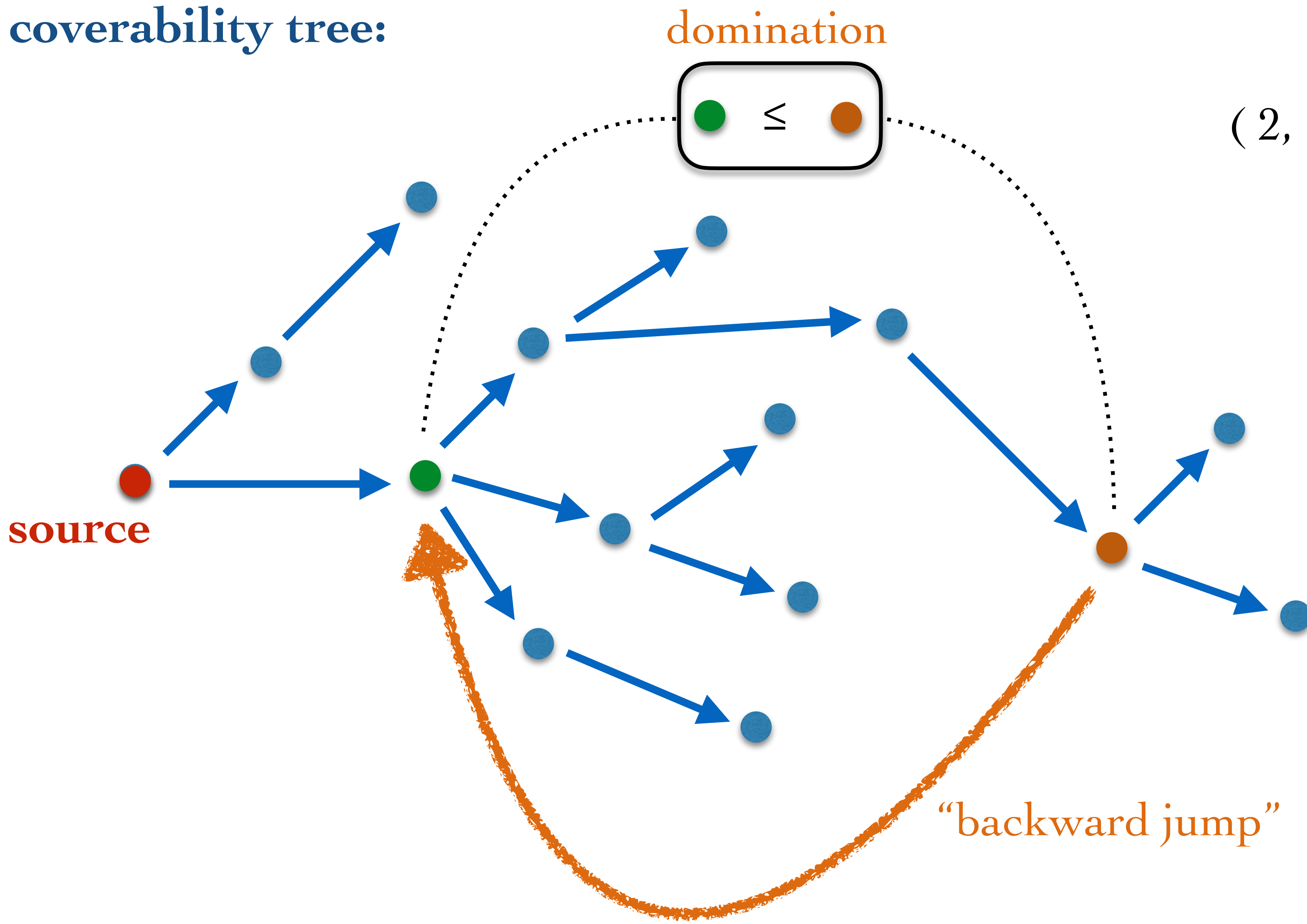
configuration graph:



configuration tree:



coverability tree:



$(2, \underline{1, 6}, 3) \leq (2, 3, 7, 3)$

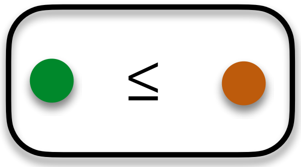
increased

$(2, \omega, \omega, 3)$

if ● = ● **then** stop generating the tree

if ● < ● **then** replace increased coordinates by ω

domination



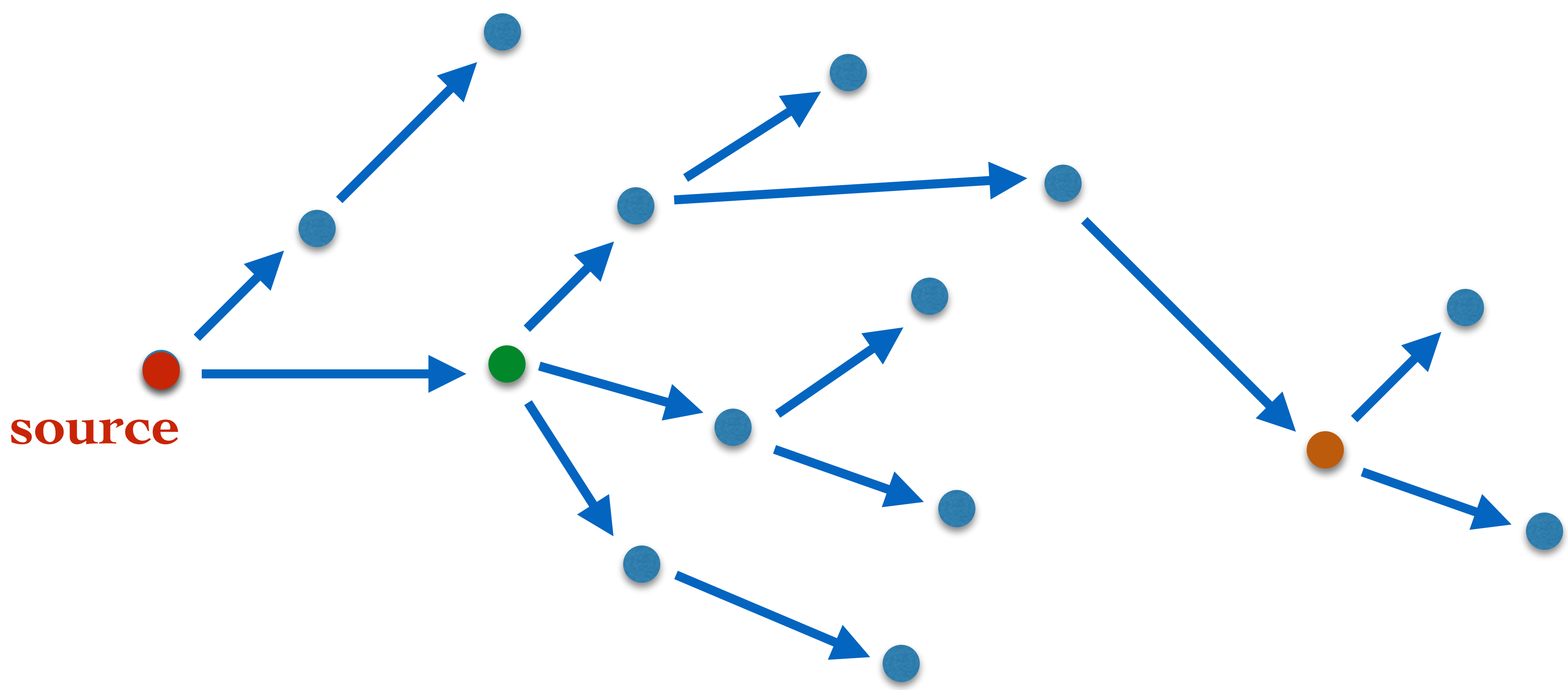
Dickson's Lemma: every infinite sequence of configurations

$_1$ $_2$ $_3$ ....

admits a domination:

$_i$ ≤ $_j$ for some $i < j$.

Coverability tree



Theorem: Coverability tree is finite.

Coverable configurations = (coverability tree) $\downarrow$

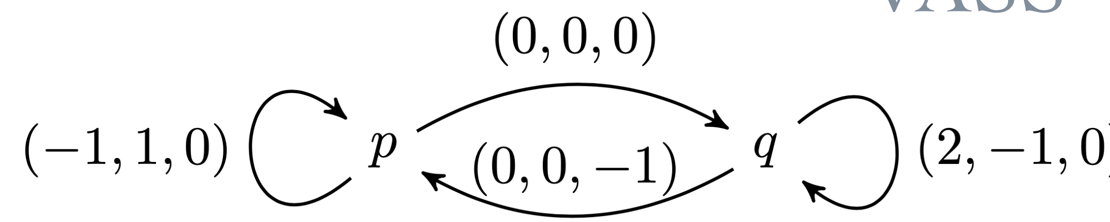
Question: What can be read out from coverability tree?

