The reachability problem for Petri nets

Sławomir Lasota
University of Warsaw

ACPN 2023, Toruń, 2023-09-05
I. Intro
II. Decidability
III. $F_\omega$-hardness
I. Intro

• reachability and coverability
• equivalent models
• coverability tree
• characteristic equation
Reachability problem in Petri nets

Coverability

Petri net:

Decision problem:

given

- Petri net
- source configuration
- target configuration

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

places and transitions

configuration: places → \( \mathbb{N} \)  \( \mathbb{N}^d \)

step relation between configurations
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
decidability of coverability [Karp, Miller JCSS]

1976 — EXPSPACE lower bound [Lipton TR Yale U]
1977 — (incomplete) decidability of reachability [Sacerdote, Tenney STOC]
1978 — EXPSPACE algorithm for coverability [Rackoff TCS]
1981 — decidability of reachability [Mayr STOC]
1982 — decidability of reachability - simplified proof [Kosaraju STOC]

Part II

huge complexity gap!

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

2011 — decidability of reachability by Presburger invariants [Leroux POPL]
2015 — upper bound $F_{\omega^3}$ [Leroux, Schmitz LICS]
2019 — Ackermannian upper bound $F_\omega$ [Leroux, Schmitz LICS]
2019—2020 — TOWER lower bound $F_3$ [Czerwiński, L., Lazic, Leroux, Mazowiecki STOC]
2019 — Ackermannian upper bound $F_\omega$ [Leroux, Schmitz LICS]

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

$$2^2^2 \ldots 2^n$$

2020

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

$$2^2^2 \ldots 2^n$$

2021 — gap closed!

2021 — Ackermannian lower bound $F_\omega$ [Czerwiński, Orlikowski FOCS] [Leroux FOCS]

2021 — improved and simplified Ackermannian lower bound $F_\omega$ [L. STACS]

Part III
Fast growing functions and induced complexity classes

\[ A_1(n) = 2n \]

\[ A_{i+1}(n) = A_i \circ A_i \circ \ldots \circ A_i(1) = A_i^n(1) \]

\[ A_\omega(n) = A_n(n) \quad \text{Ackermann function} \]

\[ F_i = \bigcup_{j_1 \ldots j_m < i} \text{DTIME}(A_i \circ A_{j_1} \circ \ldots \circ A_{j_m}) \]

\[ A_2(n) = 2^n \]

\[ A_3(n) = \text{tower}(n) = 2^{2^{2^{\ldots^2}}} \]

\[ A_4(n) = \ldots \]

\[ F_2 = \text{DTIME}(2^{O(n)}) \]

\[ F_3 = \text{TOWER} \]

\[ \ldots \]

\[ F_\omega = \text{ACKERMANN} \]
I. Intro

• reachability and coverability
• equivalent models
• coverability tree
• characteristic equation
Many faces of Petri nets

• Petri nets:

• vector addition systems with states (VASS):

\[ (-1,1,0) \xrightarrow{p} (0,0,0) \xrightarrow{q} (1,-1,0) \]

• counter programs without zero-tests:

1: loop
2: loop
3: \( x \leftarrow 1 \quad y \leftarrow 1 \)
4: loop
5: \( x \leftarrow 1 \quad y \leftarrow 1 \)
6: \( z \leftarrow 1 \)

• 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:

  $$a \xleftarrow{\text{effect of transition}} q \rightarrow p \quad q, p \in Q \quad a \in \mathbb{Z}^d$$

- configurations $(q, v) = q(v) \in Q \times \mathbb{N}^d$
- step relation:
  $$q(v) \rightarrow p(v+a)$$
- reachability relation:
  $$q(v) \rightarrow^* p(w)$$

two different graphs!
Petri nets $\Leftrightarrow$ VASS

- Petri nets:

\[
\begin{array}{c}
\text{x} \\
\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\
\text{p} & \text{q} & \text{z} & \text{y} & \text{z} & \text{y} \\
\end{array}
\]

