

# Weakly Computable Functions

J. Leroux, M. Praveen Ph. Schnoebelen, G. Sutre

LaBRI+CMI+LSV

March 12, 2018

# PN/VASS AS WEAK COMPUTERS

Definition (Weak PN computers, Rabin 1967)

Let  $f : \mathbb{N}^n \rightarrow \mathbb{N}^m$  be a total function.

A **weak PN computer** for  $f$  is a VASS with two designated locations  $q_{\text{init}}$  and  $q_{\text{final}}$ , that satisfies the following two properties.

$$\forall \mathbf{x} : \exists \mathbf{x}', \mathbf{y}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', f(\mathbf{x})), \quad (\text{CO})$$

$$\forall \mathbf{x}, \mathbf{x}', \mathbf{y}', \mathbf{z}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', \mathbf{z}') \text{ implies } \mathbf{z}' \sqsubseteq f(\mathbf{x}). \quad (\text{SA})$$

$f$  is **weakly computable** if there is a weak PN computer for it.

Here is a weak PN computer for halving, i.e., for  $x \mapsto \lfloor \frac{x}{2} \rfloor$ .



**1st Thm.** Not all recursive functions are weakly PN computable. E.g., they must be **monotonic**:  $f(\dots, v+1, \dots) \geq f(\dots, v, \dots)$

# PN/VASS AS WEAK COMPUTERS

Definition (Weak PN computers, Rabin 1967)

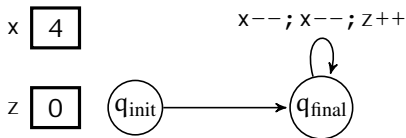
Let  $f : \mathbb{N}^n \rightarrow \mathbb{N}^m$  be a total function.

A **weak PN computer** for  $f$  is a VASS with two designated locations  $q_{\text{init}}$  and  $q_{\text{final}}$ , that satisfies the following two properties.

$$\forall \mathbf{x} : \exists \mathbf{x}', \mathbf{y}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', f(\mathbf{x})), \quad (\text{CO})$$

$$\forall \mathbf{x}, \mathbf{x}', \mathbf{y}', \mathbf{z}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', \mathbf{z}') \text{ implies } \mathbf{z}' \sqsubseteq f(\mathbf{x}). \quad (\text{SA})$$

Here is a weak PN computer for halving, i.e., for  $x \mapsto \lfloor \frac{x}{2} \rfloor$ .



**1st Thm.** Not all recursive functions are weakly PN computable. E.g., they must be **monotonic**:  $f(\dots, v+1, \dots) \geq f(\dots, v, \dots)$

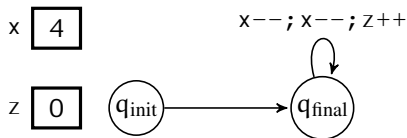
# PN/VASS AS WEAK COMPUTERS

Definition (Weak PN computers, Rabin 1967)

$$\forall x : \exists x', y' : (q_{\text{init}}, x, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, x', y', f(x)), \quad (\text{CO})$$

$$\forall x, x', y', z' : (q_{\text{init}}, x, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, x', y', z') \text{ implies } z' \sqsubseteq f(x). \quad (\text{SA})$$

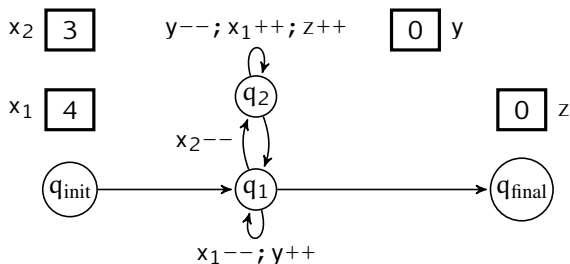
Here is a weak PN computer for halving, i.e., for  $x \mapsto \lfloor \frac{x}{2} \rfloor$ .



**1st Thm.** Not all recursive functions are weakly PN computable. E.g., they must be **monotonic**:  $f(\dots, x+1, \dots) \geq f(\dots, x, \dots)$

# MORE WEAK PN/VASS COMPUTERS

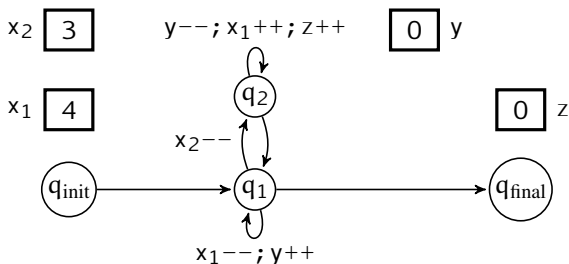
Here is a weak PN Computer for multiplication:  $x_1, x_2 \mapsto x_1 \times x_2$



