Lower bounds for reachability in VASS in fixed dimension

Wojciech Czerwiński
Sławomir Lasota
University of Warsaw

Jerome Leroux
University of Bordeaux

Agata Dubiak
Łukasz Orlikowski
University of Warsaw

Ranko Lazic
University of Warwick

Filip Mazowiecki
MPI Saarbruecken

Infinity'20, online, 2020-07-07

Plan

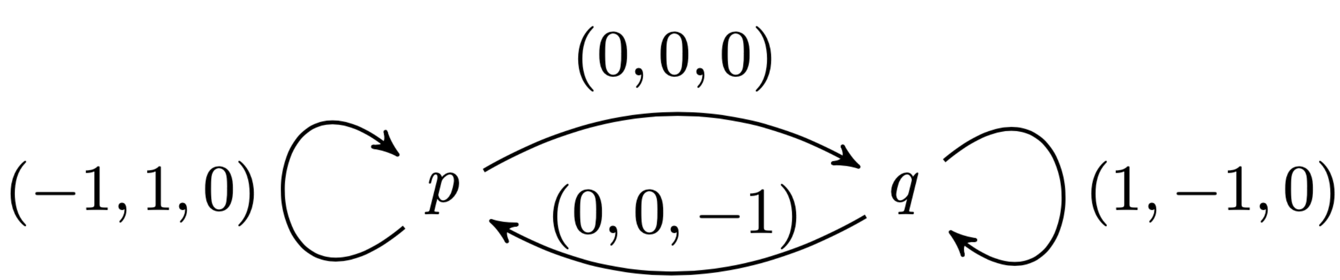
1. Vector addition systems with states (**VASS**) and the reachability problem

2. Lower bounds in **small** fixed dimensions

3. Lower bounds in **large** fixed dimensions

Many faces of vector addition systems with states

- vector addition systems with states **VASS** **[Hopcroft, Pansiot '79]**:

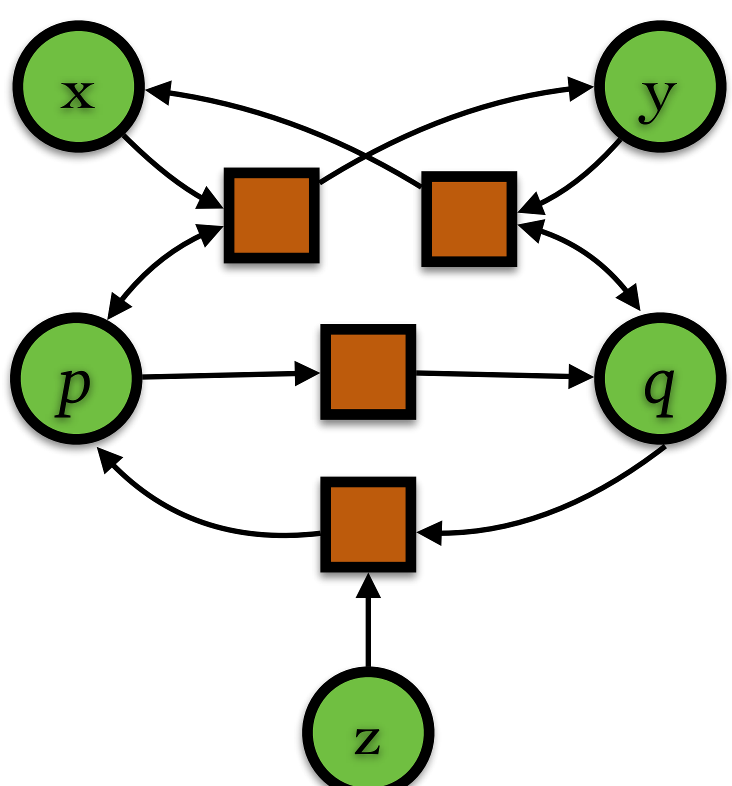


=

- counter programs **without 0-tests**:

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

- Petri nets **[Petri 1962]**:



- VAS **[Karp, Miller '69]**
- multiset rewriting
- …

Counter programs without zero tests

counters are nonnegative integer variables initially all equal zero

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')	

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: $x' = x = y = 0$

finally: $x' = 0 \quad x = 100 \quad y = 200$

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.
```

Minsky machines

the conditional jump of Minsky machines

$$\textbf{if } \mathsf{x} = 0 \textbf{ then goto } L \textbf{ else } \mathsf{x} \mathrel{-}= 1$$

is simulated by counter program **with zero tests**:

