Reachability in 1–VASS with tests
Vector Addition Systems with States
VASS

[Diagram showing the relationship between uranium waste and electricity production with two arrows indicating produce electricity and recycle uranium.]
Example from *Reachability in Vector Addition Systems is Primitive–Recursive in Fixed Dimension* by Sylvain Schmitz, Highlights 2019
Reachability in VASS

Lower bound: Tower-hard\(^1\)

Upper bound: Ackermann\(^2\)

Reachability with = already in 2–VASS is undecidable (Minsky machine)

1, Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary, 2019

2, Jérôme Leroux and Sylvain Schmitz. Reachability in vector addition systems is primitive-recursive in fixed dimension, 2019
Reachability in VASS

Lower bound: Tower-hard\textsuperscript{1}

Upper bound: Ackermann\textsuperscript{2}

Reachability with = already in 2-VASS is undecidable (Minsky machine)

Input size

\[ 2^{2^{2^{2^{2^{2^{2^{2^{2^{2^{2}}}}}}}}}} \]

A(4, 2) \sim 20,000 decimal digits

Recent results!

1. Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for Petri nets is not elementary. 2019

2. Jérôme Leroux and Sylvain Schmitz. Reachability in vector addition systems is primitive-recursive in fixed dimension. 2019
1-VASS with tests
Path length can be exponential

\[ s \xrightarrow{-2^n} t \xrightarrow{1} s \]
Reachability in 1-VASS with tests

<table>
<thead>
<tr>
<th>Without tests</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NP-complete</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Uses flow = Parikh image of the run
Reachability in 1–VASS with tests

<table>
<thead>
<tr>
<th>Without tests</th>
<th></th>
<th></th>
<th>In 2–VASS without tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-complete</td>
<td></td>
<td></td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
Reachability in 1–VASS with tests

<table>
<thead>
<tr>
<th>Without tests</th>
<th>With = tests</th>
<th>In 2–VASS without tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-complete</td>
<td>NP-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
## Reachability in 1-VASS with tests

<table>
<thead>
<tr>
<th>Without tests</th>
<th>With $=,$ tests</th>
<th>with $\leq &amp; \geq$</th>
<th>In 2-VASS without tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-complete</td>
<td>NP-complete</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
### Reachability in 1–VASS with tests

<table>
<thead>
<tr>
<th>Without tests</th>
<th>With $=$ tests</th>
<th>With $=$ &amp; $\neq$</th>
<th>with $\leq$ &amp; $\geq$</th>
<th>In 2–VASS without tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-complete</td>
<td>NP-complete</td>
<td>Open</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
1-VASS with $= \text{ and } \neq$:

![Diagram](image-url)
1-VASS with $=$ and $\neq$:

Reachability: \text{PSPACE, NP-hard}
1-VASS with $=$ and $\neq$:

My conjecture $=$ NP-complete
1-VASS with $=$ and $\neq$:

My result: NP-membership when all $\neq$ are on the same state
Proving NP–Membership

Valid run: $\rho = (s,0), (u,4), (s,2), (u,6), (s,4), (u,8), (q_1,4), (u,2), (t,0)$
Valid run: $\rho = (s,0), (u,4), (s,2), (u,6), (s,4), (u,8), (q_1,4), (u,2), (t,0)$
u–simple cycles

Valid run: $\rho = (s,0), (u,4), (s,2), (u,6), (s,4), (u,8), (q_1,4), (u,2), (t,0)$

- $weight(c_1) = 2$
- $weight(c_2) = -6$
- $drop(c_1) = 2$
- $drop(c_2) = 6$
u–simple cycles

Valid run: $\rho = (s,0), (u,4), (s,2), (u,6), (s,4), (u,8), (q_1,4), (u,2), (t,0)$
Above the guard
Above the guard
Below the guard
Below the guard but unblocked
Blocked under the guard

Valid run: $\rho = (u,4), (s,2), (u,6), (s,4), (u,8), (q_1,4), (u,2)$

- $c_1 = u, s, u \quad w = 2 \quad \text{drop} = 2$
- $c_2 = u, q_1, u \quad w = -6 \quad \text{drop} = 6$
- $c_3 = u, q_1, q_2, u \quad w = 1 \quad \text{drop} = 11$
Blocked under the guard = very restrictive!

Look at $f := \text{flow of the path}$:
Blocked under the guard = very restrictive!

Look at $f :=$ flow of the path:

- No positive cycle in $f \rightarrow f$ gives a certificate
Blocked under the guard = very restrictive!

Look at $f :=$ flow of the path:

- No positive cycle in $f \rightarrow f$ gives a certificate

- Some positive cycle in $f \rightarrow$
  All positive cycles have the same weight!
  $c_+$ positive u-simple cycle, $d = \text{drop}(c_+) \leq z_i$, $w = \text{weight}(c_+)$

minimal
Blocked under the guard

Only negative cycles in the flow
Compressing the run
Several disequality guards on one state

g_j

g_j - w

C_ = (c_2

C_ C_ C_ C_

c_+ c_+ c_+ c_+

x - w

x - w

z + w

z + w

z

z

C_ C_ C_ C_

c_+ c_+ c_+ c_+

x - w

counter values on u
## Conclusion

<table>
<thead>
<tr>
<th></th>
<th>Without tests</th>
<th>With $=$ tests</th>
<th>With $=$ &amp; $\neq$ on single state</th>
<th>With $=$ &amp; $\neq$</th>
<th>with $\leq$ &amp; $\geq$</th>
<th>In 2-VASS without tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-complete</td>
<td>NP-complete</td>
<td>Open</td>
<td>Open</td>
<td>Open</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
## Conclusion

<table>
<thead>
<tr>
<th>Without tests</th>
<th>With = tests</th>
<th>With = &amp; ≠ on single state</th>
<th>With = &amp; ≠</th>
<th>with ≤ &amp; ≥</th>
<th>In 2-VASS without tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-complete</td>
<td>NP-complete</td>
<td>NP-complete</td>
<td>Open… Not for long?</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>

Not for long?