Proves that reachability sets in PN/VASS are not always semilinear

# MORE WEAK PN/VASS COMPUTERS

Here is a weak PN Computer for multiplication:  $x_1, x_2 \mapsto x_1 \times x_2$



Proves that reachability sets in PN/VASS are not always semilinear

# COMPOSING WEAKLY COMPUTABLE FUNCTIONS?

$$\forall \mathbf{x} : \exists \mathbf{x}', \mathbf{y}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', f(\mathbf{x})), \quad (\text{CO})$$

$$\forall \mathbf{x}, \mathbf{x}', \mathbf{y}', \mathbf{z}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', \mathbf{z}') \text{ implies } \mathbf{z}' \sqsubseteq f(\mathbf{x}). \quad (\text{SA})$$

**Prop.** Weakly PN computable functions are closed under composition.

Hence all multivariate polynomials with positive coefficients are weakly PN computable.

**Example.**  $x_1, x_2, x_3 \mapsto 8x_1^3x_3 + 2x_2x_3^2$ .

Used by Mayr & Rabin to reduce Hilbert's 10th problem to [Equality of VASS reachability sets](#):  $\Rightarrow$  undecidability of VASS equivalence

# COMPOSING WEAKLY COMPUTABLE FUNCTIONS?

$$\forall \mathbf{x} : \exists \mathbf{x}', \mathbf{y}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', f(\mathbf{x})), \quad (\text{CO})$$

$$\forall \mathbf{x}, \mathbf{x}', \mathbf{y}', \mathbf{z}' : (q_{\text{init}}, \mathbf{x}, \mathbf{0}, \mathbf{0}) \xrightarrow{*} (q_{\text{final}}, \mathbf{x}', \mathbf{y}', \mathbf{z}') \text{ implies } \mathbf{z}' \sqsubseteq f(\mathbf{x}). \quad (\text{SA})$$

**Prop.** Weakly PN computable functions are closed under composition.

Hence all multivariate polynomials with positive coefficients are weakly PN computable.

**Example.**  $x_1, x_2, x_3 \mapsto 8x_1^3x_3 + 2x_2x_3^2$ .

Used by Mayr & Rabin to reduce Hilbert's 10th problem to [Equality of VASS reachability sets](#):  $\Rightarrow$  undecidability of VASS equivalence



# ITERABLE WEAKLY COMPUTABLE FUNCTIONS

Definition (Iterable Weak PN Computers, Mayr 1977)

A weak PN computer for  $f : \mathbb{N} \rightarrow \mathbb{N}$  is **iterable** if also

$$\forall \mathbf{w}, \mathbf{w}' : (q_{\text{init}}, \mathbf{w}) \xrightarrow{*} (q_{\text{final}}, \mathbf{w}') \text{ implies } \|\mathbf{w}'\| \leq f(\|\mathbf{w}\|). \quad (\text{IT})$$

**Prop.** If  $f$  is iterable (..) then  $x, y \mapsto f^x(y)$  is weakly PN computable

**Ex.** Doubling  $x \mapsto 2x$  is **iterable weakly PN computable**



**Prop.** The Fast Growing functions  $(F_i)_{i \in \mathbb{N}}$  are weakly computable

**NB:**  $F_2(x) \approx 2^x$   $F_3(x) \approx 2^{2^{\dots^2}}$  }  $x$  times  $F_x(x) \approx \text{Ackermann}(x)$

Used by Mayr & Jančar to show that **Equivalence of finitary VASS** is

# ITERABLE WEAKLY COMPUTABLE FUNCTIONS

Definition (Iterable Weak PN Computers, Mayr 1977)

A weak PN computer for  $f : \mathbb{N} \rightarrow \mathbb{N}$  is **iterable** if also

$$\forall \mathbf{w}, \mathbf{w}' : (q_{\text{init}}, \mathbf{w}) \xrightarrow{*} (q_{\text{final}}, \mathbf{w}') \text{ implies } \|\mathbf{w}'\| \leq f(\|\mathbf{w}\|). \quad (\text{IT})$$

**Prop.** If  $f$  is iterable (..) then  $x, y \mapsto f^x(y)$  is weakly PN computable

**Ex.** Doubling  $x \mapsto 2x$  is **iterable weakly PN computable**



**Prop.** The Fast Growing functions  $(F_i)_{i \in \mathbb{N}}$  are weakly computable

**NB:**  $F_2(x) \approx 2^x$   $F_3(x) \approx 2^{2^{\dots^2}}$  }  $x$  times  $F_x(x) \approx \text{Ackermann}(x)$

Used by Mayr & Jančar to show that **Equivalence of finitary VASS** is

# ITERABLE WEAKLY COMPUTABLE FUNCTIONS

Definition (Iterable Weak PN Computers, Mayr 1977)