```
1: goto 2 or 4
2: zero? x
3: goto L
4: x -= 1
```

Reachability (and coverability)

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.
```

configuration reachability

can it terminate (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.
```

control-location reachability

can it terminate (reach its halt command)?

Complexity of reachability (and coverability)

Time/space needed is at least $F_3(n) = \left.2^{2^{2^{\cdots^{2}}}}\right\}O(n)$

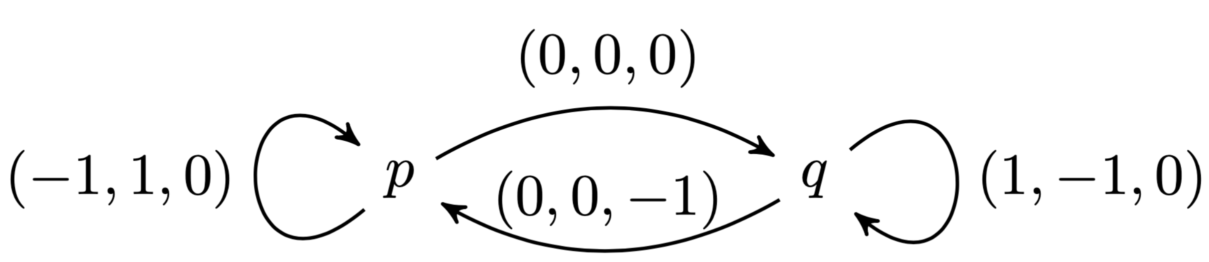
	coverability	reachability
lower bound	EXPSPACE [Lipton '76]	**Tower** [Czerwiński, L., Laz… …oux, Mazowiecki '19] ~~EXPSPACE~~ [Lipt…
upper bound	EXPSPACE [Rackoff '78]	A… …eroux, Schmitz '15, '19] ~~decidable~~ [Sacerdote, Tenney '77] [Mayr '81] [Kosaraju '82] [Lambert '92] [Leroux '09]

see Jerome Leroux's invited talk on Thursday

Time/space needed is at most $F_\omega(n)$

2. Lower bounds for reachability in **small** fixed dimensions

dimension = number of counters:




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

Reachability in dimension 1 and 2

encoding of integers

	unary	binary
dimension 1	NL* [Valiant, Peterson '75]	NP* [Haase, Kreutzer, Ouaknine, Worrell '09]
dimension 2	NL* [Englert, Lazic, Totzke '16]	PSPACE* [Blondin, Finkel, Goeller, Haase, McKenzie '15]

*complete

shortest run has **polynomial** length

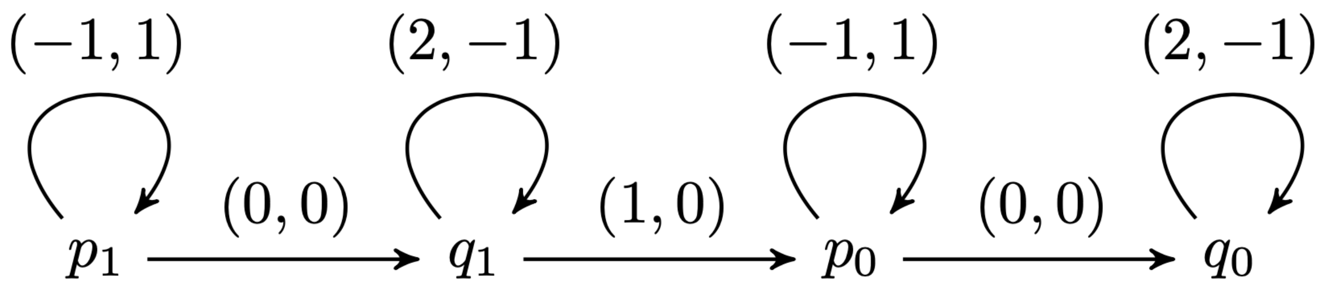
shortest run has **exponential** length

effectively **flattable** in dimension 2

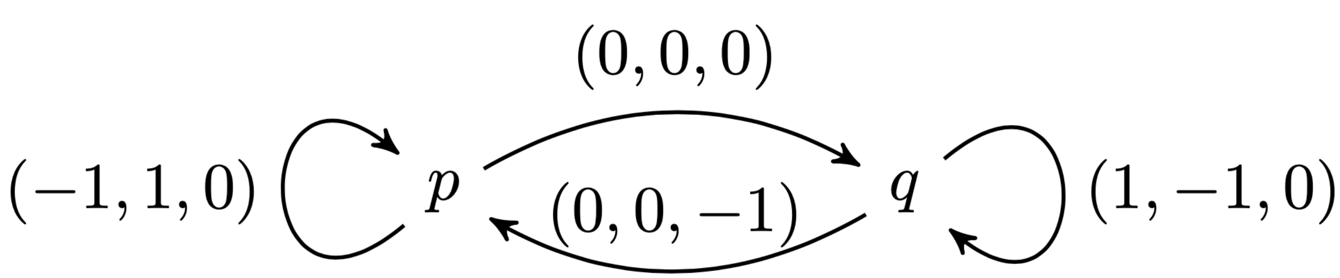
Upper bounds similar to dimension 2 for **every** fixed dimension?

…at least for flat counter programs?

Flat = no nested loops




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




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

Shortest runs in small dimensions

[Czerwiński, L., Lazic, Leroux, Mazowiecki '20]

	unary flat	unary	binary
dim 1	poly*	poly*	exp*
dim 2	poly*	poly*	exp*
dim 3	**exp***	exp*	**?**
dim 4			**2-exp***
dim 5			
…			

*upper bound
*lower bound

Key ingredient: computing **exactly** large numbers

Exponential shortest run in unary flat dim 3