I. Intro

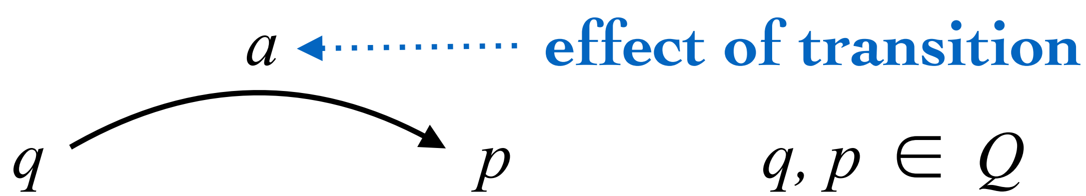
- reachability and coverability
- equivalent models
- coverability tree
- **characteristic equation**

Characteristic equation

VASS



- dimension d
- finite set of control states Q
- finite set of transitions T of the form:



$q, p \in Q$ $\qquad a \in \mathbb{Z}^d$

- source $q(v)$, target $p(w) \in Q \times \mathbb{N}^d$ $\qquad q, p$ distinct

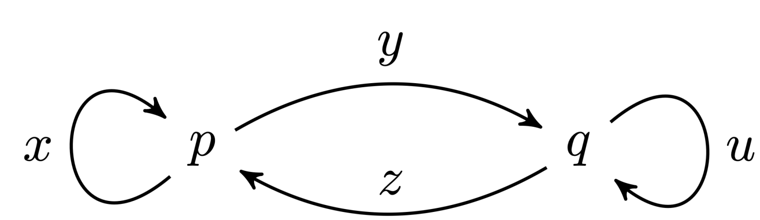
- one variable per transition in T, to represent the number of its applications
- for each control state, an equation

nr of incoming transitions = nr of outgoing transitions

except for p, q …

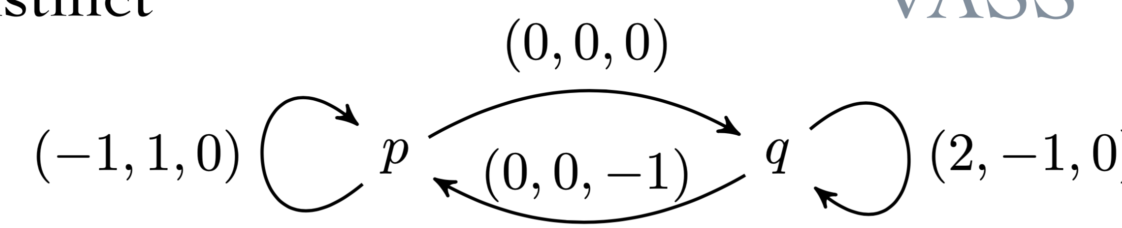
Example: $x + z + 1 = x + y$

$y + u = u + z + 1$



VASS

- source $q(v)$, target $p(w) \in Q \times \mathbb{N}^d$ $\quad q, p$ distinct



- one variable per transition in T, to represent the number of its applications
- for each control state, an equation

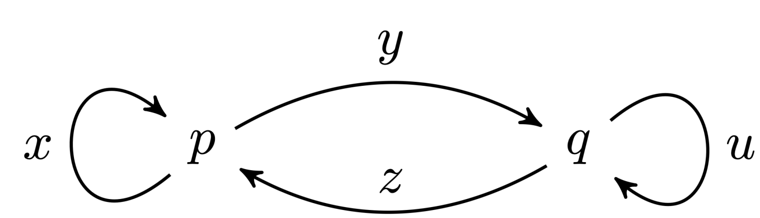
  nr of incoming transitions = nr of outgoing transitions

  except for p, q …
- d equations:

  total sum of effects = $w - v$

Example:

$$x + z + 1 = x + y$$
$$y + u = u + z + 1$$
$$-x + 2u = -1$$
$$x - u = 1$$
$$-z = -2$$



source $q(2, 0, 2)$,

target $p(1, 1, 0) \in Q \times \mathbb{N}^3$

State equation vs reachability

Fact: Characteristic equation has a solution in $\mathbb{N}$

if

$$q(v) \longrightarrow^* p(w)$$

run

configurations $\mathbb{N}^d$

Lemma: Characteristic equation has a **strongly connected** solution in $\mathbb{N}$

iff

$$q(v) \dashrightarrow^* p(w)$$

pseudo-run

pseudo-configurations $\mathbb{Z}^d$

Question: Does $q(v) \dashrightarrow^* p(w)$ imply $q(v) \longrightarrow^* p(w)$?

I. Intro

- reachability and coverability
- equivalent models
- coverability tree
- characteristic equation

II. Decidability

- **decomposition algorithm**
- perfectness: sufficient condition for reachability
- refinement