A weak PN computer for  $f : \mathbb{N} \rightarrow \mathbb{N}$  is **iterable** if also

$$\forall \mathbf{w}, \mathbf{w}' : (q_{\text{init}}, \mathbf{w}) \xrightarrow{*} (q_{\text{final}}, \mathbf{w}') \text{ implies } \|\mathbf{w}'\| \leq f(\|\mathbf{w}\|). \quad (\text{IT})$$

**Prop.** If  $f$  is iterable (..) then  $x, y \mapsto f^x(y)$  is weakly PN computable

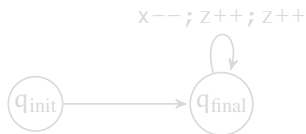
**Ex.** Our weak PN computer for halving  $x \mapsto \lfloor \frac{x}{2} \rfloor$  was **not iterable**

In fact iterable functions are strongly monotonic:  $f(x+1) \geq f(x) + 1$

**Ex.** Doubling  $x \mapsto 2x$  is **iterable weakly PN computable**

x 4

z 0



**Prop.** The Fast Growing functions  $(F_i)_{i \in \mathbb{N}}$  are weakly computable

# ITERABLE WEAKLY COMPUTABLE FUNCTIONS

Definition (Iterable Weak PN Computers, Mayr 1977)

A weak PN computer for  $f : \mathbb{N} \rightarrow \mathbb{N}$  is **iterable** if also

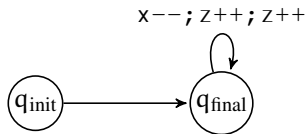
$$\forall \mathbf{w}, \mathbf{w}' : (q_{\text{init}}, \mathbf{w}) \xrightarrow{*} (q_{\text{final}}, \mathbf{w}') \text{ implies } \|\mathbf{w}'\| \leq f(\|\mathbf{w}\|). \quad (\text{IT})$$

**Prop.** If  $f$  is iterable (..) then  $x, y \mapsto f^x(y)$  is weakly PN computable

**Ex.** Doubling  $x \mapsto 2x$  is **iterable weakly PN computable**

x 4

z 0



**Prop.** The Fast Growing functions  $(F_i)_{i \in \mathbb{N}}$  are weakly computable

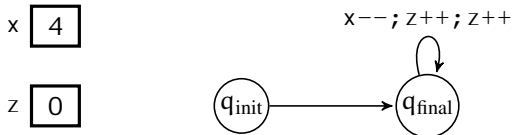
**NB:**  $F_2(x) \approx 2^x$   $F_3(x) \approx 2^{2^{\dots^2}}$  }  $x$  times  $F_x(x) \approx \text{Ackermann}(x)$

Used by Mayr & Jančar to show that **Equivalence of finitary VASS** is

# ITERABLE WEAKLY COMPUTABLE FUNCTIONS

**Prop.** If  $f$  is iterable (..) then  $x, y \mapsto f^x(y)$  is weakly PN computable

**Ex.** Doubling  $x \mapsto 2x$  is **iterable weakly PN computable**



**Prop.** The Fast Growing functions  $(F_i)_{i \in \mathbb{N}}$  are weakly computable

**NB:**  $F_2(x) \approx 2^x$   $F_3(x) \approx 2^{2^{\dots^2}}$  }  $x$  times  $F_x(x) \approx \text{Ackermann}(x)$

Used by Mayr & Jančar to show that **Equivalence of finitary VASS** is Ackermann-hard

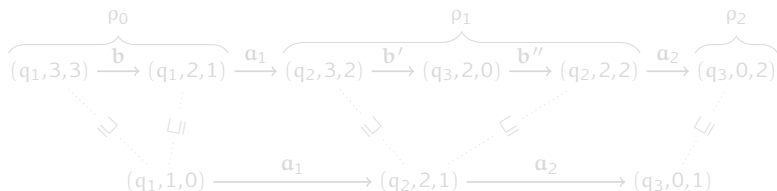
# ORDERING RUNS OF A PN/VASS

For steps:

$$(q_1, \mathbf{x}_1) \sqsubseteq (q_2, \mathbf{x}_2) \stackrel{\text{def}}{\Leftrightarrow} q_1 = q_2 \wedge \mathbf{x}_1 \sqsubseteq \mathbf{x}_2,$$

$$(c_1 \xrightarrow{\mathbf{a}_1} c'_1) \preceq (c_2 \xrightarrow{\mathbf{a}_2} c'_2) \stackrel{\text{def}}{\Leftrightarrow} c_1 \sqsubseteq c_2 \wedge \mathbf{a}_1 = \mathbf{a}_2 \wedge c'_1 \sqsubseteq c'_2.$$