```
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.
```

objective:

halts iff y divisible by L

halts iff x = $(n+1) \cdot y$

iff all multiplicatio

iff all inner loops it

iff $(n+1) \cdot y$ divisi

$$x <= \frac{n+1}{n} \cdot \frac{n}{n-1} \cdot \ldots \cdot \frac{3}{2} \cdot \frac{2}{1} \cdot y$$

```
loop
    x −= 1    z += 1
loop
    x += 5    z −= 4

loop
    x −= 1    z += 1
loop
    ...
    x −= 1    z += 1
loop
    x += 3    z −= 2

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

program size $O(n^2)$, shortest run $2^{O(n^c)}$

Complexity lower bounds for reachability

[Czerwiński, L., Lazic, Leroux, Mazowiecki '19, '20]
[Dubiak '20]

What about coverability?

	unary flat	unary	binary
dim 1	NL*	NL*	NP*
dim 2	NL* NL	NL* NL	PSPACE* PSPACE
dim 3	? NL	NL	PSPACE
dim 4	NL	NL ?	PSPACE
dim 5	NP* NL	NL	PSPACE
…	NP* NL	NL	PSPACE
dim 13	NP* NL	PSPACE* NL	EXPSPACE* PSPACE
dim 14	NP* NL	EXPSPACE* NL	2-EXPSPACE* PSPACE
dim 15	NP* NL	2-EXPSPACE* NL	3-EXPSPACE* PSPACE
…	… NL	… NL	… PSPACE

also for binary flat

*complete
*hard

3. Lower bounds for reachability in **large** fixed dimensions

Parametric lower bound for reachability

[Czerwiński, L., Lazic, Leroux, Mazowiecki '19]
[Czerwiński, L., Orlikowski '??]

	unary flat	unary	binary
dim 13		**PSPACE***	
dim 14		**EXPSPACE***	
dim 15		**2-EXPSPACE***	
...		...	

Minsky machine $\mathcal{M}$ of size n
with counters bounded by

$$\left.2^{2^{2^{\cdots^{n}}}}\right\}h+1$$

$\mathcal{O}$(n+h) ⟶ Counter program of size $\mathcal{O}$(n+h) with $\mathcal{O}$(h) = h+13 counters

$\mathcal{O}$(n+log h), improved reduction ⟶ Counter program of size $\mathcal{O}$(n+log h) with $\mathcal{O}$(log h) counters

Parametric lower bound for reachability

[Czerwiński, L., Lazic, Leroux, Mazowiecki '19]
[Czerwiński, L., Orlikowski '??]

	unary flat	unary	binary
dim 13		**PSPACE***	
dim 14		**EXPSPACE***	
dim 15		**2-EXPSPACE***	
...		...	

Reachability problem for programs of size n with d counters needs space at least

$$2^{2^{2^{\cdots^{n}}}} \Big\} O(d) \qquad d-13$$

improved lower bound:

$$2^{2^{2^{\cdots^{n}}}} \Big\} 2^{O(d)} \qquad 2^{(d-13)/3}$$

Exponential amplifier

```
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
```

```
loop
    b −= 1    b′ += 1
loop
    b′ −= 1    b += 1
    <body>
```

```
x′ += 1
loop
    i −= 1    i′ += 1    x′ += 1
zero? i
loop
    i′ −= 1    i += 1
zero? i′
```