Reachability problem for VASS

reachability instance

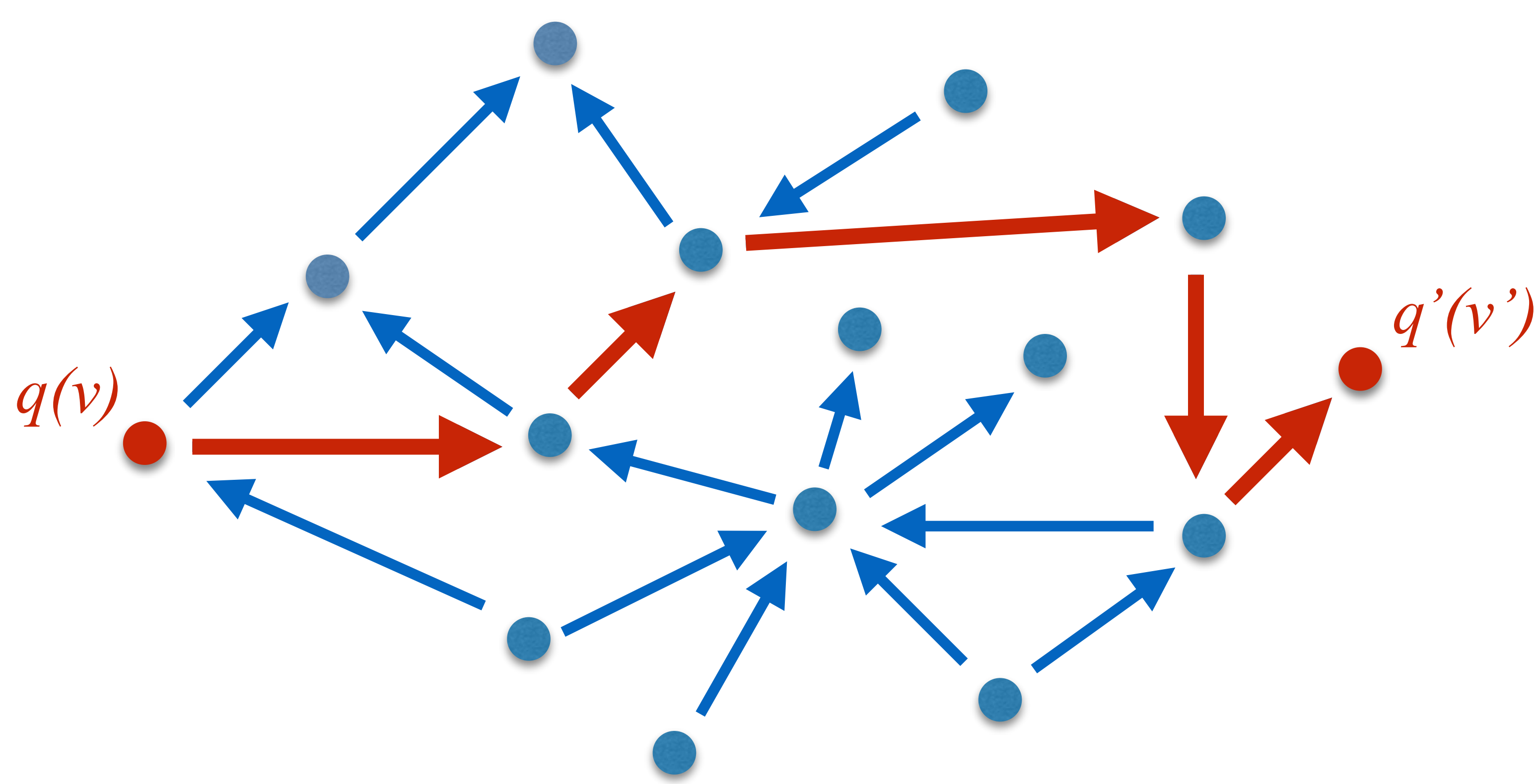
Given

- VASS
- source $q(v)$
- target $q'(v')$

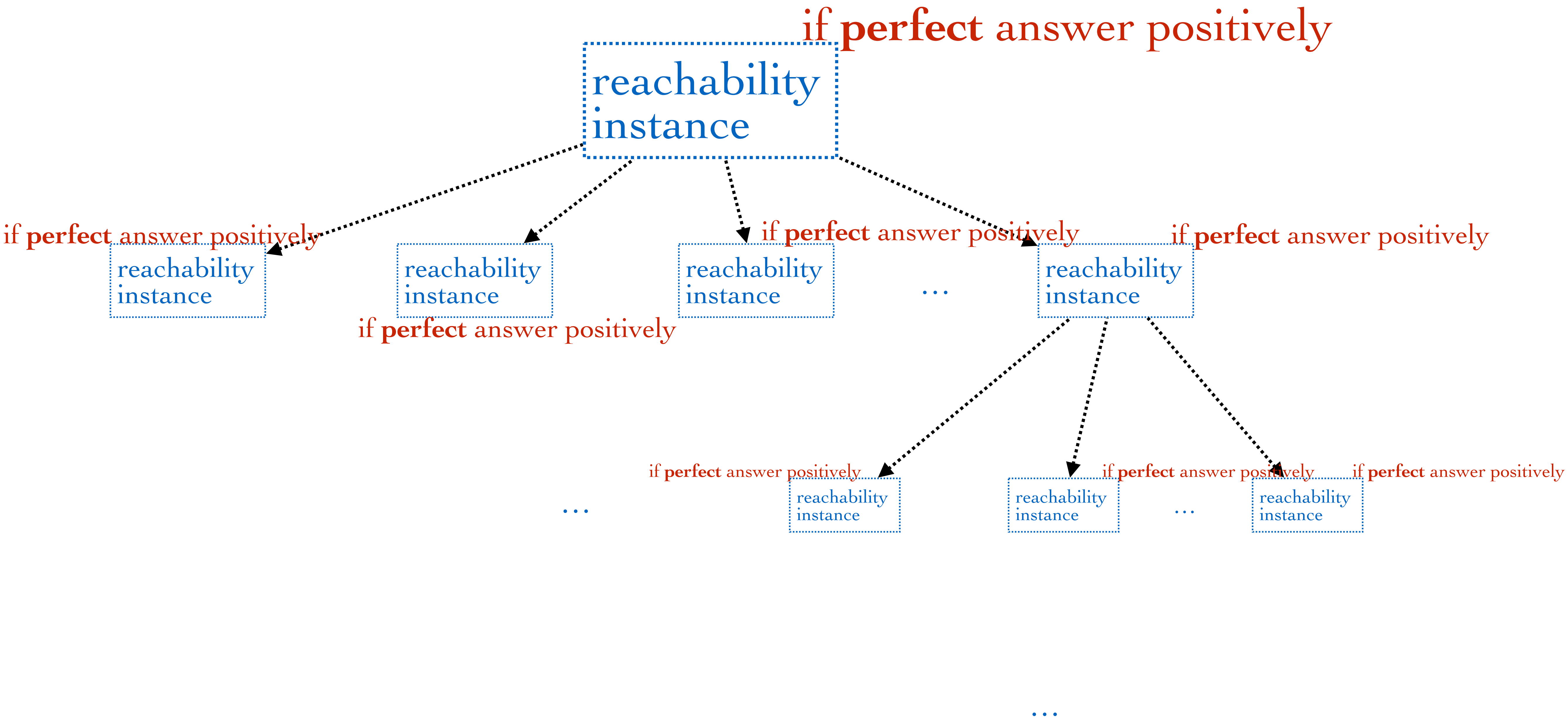
decide if $q(v) \longrightarrow^* q'(v')$

- dimension d
- finite set of control states Q
- finite set of transitions T of the form:

$$q \xrightarrow{a} p \qquad q, p \in Q \quad a \in \mathbb{Z}^d$$



Decomposition algorithm



II. Decidability

- decomposition algorithm
- **perfectness: sufficient condition for reachability**
- refinement

Perfectness: sufficient condition for reachability

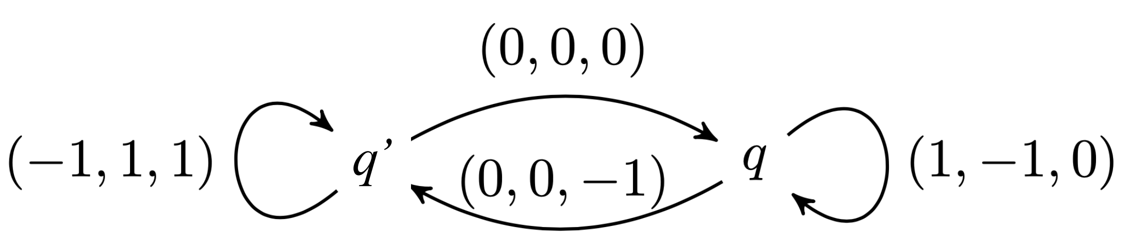
Question: Does $q(v) \dashrightarrow^{*} p(w)$ imply $q(v) \longrightarrow^{*} p(w)$?

Perfectness

($\boldsymbol{\Theta}_1$) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times unboundedness

($\boldsymbol{\Theta}_1$) $\Rightarrow$ VASS is strongly connected

Example:



source $q(2, 0, 2)$
target $q'(1, 1, 0)$

✔

Perfectness

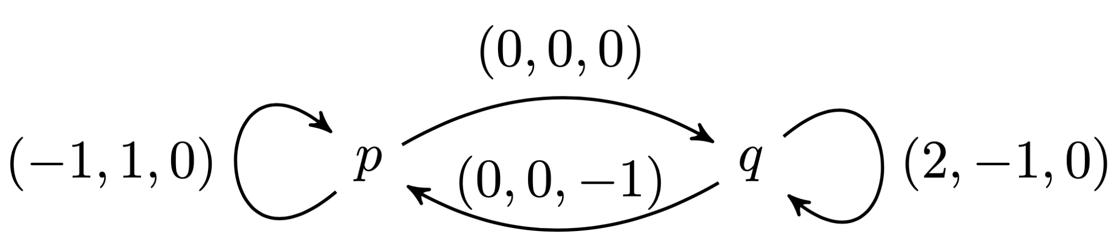
(Θ_1) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times — unboundedness

(Θ_2) For some $\Delta, \Delta' \geq 1$,

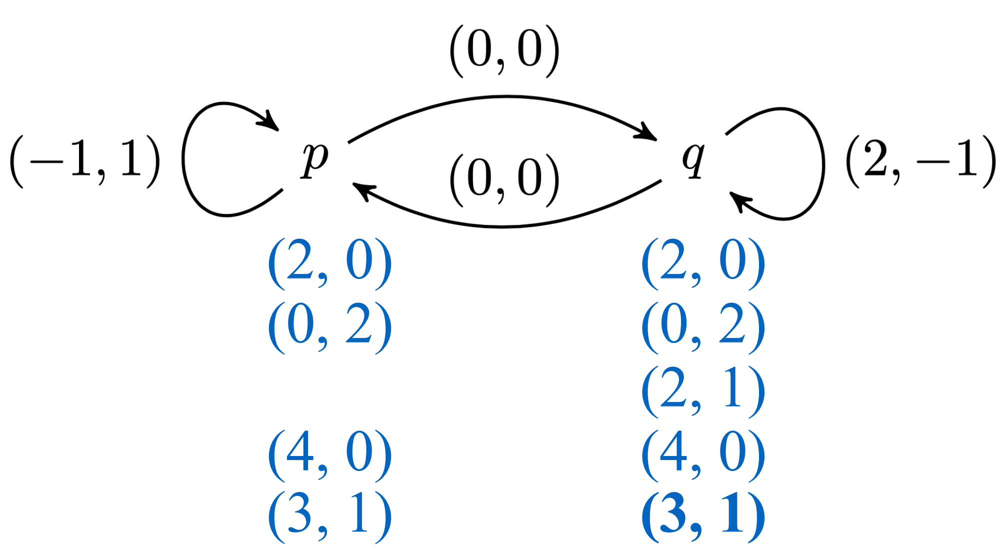
$$q(v) \longrightarrow^* q(v + \Delta)$$ — forward pumpability

$$q'(v' + \Delta') \longrightarrow^* q'(v')$$ — backward pumpability

Examples:



source $q(2, 0, 2)$ ✗



source $q(2, 0)$ ✓

Perfectness: sufficient condition for reachability

Lemma: $(\Theta_1) \wedge (\Theta_2) \Rightarrow q(v) \longrightarrow^* q'(v')$.

