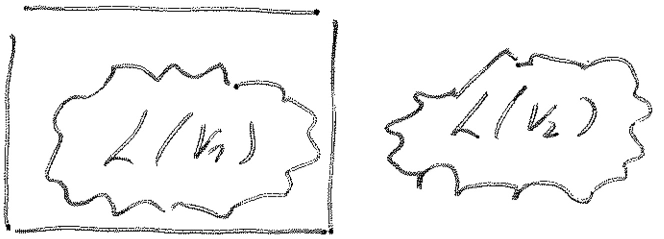
Regular Separability for VASS Reachability Languages

REGSEP:

Given: VASS V_1, V_2 with $L(V_1) \cap L(V_2) = \emptyset$.

Question: $\exists R$ regular:

$$L(V_1) \subseteq R \wedge R \cap L(V_2) = \emptyset.$$



Think: Certificate for disjointness.

Theorem (Keskin & M. '24):

REGSEP is decidable and F_ω-complete.

(has been the big open problem in the area for the past 10 years.)

Example:

(1) 

start and accept with counter value 0.

$\boxed{\text{Counters have to remain non-negative.}}$

$L(V_1) = a^n.b^n$ $\quad$ $L(V_2) = a^n.b^{n+1}$.

Separable? even.even ∪ odd.odd is a separator.

(2) $a^n.b^{>n}$ vs. $a^n.b^{<n}$ not separable.

How to decide that a separator exists?

How to even decide $L(V_1) \cap L(V_2)$

$= L(V_1 \times V_2) = \emptyset$?

<u>VASS - Reachability</u>:

- One of the biggest problems in Track-B, open for 50 years, solved these days.

  $\Rightarrow$ F_ω - complete

  ↳ Upper-bound by Leroux & Schmitz [LICS '19] (following known results).

  ↳ Lower-bound by Leroux [FOCS '21] and Czerwiński & Orlikowski [FOCS '21].

The algorithm for checking regular separability is an <u>enhanced</u> <u>version</u> of the algo for <u>checking</u> <u>reachability</u>.

Discuss the decision procedure for reachability first.

<u>Goal</u>: Check $L(V) = \emptyset$.

<u>Approach</u>:

<u>Abstraction - refinement</u>:

$$L_{\mathbb{Z}}(S_1) \supseteq L_{\mathbb{Z}}(S_2) \supseteq \dots \supseteq L(V)$$

finite sets of VASSes.

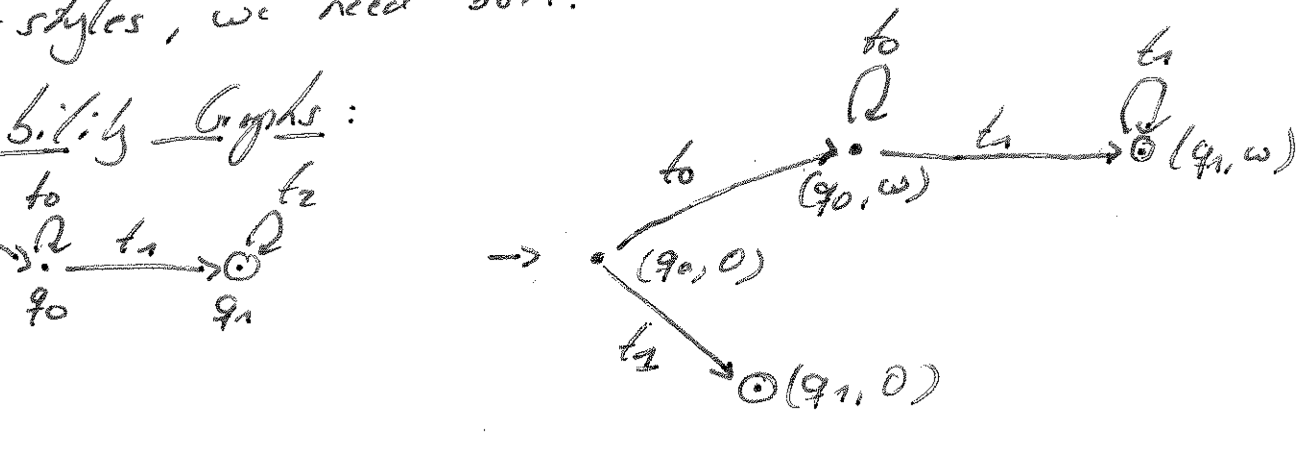
```
non-empty (V) {
    wl := V;        -- finite set of VASSes
                       organized in a work list.
    while wl ≠ ε {
        V' := deq (wl);
        if V' perfect {
            if L_Z (V') ≠ ∅ { return non-empty; }
        } else {
            wl := enq (decompose (V'));
                       finitely many
        }
    }
    return empty;
}
```

for perfect VASS, $\mathbb{Z}$-runs imply $\mathbb{N}$-runs.