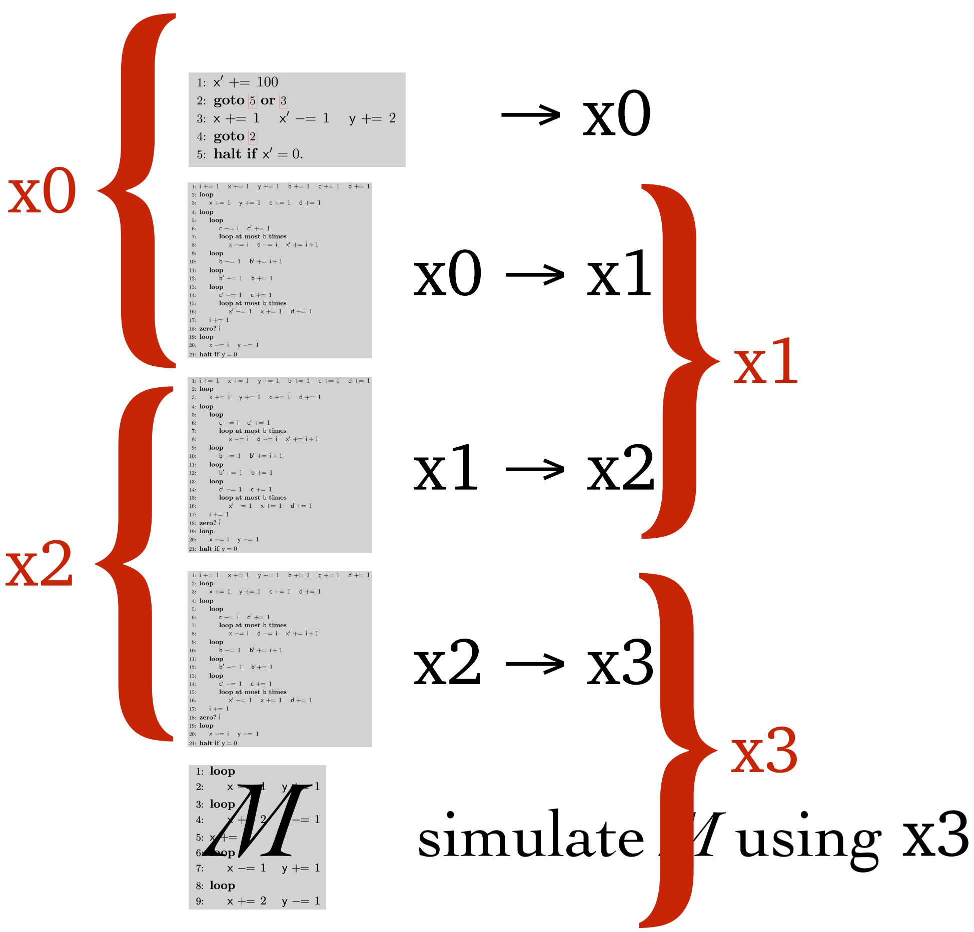
```
loop
    i −= 1    i′ += 1    x −= 1
zero? i
loop
    i′ −= 1    i += 1
zero? i′
```

x → x′: starting with x>0 and x′ = 0, computes x′ exponentially larger than x (if x = 0 at the end)

if so, also all other counters are forcedly 0 at the end

Composing exponential amplifiers

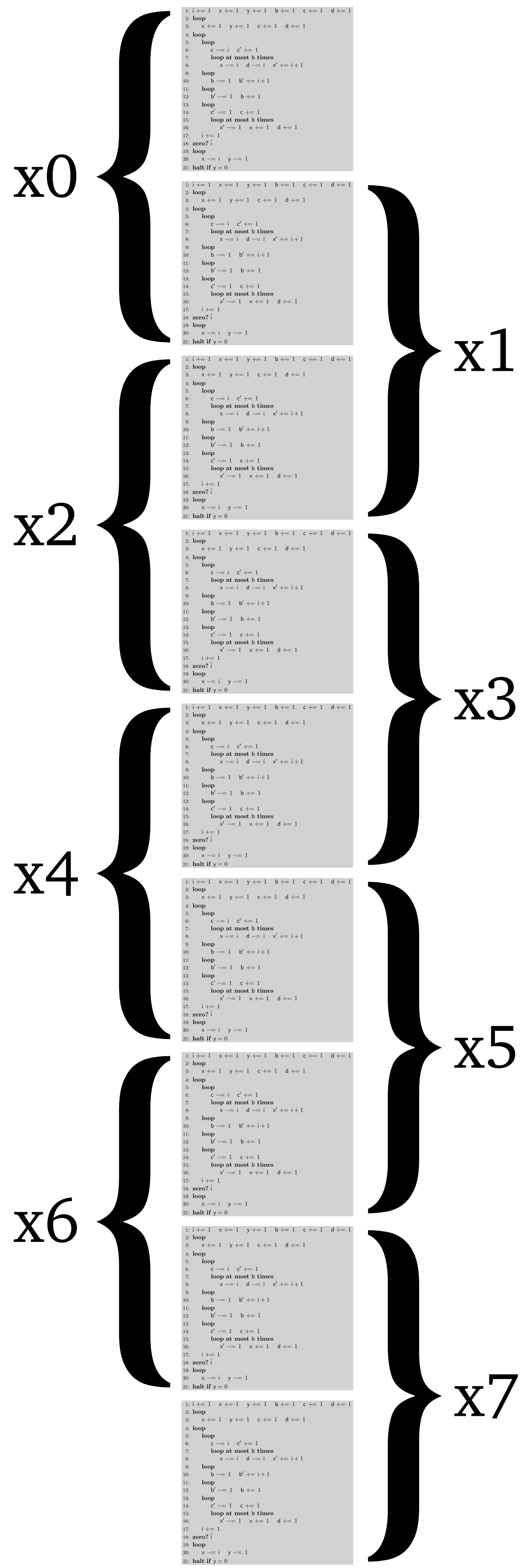
	unary flat	unary	binary
dim 13		PSPACE*	
dim 14		EXPSPACE*	
dim 15		2-EXPSPACE*	
…		…	