(Θ_1) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times

(Θ_2) For some $\Delta, \Delta' \geq \mathbf{1}$,

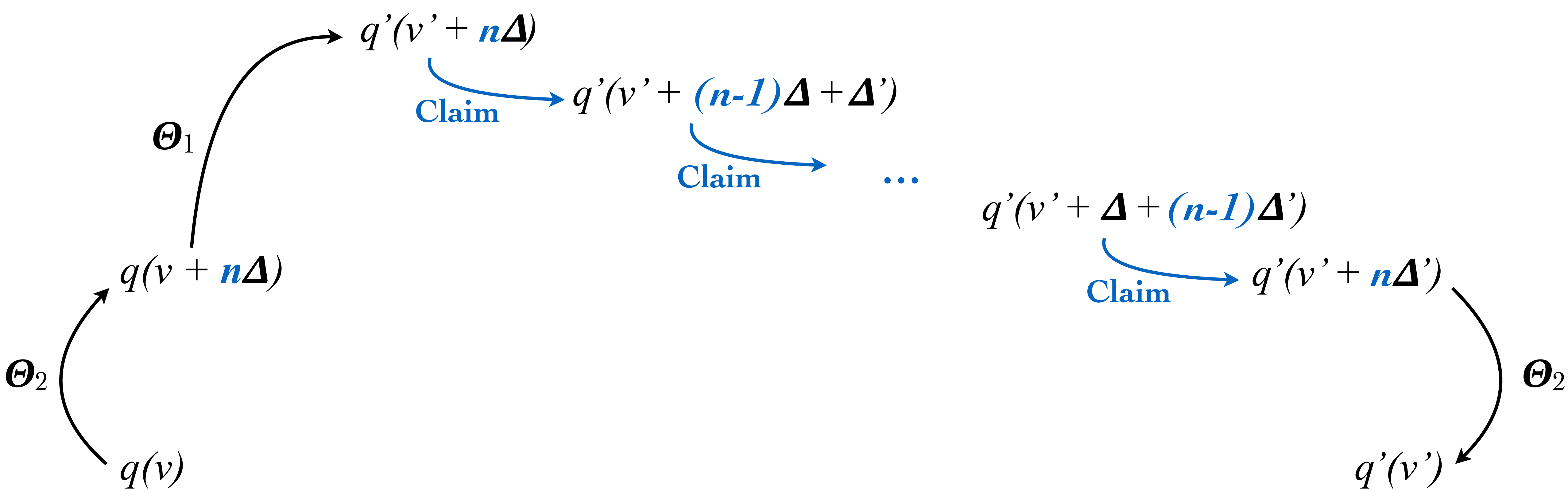
$$q(v) \longrightarrow^* q(v + \Delta)$$

$$q'(v' + \Delta') \longrightarrow^* q'(v')$$

Proof:

Choose sufficiently large n

Claim: $q'(\Delta) \dashrightarrow^* q'(\Delta')$.



(Θ_1) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times

(Θ_2) For some $\Delta, \Delta' \geq \mathbf{1}$,

Π: $q(v) \longrightarrow^* q(v + \Delta)$

Π': $q'(v' + \Delta') \longrightarrow^* q'(v')$

Claim: $q'(\Delta) \dashrightarrow^* q'(\Delta')$.

Proof:

Folding of a pseudo-run a: $F(a) \in \mathbb{N}^T$

Effect of a pseudo-run a: $E(a) \in \mathbb{Z}^d$

Observation: Given pseudo-runs $q(\_) \overset{\alpha}{\underset{\beta}{\dashrightarrow}} q'(\_)$ such that $F(\alpha) - F(\beta) \geq \mathbf{1}$,

there is a pseudo-run γ $q'(\_)$ such that $F(\gamma) = F(\alpha) - F(\beta)$

$(\Theta_1) \Rightarrow q(v) \overset{\alpha}{\underset{\beta}{\dashrightarrow}} q'(v')$ such that $F(\alpha) - F(\beta)$ arbitrarily large

$F(\alpha) - F(\beta) - F(\Pi) - F(\Pi') \geq \mathbf{1}$

$F(\alpha) - F(\Pi\,\beta\,\Pi') \geq \mathbf{1}$

By **Observation**, $q'(\_)$ γ such that $F(\gamma) = F(\alpha) - F(\beta) - F(\Pi) - F(\Pi')$

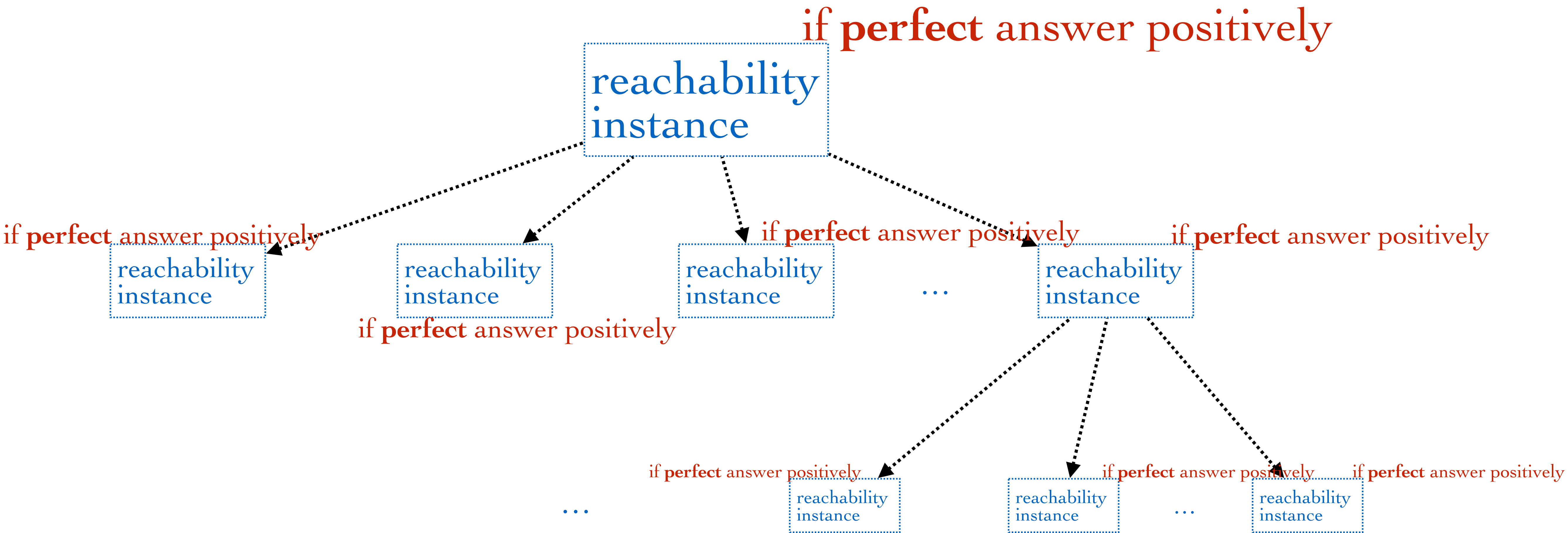
$E(\gamma) = E(\alpha) - E(\beta) - E(\Pi) - E(\Pi') =$

$\mathbf{0} \quad - \Delta \quad - (-\Delta') = \Delta' - \Delta$

II. Decidability

- decomposition algorithm
- perfectness: sufficient condition for reachability
- **refinement**

Decomposition algorithm



Question: Is $(\Theta_1) \wedge (\Theta_2)$ decidable?

Decidability of (Θ_1) ∧ (Θ_2)

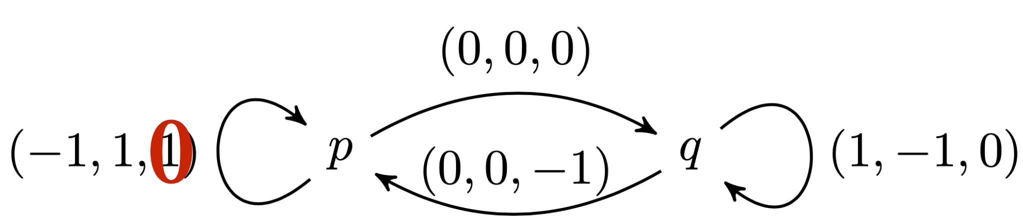
(Θ_1) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times

(Θ_2) For some $\Delta, \Delta' \geq \mathbf{1}$,
$q(v) \longrightarrow^* q(v + \Delta)$
$q'(v' + \Delta') \longrightarrow^* q'(v')$

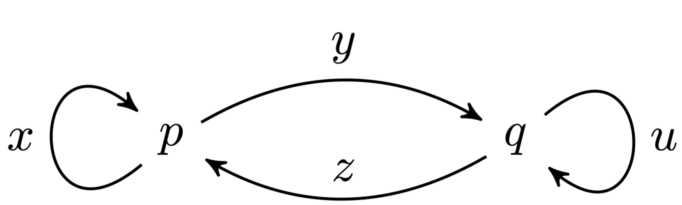
Question: How to decide (Θ_2) ?
Using coverability tree!

Question: How to decide (Θ_1) ?
Using characteristic equation!