- Vector addition systems with states (VASS):

\[
\begin{array}{c}
(0, 0, 0) \quad (0, 0, 1) \\
(-1, 1, 0) \quad p \quad (0, 0, -1) \quad q \quad (1, -1, 0) \\
\end{array}
\]

split every transition

\[
\begin{array}{c}
\text{x} \\
\downarrow \quad \downarrow \\
\text{p} & \text{q} & \text{y} \\
\end{array}
\]

into input and output:

\[
\begin{array}{c}
\text{x} \\
\downarrow \quad \downarrow \\
\text{p} & \text{q} \quad \text{y} \\
\end{array}
\]

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:

- \( x += n \) (increment counter \( x \) by \( n \))
- \( x -= n \) (decrement counter \( x \) by \( n \))
- \( \text{goto} \ L \text{ or } L' \) (jump to either line \( L \) or line \( L' \))

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

- \( \text{halt if} \ x_1, \ldots, x_l = 0 \) (terminate provided all the listed counters are zero)

Abort if \( x < n \)

Nondeterminism

**Example:**

1: \( x' += 100 \)
2: \( \text{goto} \ 5 \text{ or } 3 \)
3: \( x += 1 \) \( x' -= 1 \) \( y += 2 \)
4: \( \text{goto} \ 2 \)
5: \( \text{halt if} \ x' = 0. \)

Initially: \( x' = x = y = 0 \)

Finally: \( x' = 0 \) \( x = 100 \) \( y = 200 \)
Counter programs $\rightarrow$ VASS

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

**vector addition systems with states (VASS):**

\[
\begin{align*}
(-1, 1, 0) &\xrightarrow{\circlearrowleft} (0, 0, 0) \\
(0, 0, -1) &\xrightarrow{\circlearrowleft} (1, -1, 0)
\end{align*}
\]

**counter programs without zero-tests:**

1: **loop**
2: **loop**
3: $x \leftarrow= 1 \quad y \leftarrow= 1$
4: **loop**
5: $x \leftarrow= 1 \quad y \rightarrow= 1$
6: $z \rightarrow= 1$
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:

source
coverability tree:

if $=$ then stop generating the tree

if $<$ then replace increased coordinates by $\omega$

$\omega$ domination $(2, 1, 6, 3) \leq (2, 3, 7, 3)$

increased

Petri nets or VASS

(2, $\omega$, $\omega$, 3)
Dickson’s Lemma: every infinite sequence of configurations

\[
1 \leq 2 \leq 3 \ldots
\]

admits a domination:

\[
i \leq j \quad \text{for some } i < j.
\]
**Theorem:** Coverability tree is finite.

Coverable configurations = (coverability tree) ↓

**Question:** What can be read out from coverability tree?
I. Intro

• reachability and coverability
• equivalent models
• coverability tree
• characteristic equation
Characteristic equation

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

  $$ a \leftarrow \text{effect of transition} $$

  \[ q \rightarrow p \]

  \[ q, p \in Q \quad a \in \mathbb{Z}^d \]

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

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

  \[ \text{nr of incoming transitions} = \text{nr of outgoing transitions} \]

  except for \( p, q \) …

**Example:**

\[ x + z + 1 = x + y \]
\[ y + u = u + z + 1 \]
• source $q(v)$, target $p(w) \in Q \times \mathbb{N}^d$ for $q, p$ distinct


• one variable per transition in $T$, to represent the number of its applications

• for each control state, an equation

\[
\text{nr of incoming transitions} = \text{nr of outgoing transitions}
\]

except for $p, q$ …

• $d$ equations:

\[
\text{total sum of effects} = w - v
\]

Example:

\[
\begin{align*}
x + z + 1 & = x + y \\
y + u & = u + z + 1 \\
-x + 2u & = -1 \\
x - u & = 1 \\
-z & = -2
\end{align*}
\]
State equation vs reachability

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

$q(v) \xrightarrow{*} p(w)$

run configurations $\mathbb{N}^d$

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

$q(v) \xrightarrow{\text{pseudo}}* p(w)$

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

**Question:** Does $q(v) \xrightarrow{\text{pseudo}}* p(w)$ imply $q(v) \xrightarrow{*} p(w)$?
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