halt if x0, x1, x2, x3 = 0

relay-race

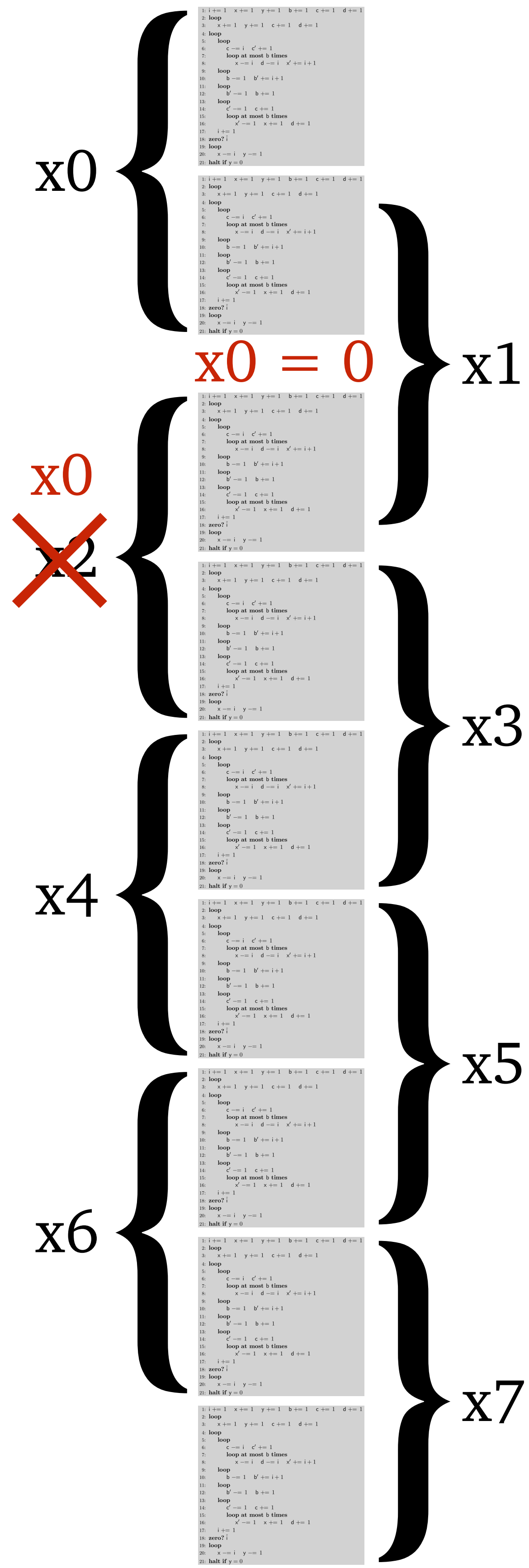
Relay-race



objective: decrease the number of counters to logarithmic

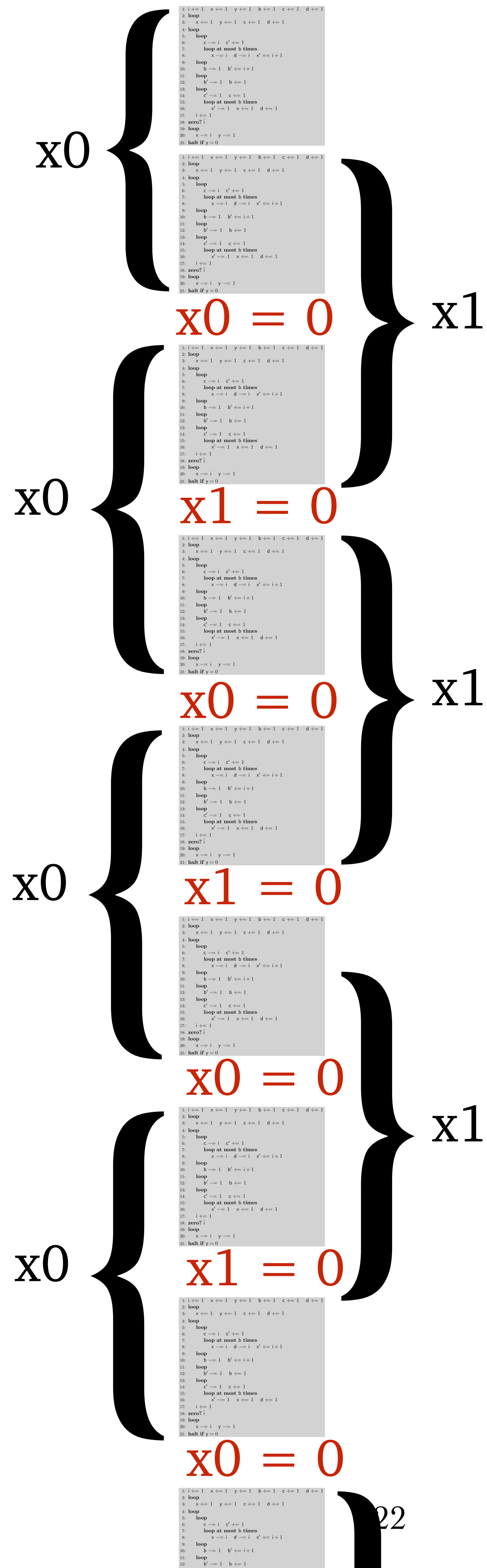
halt if x0, x1, x2, x3, x4, x5, x5, x7 = 0