Example:



source $q(2, 0, 2)$
target $p(1, 1, 0)$



	homogeneous system:	
$z - y = 1$	$z - y = 0$	✔
$x - u = 1$	$x - u = 0$	
z - ~~x~~ $= 2$	z - ~~x~~ $= 0$	✘

Refinement

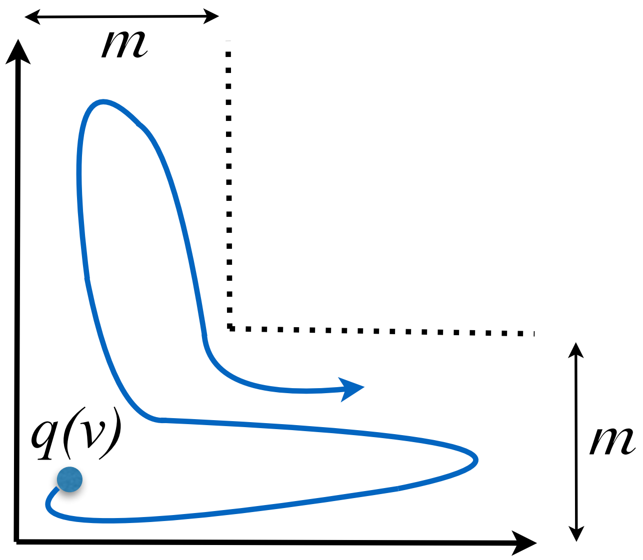
(Θ_1) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times

(Θ_2) For some $\boldsymbol{\Delta}, \boldsymbol{\Delta}' \geq \mathbf{1}$,
$q(v) \longrightarrow^* q(v + \boldsymbol{\Delta})$
$q'(v' + \boldsymbol{\Delta}') \longrightarrow^* q'(v')$

(Θ_2) fails:

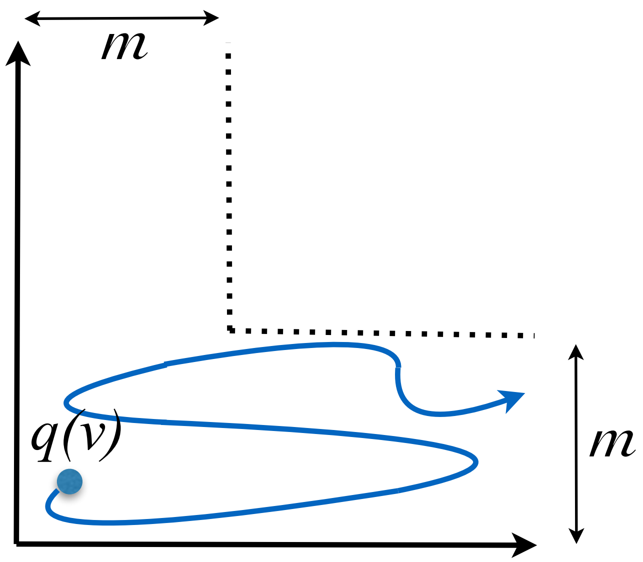
computable - how?

there exists m s.t. every configuration reachable from $q(v)$ has some coordinate $< m$



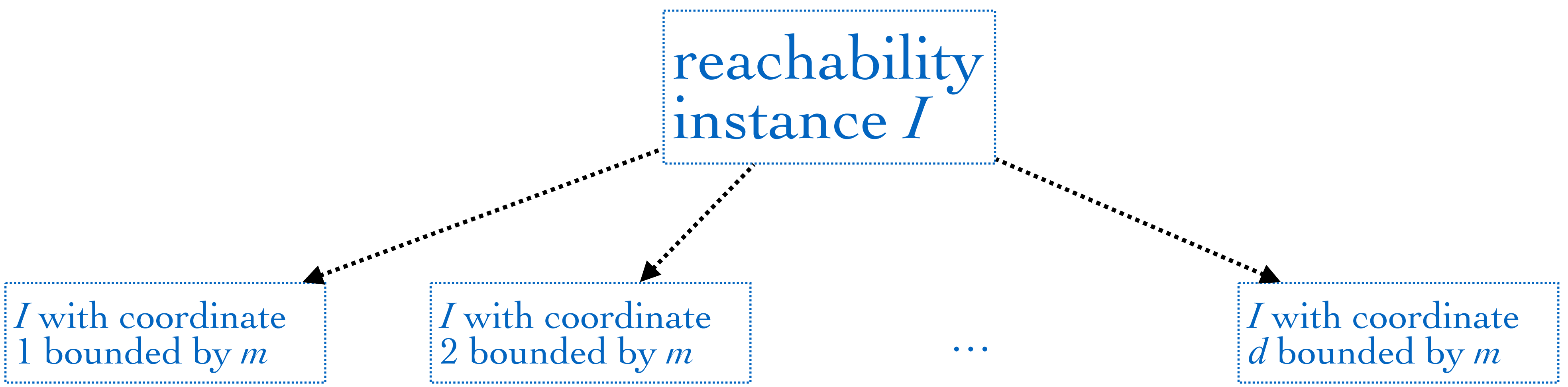
due to coverability tree

there exists m s.t. every run from $q(v)$ has some coordinate $< m$



Refinement

(Θ_2) fails: there exists m s.t. every run from $q(v)$ has some coordinate $< m$



Refinement

(Θ_1) For every m, $q(v) \dashrightarrow^* q'(v')$
using every transition $\geq m$ times

(Θ_2) For some $\Delta, \Delta' \geq \mathbf{1}$,
$q(v) \longrightarrow^* q(v + \Delta)$
$q'(v' + \Delta') \longrightarrow^* q'(v')$

(Θ_1) fails:

computable, using a bound on minimal solutions of state equation

there exists m s.t. every pseudo-run $q(v) \dashrightarrow^* q'(v')$ uses some transition $< m$ times

reachability instance I

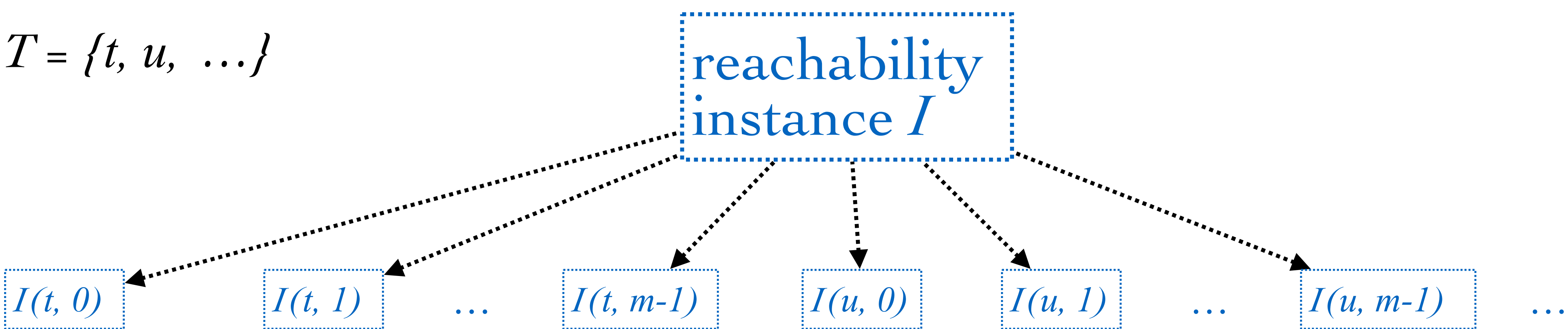
$t \in T$, $k < m$

$I(t, k)$:= $I - t \xrightarrow{t} I - t \xrightarrow{t} \ldots \xrightarrow{t} I - t$

(t appears k times)

some cheating here!

$T = \{t, u, \ldots\}$



are these instances smaller?

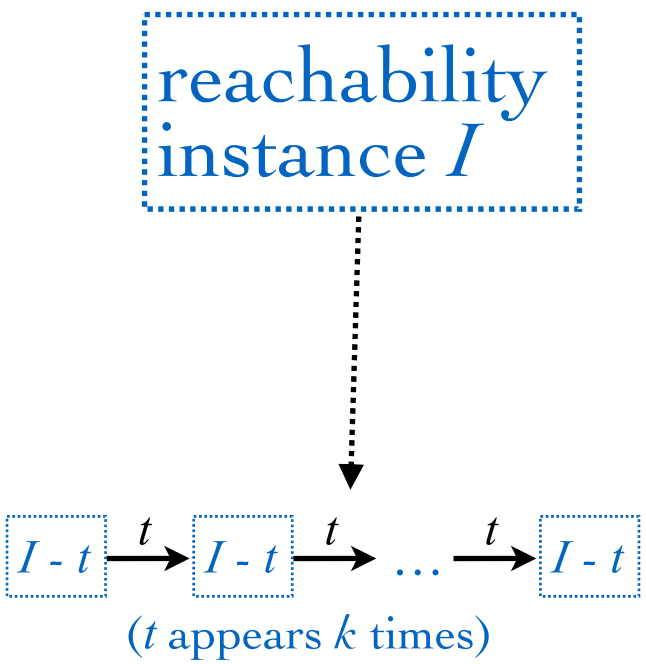
Refinement

($\boldsymbol{\Theta}_1$) For every m, $q(v) \dashrightarrow^* q'(v')$ using every transition $\geq m$ times

($\boldsymbol{\Theta}_2$) For some $\boldsymbol{\Delta}, \boldsymbol{\Delta}' \geq \mathbf{1}$,
$$q(v) \longrightarrow^* q(v + \boldsymbol{\Delta})$$
$$q'(v' + \boldsymbol{\Delta}') \longrightarrow^* q'(v')$$

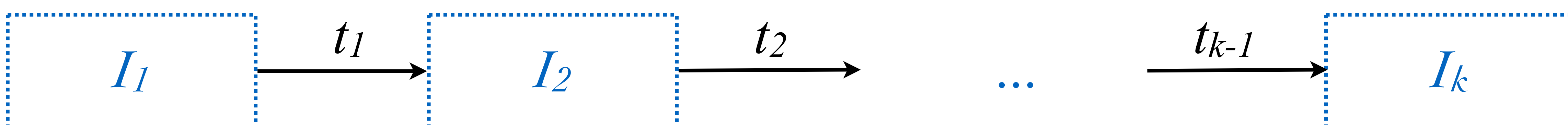
($\boldsymbol{\Theta}_1$) fails:

there exists m s.t. every pseudo-run $q(v) \dashrightarrow^* q'(v')$ uses some transition $< m$ times



is this instance smaller?

is this an instance at all?