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

decide if $q(v) \rightarrow^{*} q'(v')$
Decomposition algorithm

if \textit{perfect} answer positively

reachability instance

...
II. Decidability

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

**Question:** Does $q(v) \rightarrow p(w)$ imply $q(v) \rightarrow p(w)$?
Perfectness

\[(\Theta_1) \text{ For every } m, \ q(v) \overset{*}{\rightarrow} q'(v') \text{ using every transition } \geq m \text{ times unboundedness} \]

\[(\Theta_1) \implies \text{VASS is strongly connected} \]

Example:

\[
\begin{array}{c}
(-1,1,1) \xrightarrow{q'} (0,0,-1) \xrightarrow{q} (1,-1,0) \\
\end{array}
\]

source \(q(2,0,2)\)

target \(q'(1,1,0)\)
Perfectness

(\(\Theta_1\)) For every \(m\), \(q(v) \xrightarrow{\ast} q'(v')\) using every transition \(\geq m\) times unboundedness

(\(\Theta_2\)) For some \(\Delta, \Delta' \geq 1\),

\[
\begin{align*}
q(v) & \xrightarrow{\ast} q(v + \Delta) & \text{forward pumpability} \\
q'(v' + \Delta') & \xrightarrow{\ast} q'(v') & \text{backward pumpability}
\end{align*}
\]

Examples:

\((-1, 1, 0)\) \(\xrightarrow{p} (0, 0, 0)\) \(\xrightarrow{q} (2, -1, 0)\) source \(q(2, 0, 2)\) \(\times\)

\((-1, 1)\) \(\xrightarrow{p} (0, 0)\) \(\xrightarrow{q} (2, -1)\) source \(q(2, 0)\) \(\checkmark\)
Perfectness: sufficient condition for reachability

**Lemma:** \((\Theta_1) \land (\Theta_2) \implies q(v) \xrightarrow{*} q'(v').

**Proof:**

Choose sufficiently large \(n\)

\[
\Theta_1 \quad q'(v' + n\Delta') \quad \Theta_2
\]

\[
\Theta_1 \quad q'(v' + (n-1)\Delta + \Delta') \quad \Theta_2
\]

\[
\Theta_1 \quad q'(v' + \Delta + (n-1)\Delta') \quad \Theta_2
\]

\[
\Theta_1 \quad q'(v' + n\Delta) \quad \Theta_2
\]

\[
\Theta_1 \quad q(v) \quad \Theta_2
\]

**Claim:** \(q'(\Delta) \xrightarrow{*} q'(\Delta').

(\(\Theta_1\)) For every \(m\), \(q(v) \xrightarrow{*} q'(v')\) using every transition \(\geq m\) times

(\(\Theta_2\)) For some \(\Delta, \Delta' \geq 1\),

- \(q(v) \xrightarrow{*} q(v + \Delta)\)
- \(q'(v' + \Delta') \xrightarrow{*} q'(v')\)
Claim: \( q'(\Delta) \rightarrow^* q'(\Delta') \).

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 \( q(\_ ) \xrightarrow{\beta} q'(\_ ) \) such that \( F(\alpha) - F(\beta) \geq 1 \), there is a pseudo-run \( \gamma \xrightarrow{\gamma} q'(\_ ) \) such that \( F(\gamma) = F(\alpha) - F(\beta) \)

\((\Theta_1)\) \rightarrow \( q(v) \xrightarrow{\beta} q'(v') \) such that \( F(\alpha) - F(\beta) \) arbitrarily large

\[ F(\alpha) - F(\beta) - F(\Pi) - F(\Pi') \geq 1 \]

\[ F(\alpha) - F(\Pi \beta \Pi') \geq 1 \]

By Observation, \( q'(\_ ) \xrightarrow{\gamma} q'(\_ ) \) such that \( F(\gamma) = F(\alpha) - F(\beta) - F(\Pi) - F(\Pi') \)

\[ E(\gamma) = E(\alpha) - E(\beta) - E(\Pi) - E(\Pi') = 0 - \Delta - (-\Delta') = \Delta' - \Delta \]

\((\Theta_2)\) For every \( m \), \( q(v) \rightarrow^* q'(v') \) using every transition \( \geq m \) times