Counter recycling?

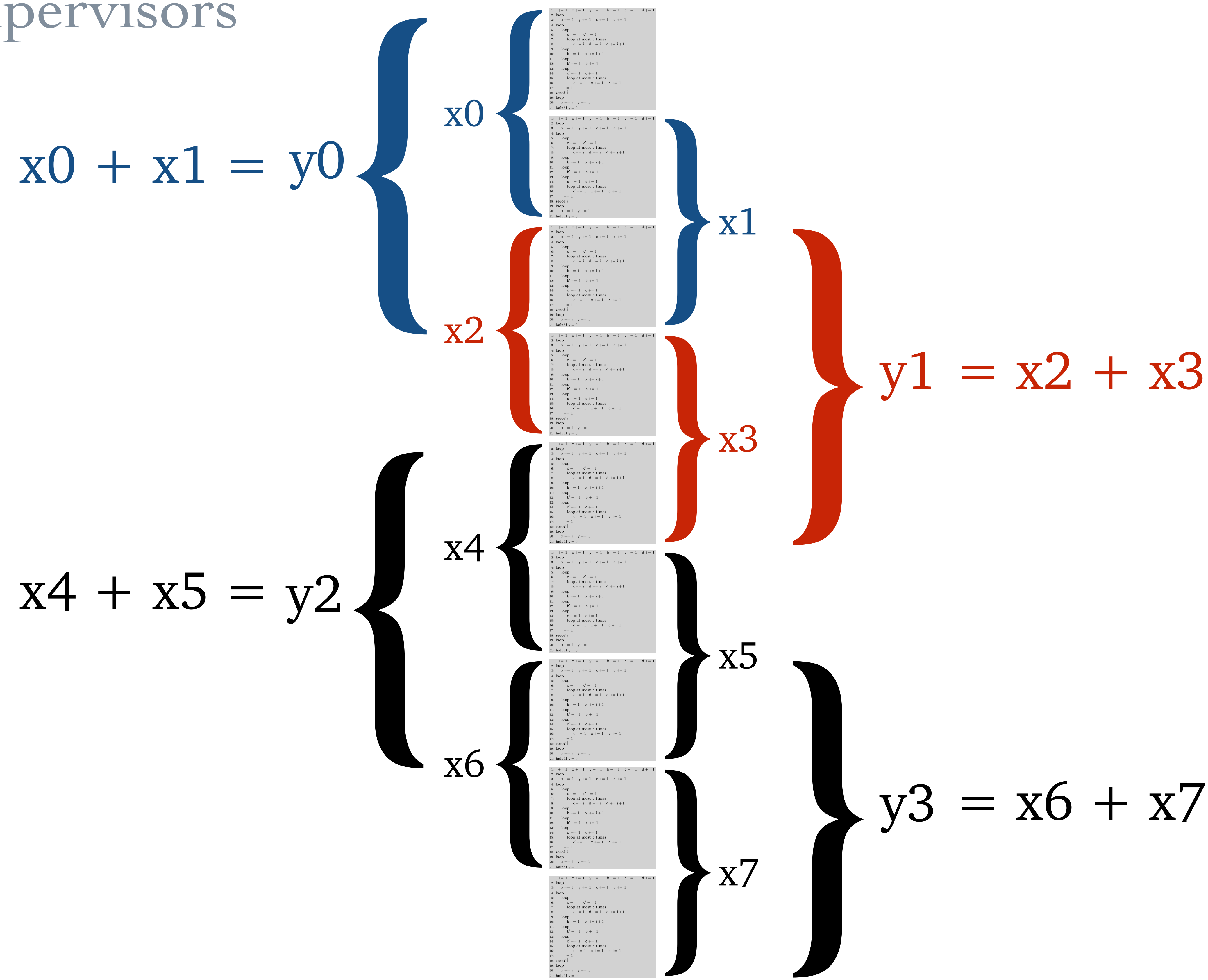


halt if x0, x1, ~~x2~~, x3, x4, x5, x5, x7 = 0

Counter recycling?