I. Intro

- reachability and coverability
- equivalent models
- coverability tree
- state equation

II. Decidability

- decomposition algorithm
- perfectness: sufficient condition for reachability
- refinement

III. F_ω -hardness

Reachability problem for counter programs

Reachability problem: given a counter program **without zero tests,**

```
1: x' += 100
2: goto 5 or 3
3: x += 1   x' -= 1   y += 2
4: goto 2
5: halt if x' = 0.
```

can it halt? (successfully execute its halt command)

Coverability problem: given a counter program **without zero tests**
with trivial halt command,

```
1: x' += 100
2: goto 5 or 3
3: x += 1   x' -= 1   y += 2
4: goto 2
5: halt.
```

can it halt?

Loop programs

```
1: x' += 100
2: goto 5 or 3
3: x += 1     x' −= 1     y += 2
4: goto 2
5: halt if x' = 0.
```

```
1: x' += 100
2: loop
3:     x += 1     x' −= 1     y += 2
4: halt if x' = 0.
```

III. F_ω-hardness

- **reduction**
- multipliers and simulation of zero-tests
- amplifiers
- open questions

$\mathbf{F}_\omega$ -hardness of reachability

counter programming

counter program **with zero-tests** of size n → counter program **without zero-tests**

```
i += 1   x += 1   y += 1   b += 1   c += 1   d += 1
loop
    x += 1   y += 1   c += 1   d += 1
loop
    loop
        c −= i   c′ += 1
        loop at most b times
            x −= i   d −= i   x′ += i + 1
    loop
        b −= 1   b′ += i + 1
    loop
        b′ −= 1   b += 1
    loop
        c′ −= 1   c += 1
        loop at most b times
            x′ −= 1   x += 1   d += 1
    i += 1
zero? î
loop
    x −= i   y −= 1
halt if y = 0
```

P

$A_\omega(n)$ = $A_n(n)$

can it halt in $A_n(n)$ **steps**?
can it halt in $A_n(n)/2$ **steps**?
can it halt after $A_n(n)/2$ **zero-tests**?

```
x += 1   y += 1
loop
    x += 1   y += 1
for i := n down to 1 do
    loop
        x −= 1   z += 1
    loop
        x += i + 1   z −= i
loop
    x −= n + 1   y −= 1
halt if y = 0.
```

P’

can it halt?

P can halt after $A_n(n)/2$ **zero-tests** iff ***P’*** can halt

III. $\mathbf{F}_{\omega}$-hardness

- reduction
- **multipliers and simulation of zero-tests**
- amplifiers
- open questions

The set computed by a counter program

initial valuation: all counters 0

```
x += 1   y += 1
loop
    x += 1   y += 1
for i := n down to 1 do
    loop
        x −= 1   z += 1
    loop
        x += i + 1   z −= i
loop
    x −= n + 1   y −= 1
halt if y = 0.
```

consider all runs
(nondeterminism)

the set of all valuations at successful halt

B-multiplier

$B \in \mathbb{N}$ - fixed positive integer

initial valuation: all counters 0

1: x += 1 y += 1

```
1:  b += B    d += B    c += 1
2:  loop
3:        d += B    c += 1
```

11: **halt if** y = 0.

- ~~b = B~~ • b > 0 ?
- c > 0
- d = b · c
- all other counters 0

RATIO(b, c, d, B)

Hilbert's 10th problem!

One can compute **$A_n(n)$-multiplier** of size $O(n)$

F_ω -hardness of reachability

program of size n
with two **zero-tested** counters:

```
i += 1   x += 1   y += 1   b += 1   c += 1   d += 1
loop
   x += 1   y += 1   c += 1   d += 1
loop
   loop
      c −= i   c′ += i
      loop at most b times
         x −= i   d −= i   x += i+1
   loop
      b −= 1   b′ += i+1
   loop
      b′ −= 1   b += 1
   loop
      c′ −= 1   c += 1
      loop at most b times
         x′ −= 1   x += 1   d += 1
   i += 1
zero? î
loop
   x −= i   y −= 1
halt if y = 0
```

P

can halt after $A_n(n)/2$ **zero-tests**?

program **without** zero-tests:

```
x += 1   y += 1
loop
   x += 1   y += 1
for i := n down to 1 do
   loop
      x −= 1   z += 1
   loop
      x += i+1   z −= i
loop
   x −= n+1   y −= 1
halt if y = 0.
```

$A_n(n)$-multiplier

RATIO(b, c, d, $A_n(n)$)

```
i += 1   x += 1   y += 1   b += 1   c += 1   d += 1
loop
   x += 1   y += 1   c += 1   d += 1
loop
   loop
      c −= i   c′ += i
      loop at most b times
         x −= i   d −= i   x += i+1
   loop
      b −= 1   b′ += i+1
   loop
      b′ −= 1   b += 1
   loop
      c′ −= 1   c += 1
      loop at most b times
         x′ −= 1   x += 1   d += 1
   i += 1
zero? î
loop
   x −= i   y −= 1
halt if y = 0
```

P

instrumented using b, c, d

can halt?

Instrumentation - simulation of zero tests

- b = $A_n(n)$
- c > 0
- d = b · c
- x = y = 0 **zero-tested** counters

```
i += 1   x += 1   y += 1   b += 1   c += 1   d += 1
loop
    x += 1   y += 1   c += 1   d += 1
loop
    loop
        c −= i   c′ +=
        loop at most
            x −= i   d   x′ += i + 1
    loop
        b −= 1   b′   i + 1
    loop
        b′ −= 1   b += 1
    loop
        c′ −= 1   c += 1
        loop at most b times
            x′ −= 1   x += 1   d += 1
    i += 1
zero? î
loop
    x −= i   y −= 1
halt if y = 0
```

P

instrumented using b, c, d

- **instrument increments and decrements:**

command	replaced by	
x += 1	x += 1	c −= 1
x −= 1	x −= 1	c += 1

put x, y on **budget** c

c + x + y const

- **replace** **zero?** x **by**

```
ZERO? x:
loop
    y −= 1   x += 1   d −= 1
loop
    c −= 1   y += 1   d −= 1
loop
    y −= 1   c += 1   d −= 1
loop
    x −= 1   y += 1   d −= 1
b −= 2
```

- **replace** **halt** **by**

halt if . . . , d = 0.

halt of M

Aim:
simulate $A_n(n)/2$ **zero-tests** on x, y

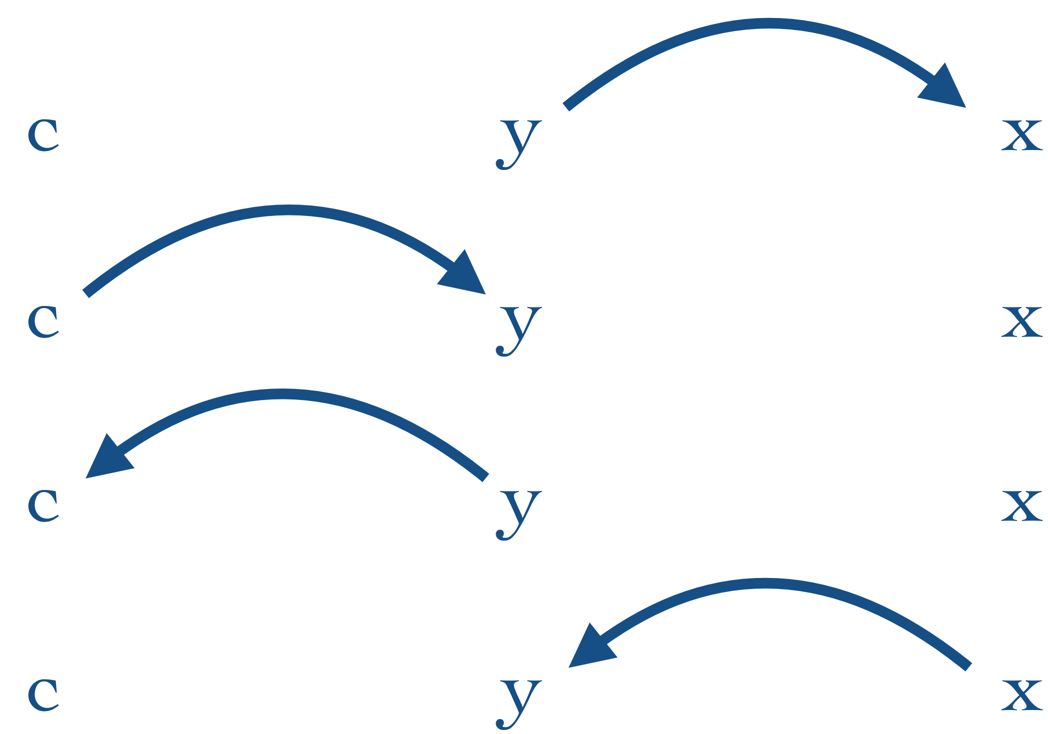
- simulation of zero tests

put x, y on **budget** c

$d = b \cdot \underbrace{(c + x + y)}_{\text{const}}$

ZERO? x:

```
loop
    y −= 1    x += 1    d −= 1
loop
    c −= 1    y += 1    d −= 1
loop
    y −= 1    c += 1    d −= 1
loop
    x −= 1    y += 1    d −= 1
b −= 2
```