\( \Pi : q(v) \rightarrow^* q(v + \Delta) \)

\( \Pi' : q'(v + \Delta') \rightarrow^* q'(v') \)
II. Decidability

- decomposition algorithm
- perfectness: sufficient condition for reachability
- refinement
**Question:** Is $(\Theta_1) \land (\Theta_2)$ decidable?
Decidability of \((\Theta_1) \land (\Theta_2)\)

**Question:** How to decide \((\Theta_2)\)?
- Using coverability tree!

**Question:** How to decide \((\Theta_1)\)?
- Using characteristic equation!

**Example:**

\[
\begin{align*}
(0, 0, 0) & \quad (0, 0, -1) & \quad q & \quad (1, -1, 0) \\
(1, 1, 0) & \quad p & \quad \langle 0, 0, 1 \rangle
\end{align*}
\]

source \(q(2, 0, 2)\)

target \(p(1, 1, 0)\)

**Homogeneous system:**

\[
\begin{align*}
z - y &= 1 & z - y &= 0 \\
x - u &= 1 & x - u &= 0 \\
z &= 2 & z &= 0
\end{align*}
\]
Refinement

(Θ₂) 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 \(< m\)

(Θ₁) For every \( m \), \( q(v) \rightarrow^* q'(v') \)
using every transition \( \geq m \) times

(Θ₂) For some \( \Delta, \Delta' \geq 1 \),
\[ q(v) \rightarrow^* q(v + \Delta) \]
\[ q'(v' + \Delta') \rightarrow^* q'(v') \]
Refinement

$(\Theta_2)$ fails: there exists $m$ s.t. every run from $q(v)$ has some coordinate $< m$
Refinement

\((\Theta_1)\) fails:

computable, using a bound on minimal solutions of state equation

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

\(\Theta_1\): For every \(m\), \(q(v) \rightarrow^* q'(v')\) using every transition \(\geq m\) times

\(\Theta_2\): For some \(\Delta, \Delta' \geq 1\),
\(q(v) \rightarrow^* q(v + \Delta)\)
\(q'(v' + \Delta') \rightarrow^* q'(v')\)

reachability instance \(I\)

\(I(t, k) := \begin{array}{c}
I - t \\
\rightarrow t
\end{array} \cdots \begin{array}{c}
I - t
\rightarrow t
\end{array} \begin{array}{c}
I - t
\rightarrow t
\end{array} \begin{array}{c}
I - t
\rightarrow t
\end{array}

(t appears \(k\) times)

\(T = \{t, u, \ldots\}\)

are these instances smaller?
(Θ₁) fails:

there exists \( m \) s.t. every pseudo-run \( q(v) \rightarrow^* q'(v') \)

uses some transition \( < m \) times

---

(Θ₁) For every \( m, q(v) \rightarrow^* q'(v') \) using every transition \( \geq m \) times

(Θ₂) For some \( \Delta, \Delta' \geq 1, q(v) \rightarrow^* q(v + \Delta) \)

\( q'(v' + \Delta') \rightarrow^* q'(v') \)
I. Intro

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

II. Decidability

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

III. $F_\omega$ -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: \texttt{goto 5 or 3}
3: \( x \) += 1 \( x' \) -= 1 \( y \) += 2
4: \texttt{goto 2}
5: \texttt{halt if} \( x' = 0 \).

1: \( x' \) += 100
2: \texttt{loop}
3: \( x \) += 1 \( x' \) -= 1 \( y \) += 2
4: \texttt{halt if} \( x' = 0 \).
III. $F_\omega$-hardness