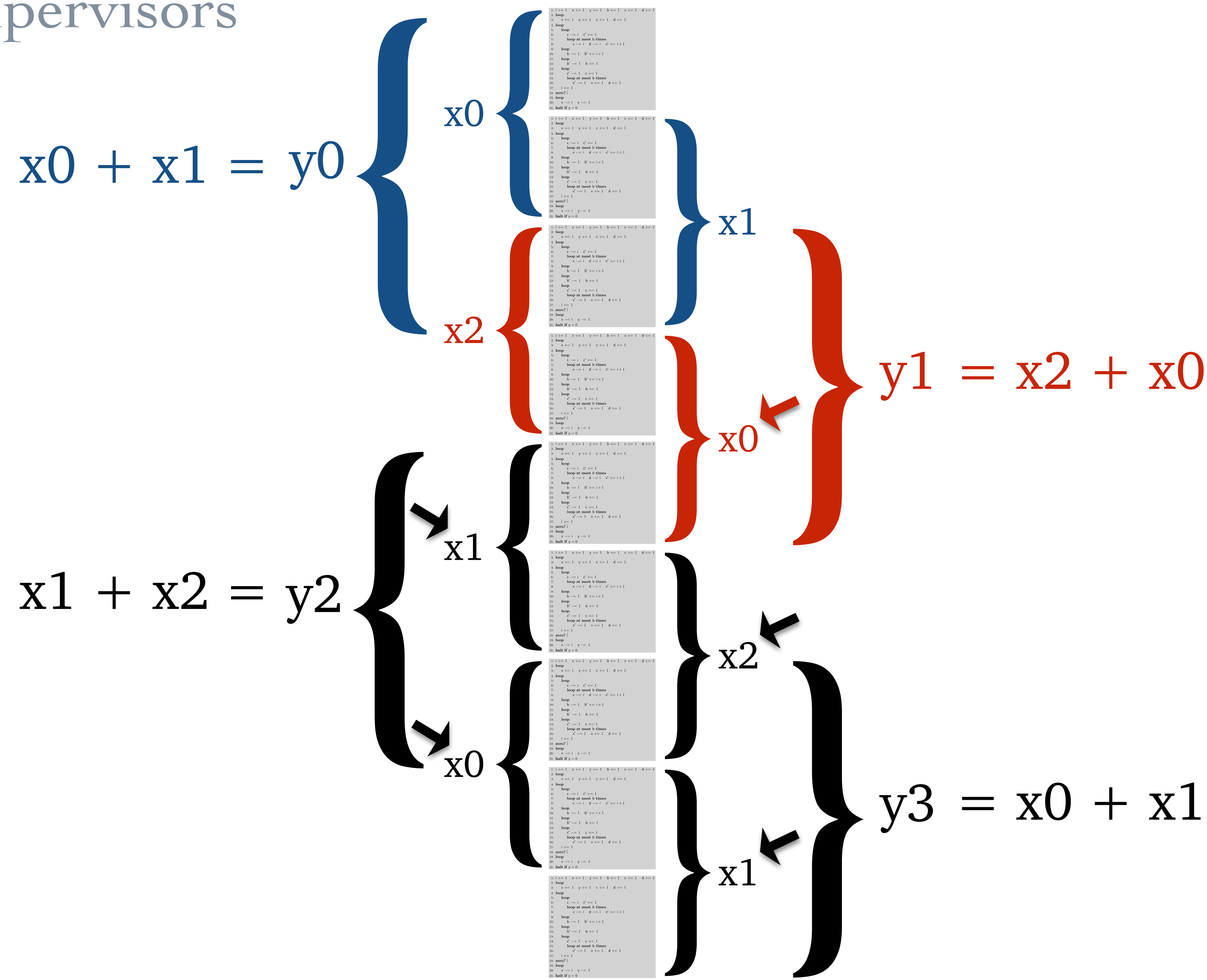


Supervisors



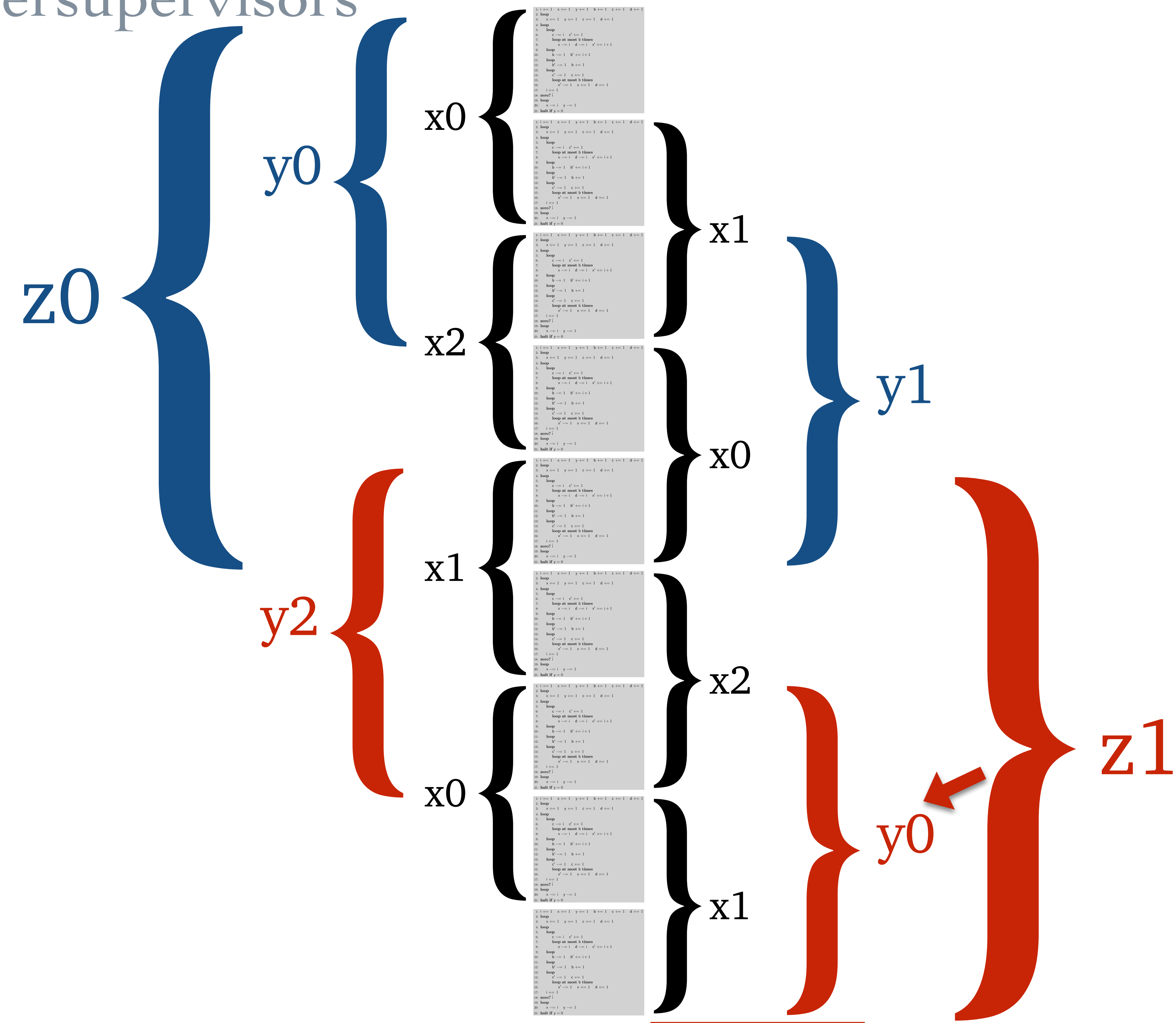
halt if x0, x1, x2, x3, x4, x5, x5, x7,
y0, y1, y2, y3, … = 0

Supervisors



halt if x0, x1, x2, y0, y1, y2, y3, … = 0

Supersupervisors



halt if x0, x1, x2,
y0, y1, y2,
z0, z1, … = 0

and so on…

Summary

	unary flat	unary	binary
dim 1	NL*	NL*	NP*
dim 2	NL*	NL*	PSPACE*
dim 3	?	?	?
dim 4	?	?	?
dim 5	NP*	?	?
…	NP*	?	?
dim 13	NP*	PSPACE*	EXPSPACE*
dim 14	NP*	EXPSPACE*	2-EXPSPACE*
dim 15	NP*	2-EXPSPACE*	3-EXPSPAC
…	…	…	…

*complete
*hard

	unary flat	unary	binary
dim 1	poly*	poly*	exp*
dim 2	poly*	poly*	exp*
dim 3	exp*	exp*	?
dim 4			2-exp*
dim 5			
…			

*upper bound
*lower bound

I recruit for a fully-funded PhD position in the NCN grant

Reachability problem for d-dimensional VASS of size n requires space at least

$$2^{2^{2^{\cdot^{\cdot^{\cdot^{n}}}}}}\Bigg\} 2^{O(d)}$$

thank you!