Why does this terminate?
Every decomposition makes the VASS more perfect,
reduces a measure in a well-founded partial order.

<u>Approximating VASS</u>:
↳ 2-styles, we need both.

<u>Coverability Graphs</u>:



Good: • Keeps counters non-negative.

Bad: • Needs pumping to translate paths.

• Cannot guarantee exact values.

Marking Equation:

$$\begin{array}{c} c \\ q_0 \\ q_1 \end{array} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Comments: • Order of transitions does not matter. • Solves $L_{\mathbb{Z}}(V) = \emptyset$.

Good: • Can guarantee exact values.

Bad: • Cannot keep counters non-negative.

Solving Reachability:

Idea: • Build coverability graph.

• Determine associated marking equation.

• Solve that

Catch: • Pumping does not respect marking equation.

• It does, when we have perfectness.

1. Petri Nets, VAS, and VASS

Definition:

A Petri net (PN) is a triple (S, T, W)

↳ S = finite set of places.

↳ T = finite set of transitions with $S \cap T = \emptyset$,
↳ $W : (S \times T) \cup (T \times S) \to \mathbb{N}$ weight function.

<u>Marking</u>: $M : S \to \mathbb{N}$, called <u>tokens</u> on places.

<u>Enabledness</u>: Transition <u>t is enabled in M</u>, $M[t\rangle$,
if $M \geq W(-,t)$.

<u>Firing relation</u>: $[\,\rangle \subseteq \mathbb{N}^S \times T \times \mathbb{N}^S$, with
$M_1[t\rangle M_2$, if $M_1[t\rangle$ and
$M_2 = M_1 - W(-,t) + W(t,-)$.

<u>Lemma</u> (<u>Monotonicity of firing</u>):
$M_1[\sigma\rangle M_2$ implies $M_1 + M[\sigma\rangle M_2 + M$.

<u>Lemma</u> (<u>State independence</u>/<u>marking equation</u>):
$M_1[\sigma\rangle M_2$ implies $M_2 = M_1 + \underbrace{\sum_{t \in T} (\overbrace{\Psi(\sigma)}^{: T \to \mathbb{N}})(t) \cdot (-W(-,t) + W(t,-))}$.

Looks like a matrix multiplication.

<u>Forward</u> <u>matrix</u>
$$\mathbb{F} = (W(-,t_1) \ldots W(-,t_n))$$

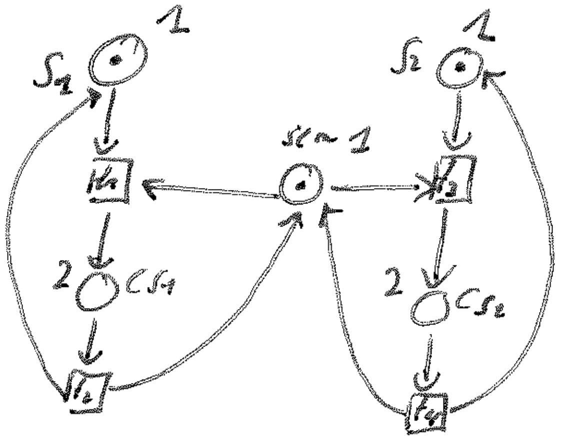
<u>Backward</u> <u>matrix</u>
$$\mathbb{B} = (W(t_1,-) \ldots W(t_n,-))$$

<u>Connectivity</u> <u>matrix</u>
$$\mathbb{C} = \mathbb{B} - \mathbb{F}.$$

Lemma (Marking equation):

$M_1 [\sigma\rangle M_2$ implies $M_2 = M_1 + \mathbb{C} \cdot p(\sigma)$.

Example:



Mutual exclusion holds.

Excursion: Inductive Invariants

Set of markings I

with
- $M_0 \in I$
- $post(I) \subseteq I$
- $I \cap Bad = \emptyset$.

For PN:

Weighted token sum stays unchanged under transitions.

Definition:

An S-invariant $I: S \to \mathbb{Z}$

is a solution to

$$\mathbb{C}^T x = 0.$$

Lemma: $M_0 [\sigma\rangle M$ implies $I^T \cdot M = I^T \cdot M_0$.