- reduction
- multipliers and simulation of zero-tests
- amplifiers
- open questions
$F_\omega$ - hardness of reachability

counter program with zero-tests of size $n$

\[
\begin{align*}
1: & \quad i := 1 \quad x := 1 \quad y := 1 \quad b := 1 \quad c := 1 \quad d := 1 \\
2: & \quad \text{loop} \\
3: & \quad x := 1 \quad y := 1 \quad c := 1 \quad d := 1 \\
4: & \quad \text{loop} \\
5: & \quad c := i \quad c' := 1 \\
6: & \quad \text{loop at most } b \text{ times} \\
7: & \quad x := i \quad y := i \quad c := 1 \quad d := 1 \\
8: & \quad \text{loop} \\
9: & \quad b := 1 \quad y' := 1 \\
10: & \quad \text{loop} \\
11: & \quad y' := 1 \\
12: & \quad \text{loop} \\
13: & \quad c := 1 \quad c := 1 \\
14: & \quad \text{loop at most } b \text{ times} \\
15: & \quad x := i \quad d := 1 \\
16: & \quad \text{halt if } y' = 0 \\
17: & \quad i := 1 \\
18: & \quad \text{zero?} \\
19: & \quad \text{loop} \\
20: & \quad x := i \quad y := 1 \\
21: & \quad \text{halt if } y = 0 \\
\end{align*}
\]

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

counter program without zero-tests

\[
\begin{align*}
1: & \quad x := 1 \quad y := 1 \\
2: & \quad \text{loop} \\
3: & \quad x := 1 \quad y := 1 \\
4: & \quad \text{for } i := n \text{ down to 1 do} \\
5: & \quad \text{loop} \\
6: & \quad x := i \quad z := 1 \\
7: & \quad \text{loop} \\
8: & \quad x := i + 1 \quad z := i \\
9: & \quad \text{loop} \\
10: & \quad x := n + 1 \quad y := 1 \\
11: & \quad \text{halt if } y = 0. \\
\end{align*}
\]

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?

can it halt?

$P$ can halt after $A_n(n)/2$ zero-tests iff $P'$ can halt
III. $F_\omega$-hardness

- reduction
- multipliers and simulation of zero-tests
- amplifiers
- open questions
The set computed by a counter program

**initial valuation:** all counters 0

1: `x += 1`  `y += 1`
2: `loop`
3: `x += 1`  `y += 1`
4: `for i := n`  `down to`  `1`  `do`
5: `loop`
6: `x -= 1`  `z += 1`
7: `loop`
8: `x += i + 1`  `z -= i`
9: `loop`
10: `x -= n + 1`  `y -= 1`
11: `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

\begin{align*}
1: & \quad b \gets B \\
2: & \quad \text{loop} \\
3: & \quad d \gets B \\
\end{align*}

- \( b \geq B \)
- \( b > 0 \) ?
- \( c > 0 \)
- \( d = b \cdot c \)
- all other counters 0

\( \text{Hilbert’s 10th problem!} \)

One can compute \( A_n(n) \)-multiplier of size \( O(n) \)
F₀ -hardness of reachability

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

```plaintext
i := 1; x := 1; y := 1; b := 1; c := 1; d := 1
```

...and loop

```plaintext
1: i := 1; x := 1; y := 1; b := 1; c := 1; d := 1
2: loop
3: x := 1; y := 1; c := 1; d := 1
4: loop
5: loop
6: c := i; c' := i
7: loop at most \( i \) times
8: x := i
9: loop
10: b := 1; i := i + 1
11: loop
12: b' := 1; b := 1
13: loop
14: c' := 1; c := 1
15: loop at most \( b \) times
16: x' := 1; x := 1; d := 1
17: i := 1
18: zero? i
19: loop
20: x := i; y := 1
21: halt if \( y = 0 \)
```

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

program without zero-tests:

```plaintext
A_n(n)-multiplier
```

```plaintext
1: x := 1; y := 1
2: loop
3: x := 1; y := 1
4: for i := n down to 1 do
5: x := 1; z := i
6: loop
7: x := i + 1; z := i
8: loop
9: x := n + 1; y := 1
10: halt if \( y = 0 \)
```