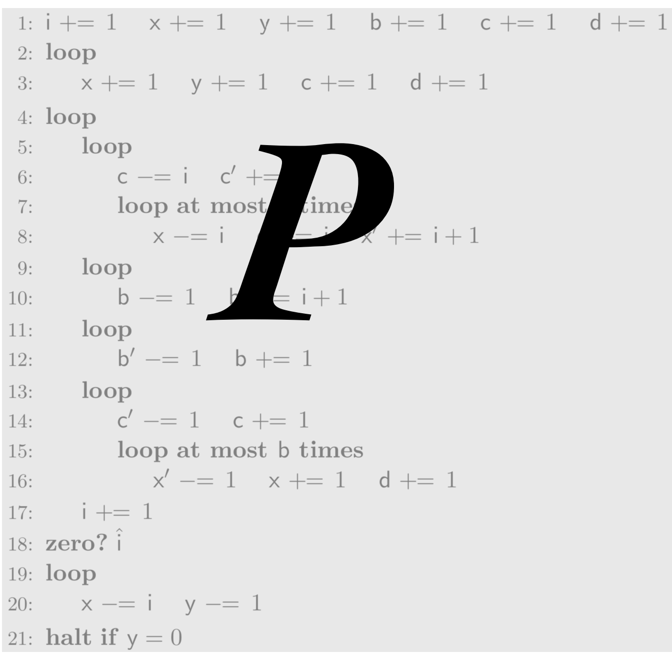


d decreases by <= $2 \cdot (c + x + y)$
b decreases by 2

- d decreases by $2 \cdot (c + x + y)$ ⟶ x = 0 initially and finally, y preserved
- d decreases by < $2 \cdot (c + x + y)$ ⟶ **halt if** . . . , $d = 0$. will surely fail

F_ω -hardness of reachability

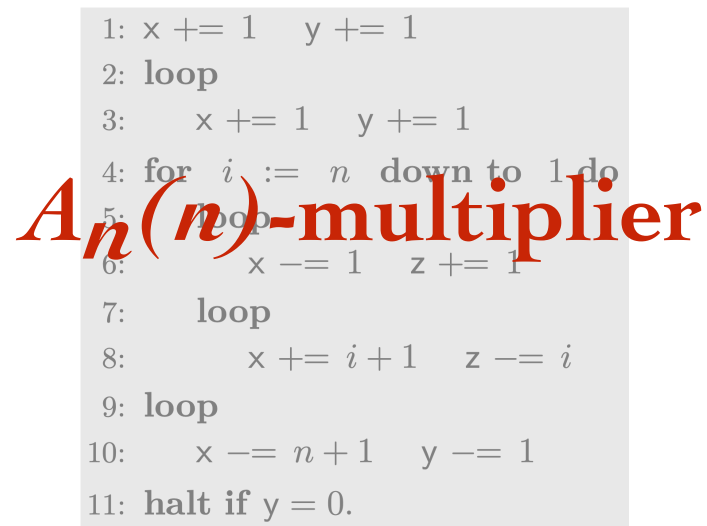
program of size n
with two **zero-tested** counters:



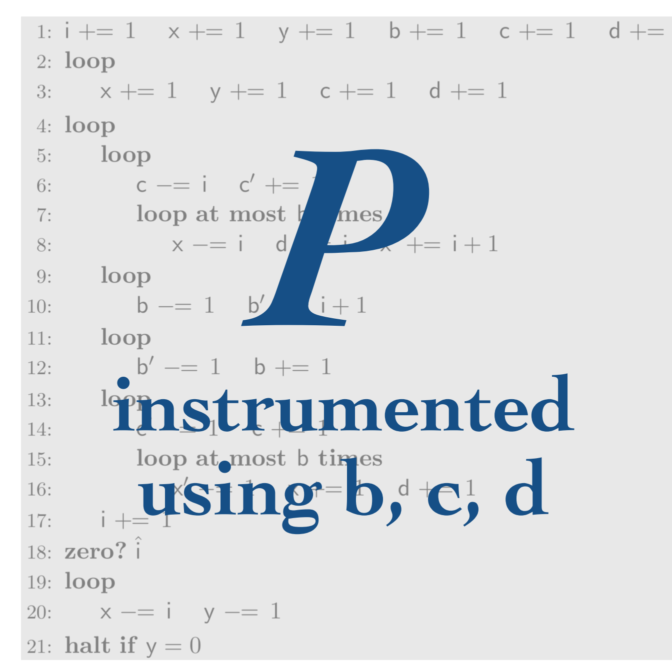
can halt after *$A_n(n)/2$* **zero-tests**?

One can compute
***$A_n(n)$*-multiplier**
of size $O(n)$

program **without** zero-tests:



RATIO(b, c, d, $A_n(n)$)



can halt?

III. $\mathbf{F}_\omega$-hardness

- reduction
- multipliers and simulation of zero-tests
- **amplifiers**
- open questions

$A_n(n)$-multiplier

$$A_1(n) = 2n$$

$$A_{i+1}(n) = \underbrace{A_i \circ A_i \circ \dots \circ A_i}_{n}(1) = A_i^{n}(1)$$

One can compute
$A_n(n)$-multiplier
of size $O(n)$

A_n-amplifier → $A_n(n)$-multiplier

...

...

A_2-amplifier

A_1-amplifier

The set computed by a counter program **from a set I**

a set I of initial valuations

initial valuation: all counters 0

```
x += 1   y += 1
loop
    x += 1   y += 1
for i := n down to 1 do
    loop
        x −= 1   z += 1
    loop
        x += i+1   z −= i
loop
    x −= n+1   y −= 1
halt if y = 0.
```

consider all runs **starting in I** (nondeterminism)

the set of all valuations at successful halt

F-amplifier

RATIO(b, c, d, B)
- b = B
- c > 0
- d = b · c
- all other counters 0

$F : \mathbb{N} \to \mathbb{N}$ - fixed function

For every fixed B:

RATIO(b, c, d, B)

```
x += 1    y += 1
loop
    x += 1    y += 1
for i := n down to 1 do
    loop
        x -= 1    z += 1
    loop
        x += i+1    z -= i
loop
    x -= n+1    y -= 1
halt if y = 0.
```

P **(b, c, d, b', c', d')**

halt if d=0

RATIO(b', c', d', $F(B)$)

consider all runs starting in RATIO(b, c, d, B) (nondeterminism)

A_n-amplifier ⟶ $A_n(n)$-multiplier

initial valuation: all counters 0

```
1: b += n    d += n    c += 1
2: loop
3:     d += n    c += 1
```

n-multiplier

A_n(n)-multiplier

RATIO(b, c, d, n)

```
x += 1    y += 1
loop
    x += 1    y += 1
for i := n down to 1 do
    loop
        x -= 1    z += 1
    loop
        x += i+1    z -= i
loop
    x -= n+1    y -= 1
halt if y = 0.
```

A_n-amplifier P(b, c, d, b’, c’, d’)

RATIO(b’, c’, d’, $A_n(n)$)

A_n-amplifier

$$A_1(n) = 2n$$

$$A_{k+1}(n) = \underbrace{A_k \circ A_k \circ \ldots \circ A_k}_{n/4}(4) = A_k^{n/4}(4)$$

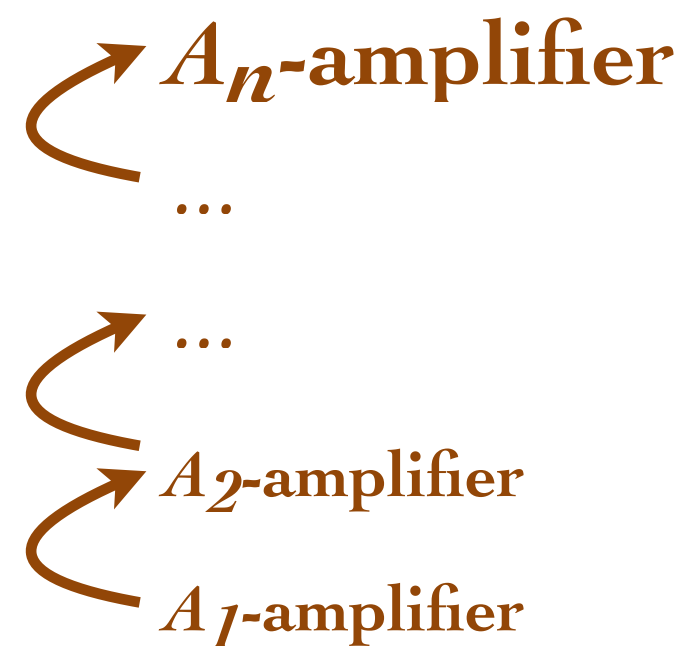
One can compute **A_n-amplifier P(b, c, d, b', c', d')** with $3n+2$ counters, of size $O(n)$