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

```plaintext
i := 1; x := 1; y := 1; b := 1; c := 1; d := 1
2: loop
3: x := 1; y := 1; b := 1; c := 1; d := 1
4: loop
5: loop
6: c := i; c' := i
7: loop at most \( i \) times
8: x := i; d := i + 1
9: loop
10: b := 1; b := 1
11: loop
12: b' := 1; b := 1
13: loop
14: loop at most \( b \) times
15: b := 1; b := 1
16: i := 1
17: zero? i
18: loop
19: x := i; y := 1
20: halt if \( y = 0 \)
```

inspected using b, c, d

can halt?
Instrumentation - simulation of zero tests

- \( b = A_n(n) \)
- \( c > 0 \)
- \( d = b \cdot c \)
- \( x = y = 0 \) zero-tested counters

\[
\begin{align*}
1: & \quad i += 1 \quad x += 1 \quad y += 1 \quad b += 1 \quad c += 1 \quad d += 1 \\
2: & \quad \text{loop} \\
3: & \quad x += 1 \quad y += 1 \quad c += 1 \quad d += 1 \\
4: & \quad \text{loop} \\
5: & \quad \text{loop} \\
6: & \quad c -= i \quad c' += i \\
7: & \quad \text{loop at most} \ A_n(n) \text{times} \\
8: & \quad x -= 1 \quad c += 1 \quad x' += i + 1 \\
9: & \quad \text{loop} \\
10: & \quad b -= 1 \quad d += i + 1 \\
11: & \quad \text{loop} \\
12: & \quad \text{loop} \\
13: & \quad c' -= 1 \quad h += 1 \\
14: & \quad \text{loop} \\
15: & \quad c' = c \\
16: & \quad x' -= 1 \quad x += 1 \quad d += 1 \\
17: & \quad i += 1 \\
18: & \quad \text{zero? } i \\
19: & \quad \text{loop} \\
20: & \quad x -= i \quad y -= 1 \\
21: & \quad \text{halt if } y = 0
\end{align*}
\]

Aim:

simulate \( A_n(n)/2 \) zero-tests on \( x, y \)

\[ P \]

\begin{itemize}
\item instrument increments and decrements:
\begin{align*}
\text{command} & \quad & \text{replaced by} \\
x += 1 & \quad & x += 1 \quad c -= 1 \\
x -= 1 & \quad & x -= 1 \quad c += 1
\end{align*}
\end{itemize}

\begin{itemize}
\item replace \textbf{zero? } x \textbf{ by}
\end{itemize}

\begin{itemize}
\item replace \textbf{halt by}
\end{itemize}

halt if \( \ldots, d = 0 \).

halt of \( M \)
- simulation of zero tests

\[ d = b \cdot (c + x + y) \]

\[ \text{const} \]

**ZERO? x:**

1: **loop**

2: \[ y -= 1 \quad x += 1 \quad d -= 1 \]

3: **loop**

4: \[ c -= 1 \quad y += 1 \quad d -= 1 \]

5: **loop**

6: \[ y -= 1 \quad c += 1 \quad d -= 1 \]

7: **loop**

8: \[ x -= 1 \quad y += 1 \quad d -= 1 \]

9: \[ b -= 2 \]

- d decreases by \( 2 \cdot (c + x + y) \)
- d decreases by \(< 2 \cdot (c + x + y)\)
- x = 0 initially and finally, y preserved
- d decreases by \( \leq 2 \cdot (c + x + y) \)
- b decreases by 2
- will surely fail

put x, y on budget c
$F_{\omega}$-hardness of reachability

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

```plaintext
1: i = 1
2: x = 1
3: y = 1
4: b = 1
5: c = 1
6: d = 1
7: loop
8: x = i
9: y = i
10: b = i
11: c = i
12: d = i
13: loop
14: x = x
15: y = y
16: b = b
17: c = c
18: d = d
19: loop
20: x = 1
21: y = 1
22: halt if y = 0
```

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

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

program **without** zero-tests:

```
1: x += 1
2: y += 1
3: loop
4: x += 1
5: y += 1
6: for i := n down to 1 do
7:     x := i
8:     z := i
9:     loop
10:     x := i + 1
11:     z := i
12:     loop
13:     x := n
14:     y := 1
15:     halt if y = 0
```

$A_n(n)$-multiplier

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

---

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

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

program **without** zero-tests:

```
1: i += 1
2: x += 1
3: y += 1
4: b = 1
5: c = 1
6: d = 1
7: loop
8: x += i
9: y += i
10: b = i
11: c = i
12: d = i
13: loop
14: x += i
15: y += i
16: b += i
17: c += i
18: d += i
19: loop
20: x = i
21: y = i
22: halt if y = 0
```
III. $\mathcal{F}_\omega$-hardness