- **A_1-amplifier:**

```
loop
    loop
        c −= 1    c′ += 1    d −= 1    d′ += 2
    loop
        c′ −= 1    c += 1    d −= 1    d′ += 2
    b −= 2    b′ += 4
loop
    c −= 1    c′ += 1    d −= 2    d′ += 4
b −= 2    b′ += 4
```

- amplifier lifting:

A_k-amplifier ⟶ **A_{k+1}-amplifier**



Amplifier lifting

$$A_{k+1}(n) = \underbrace{A_k \circ A_k \circ \ldots \circ A_k}_{n/4}(4) = A_k^{n/4}(4)$$

$\mathcal{M}$ 4-multiplier

RATIO(b_1, c_1, d_1, 4)

RATIO(b_1, c_1, d_1, B)

$\mathcal{P}$ A_k-amplifier

RATIO(b_2, c_2, d_2, $A_k(B)$)

RATIO(b_2, c_2, d_2, B)

$\mathcal{L}$ identity-amplifier

RATIO(b_1, c_1, d_1, B)

A_{k+1}-amplifier

RATIO(b, c, d, B)

```
M
loop
    P
    zero? d1
    L
    zero? d2
P
zero? d1
```

RATIO(b_2, c_2, d_2, $A_{k+1}(B)$)

```
M
loop
    P
    zero? d1
    L
    zero? d2
P
zero? d1
```

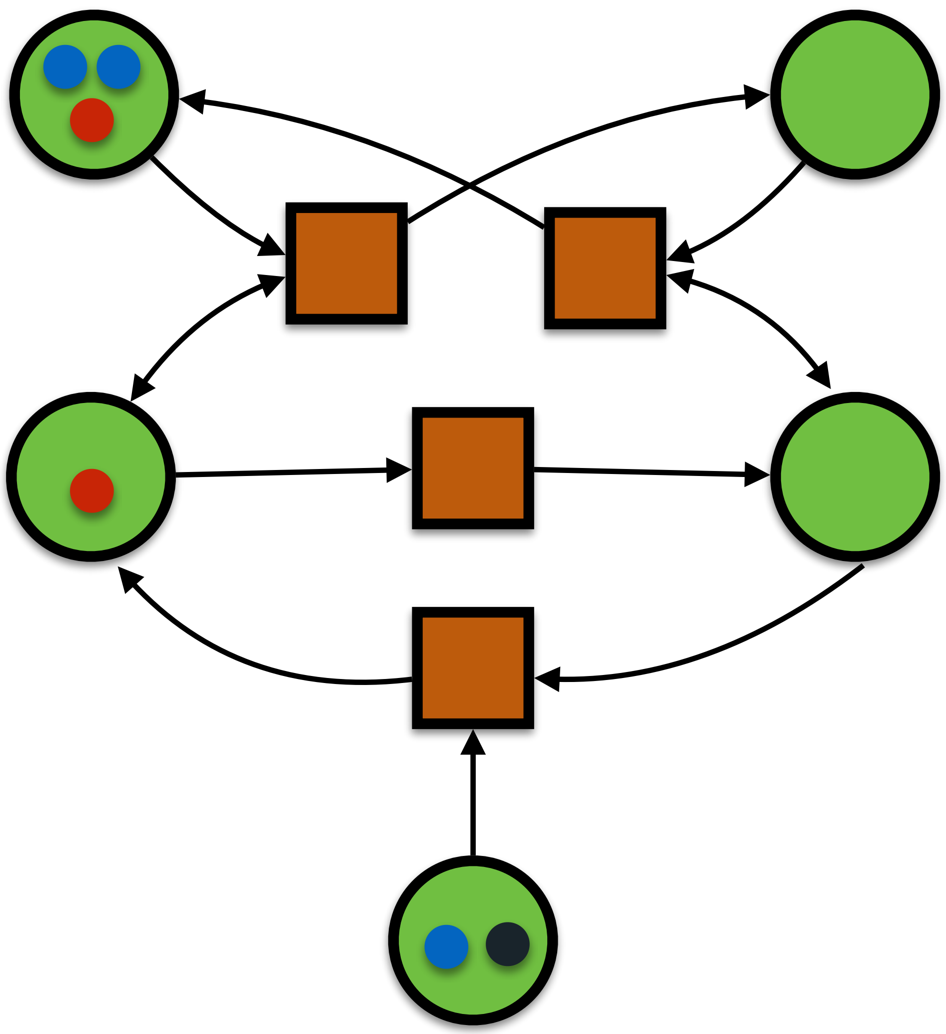
instrumented using b, c, d

III. $\mathfrak{F}_\omega$-hardness

- reduction
- multipliers and simulation of zero-tests
- amplifiers
- **open questions**

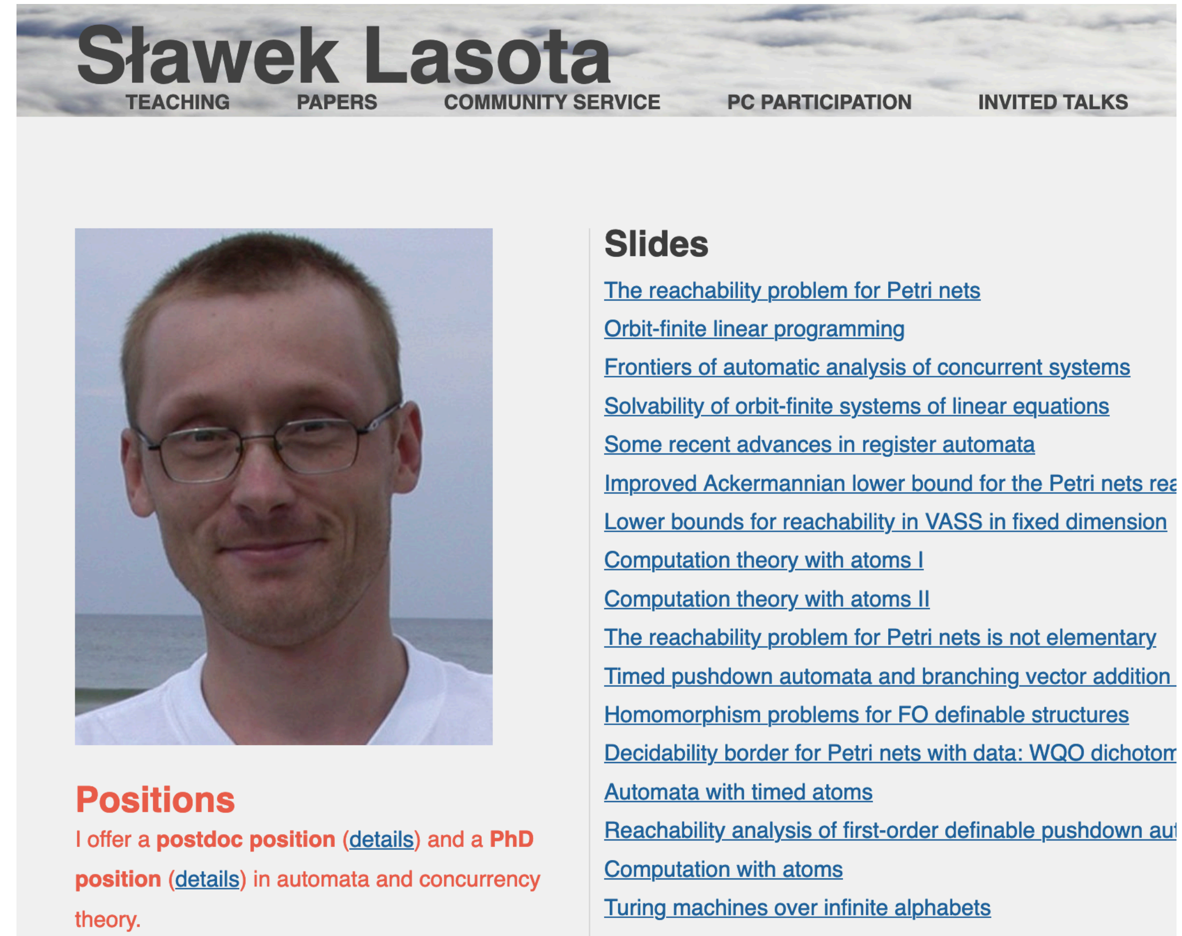
Open questions

- dimension-parametric complexity: $\mathbf{F}_k$-hardness for which dimension?
- small fixed dimension
- extensions:
  - data Petri nets
  - pushdown Petri nets
  - branching Petri nets



I. Intro

- reachability and coverability
- equivalent models
- coverability tree
- characteristic equation



II. Decidability

- decomposition algorithm
- perfectness: sufficient condition for reachability
- refinement

III. $\mathcal{F}_\omega$-hardness

- reduction
- multipliers and simulation of zero-tests
- amplifiers
- open questions

thank you!