- reduction
- multipliers and simulation of zero-tests
- amplifiers
- open questions
One can compute $A_n(n)$-multiplier of size $O(n)$.
The set computed by a counter program from a set \( I \)

**a set \( I \) of initial valuations**

initial valuation: all counters 0

```
1: x += 1    y += 1
2: loop
3: x += 1    y += 1
4: for i := n down to 1 do
5:   loop
6:     x -= 1    z += 1
7:   loop
8:     x += i + 1  z -= i
9:   loop
10: x -= n + 1  y -= 1
11: halt if y = 0.
```

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

the set of all valuations at successful halt
**F-amplifier**

\[ F : \mathbb{N} \rightarrow \mathbb{N} - \text{ fixed function} \]

For every fixed \( B \):

\[ \text{consider all runs starting in } RATIO(b, c, d, B) \text{ (nondeterminism)} \]

\[ P(b, c, d, b', c', d') \]

\[ \text{halt if } d = 0 \]

\[ RATIO(b', c', d', F(B)) \]

\[ \begin{align*}
RATIO(b, c, d, B) & \{ \\
& \cdot b = B \\
& \cdot c > 0 \\
& \cdot d = b \cdot c \\
& \cdot \text{all other counters } 0
\end{align*} \]
$A_n$-amplifier $\rightarrow A_n(n)$-multiplier

initial valuation: all counters 0

1: \(b \leftarrow n\) \(d \leftarrow n\) \(c \leftarrow 1\)

2: \textbf{loop} \quad \text{$n$-multiplier}

3: \(d \leftarrow n\) \(c \leftarrow 1\)

$A_n(n)$-multiplier

\[
RATIO(b, c, d, n)
\]

1: \(x \leftarrow 1\) \(y \leftarrow 1\)

2: \textbf{loop}

3: \(x \leftarrow 1\) \(y \leftarrow 1\)

4: \textbf{for} \(i \leftarrow n\) \text{down to} \(1\) \text{do}

5: \textbf{loop}

6: \textbf{loop}

7: \(x \leftarrow i + 1\) \(z \leftarrow i\)

8: \textbf{loop}

9: \(x \leftarrow n + 1\) \(y \leftarrow 1\)

10: \text{halt if} \(y = 0\).

$A_n$-amplifier $\rightarrow P(b, c, d, b', c', d')$

$RATIO(b', c', d', A_n(n))$
$A_n$-amplifier

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

- $A_1$-amplifier:

```
1: loop
2:   loop
3:     c := 1  c' += 1  d := 1  d' += 2
4:   loop
5:     c' := 1  c += 1  d := 1  d' += 2
6:     b := 2  b' += 4
7:   loop
8:     c := 1  c' += 1  d := 2  d' += 4
9:     b := 2  b' += 4
```

- amplifier lifting:

$A_k$-amplifier $\rightarrow A_{k+1}$-amplifier

$A_1(n) = 2n$

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

$A_1(n) = 2n$

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

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

$\mathcal{M}$  4-multiplier
$\text{RATIO}(b_1, c_1, d_1, 4)$

$\mathcal{P}$  $A_k$-amplifier
$\text{RATIO}(b_1, c_1, d_1, B)$

$\mathcal{L}$  identity-amplifier
$\text{RATIO}(b_1, c_1, d_1, B)$

$\mathcal{P}$  $A_k$-amplifier
$\text{RATIO}(b_2, c_2, d_2, A_k(B))$

$\text{RATIO}(b_2, c_2, d_2, B)$

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

instrumented using $b, c, d$

$1: \mathcal{M}$
$2: \text{loop}$
$3: \mathcal{P}$
$4: \text{zero? } d_1$
$5: \mathcal{L}$
$6: \text{zero? } d_2$
$7: \mathcal{P}$
$8: \text{zero? } d_1$
III. $F_\omega$-hardness

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

- dimension-parametric complexity: $F_k$-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. $F_\omega$-hardness

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