For runs [Jančar 1990]:  $\rho \preceq \rho'$  if  $\rho = c_0 \mathbf{a}_1 c_1 \dots \mathbf{a}_k c_k$  and  $\rho'$  is some  $\rho' = \rho_0 \mathbf{a}_1 \rho_1 \dots \mathbf{a}_k \rho_k$  with  $c_j \sqsubseteq \text{src}(\rho_j)$  and  $c_j \sqsubseteq \text{tgt}(\rho_j)$  for all  $j = 0, \dots, k$ .



**Lem.** (WQO) any sequence  $\rho_0, \rho_1, \dots$  contains an increasing  $\rho_i \preceq \rho_j$

**Pumping Lemma.** Let  $\rho \preceq \rho'$  and let  $\mathbf{u}, \mathbf{v} \in \mathbb{N}^d$  such that  $\text{src}(\rho') = \text{src}(\rho) + \mathbf{u}$  and  $\text{tgt}(\rho') = \text{tgt}(\rho) + \mathbf{v}$ . Then

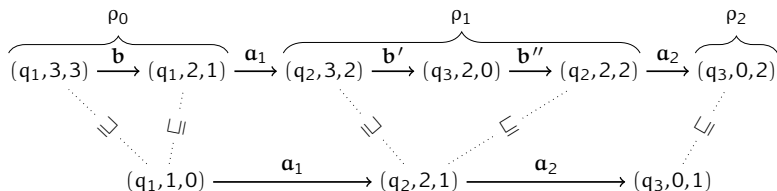
# ORDERING RUNS OF A PN/VASS

For steps:

$$(q_1, \mathbf{x}_1) \sqsubseteq (q_2, \mathbf{x}_2) \stackrel{\text{def}}{\Leftrightarrow} q_1 = q_2 \wedge \mathbf{x}_1 \sqsubseteq \mathbf{x}_2,$$

$$(c_1 \xrightarrow{\mathbf{a}_1} c'_1) \preceq (c_2 \xrightarrow{\mathbf{a}_2} c'_2) \stackrel{\text{def}}{\Leftrightarrow} c_1 \sqsubseteq c_2 \wedge \mathbf{a}_1 = \mathbf{a}_2 \wedge c'_1 \sqsubseteq c'_2.$$

For runs [Jančar 1990]:  $\rho \preceq \rho'$  if  $\rho = c_0 \mathbf{a}_1 c_1 \dots \mathbf{a}_k c_k$  and  $\rho'$  is some  $\rho' = \rho_0 \mathbf{a}_1 \rho_1 \dots \mathbf{a}_k \rho_k$  with  $c_j \sqsubseteq \text{src}(\rho_j)$  and  $c_j \sqsubseteq \text{tgt}(\rho_j)$  for all  $j = 0, \dots, k$ .



**Lem.** (WQO) any sequence  $\rho_0, \rho_1, \dots$  contains an increasing  $\rho_i \preceq \rho_j$

**Pumping Lemma.** Let  $\rho \preceq \rho'$  and let  $\mathbf{u}, \mathbf{v} \in \mathbb{N}^d$  such that  $\text{src}(\rho') = \text{src}(\rho) + \mathbf{u}$  and  $\text{tgt}(\rho') = \text{tgt}(\rho) + \mathbf{v}$ . Then

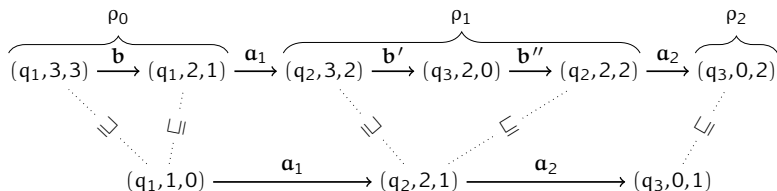
# ORDERING RUNS OF A PN/VASS

For steps:

$$(q_1, \mathbf{x}_1) \sqsubseteq (q_2, \mathbf{x}_2) \stackrel{\text{def}}{\Leftrightarrow} q_1 = q_2 \wedge \mathbf{x}_1 \sqsubseteq \mathbf{x}_2,$$

$$(c_1 \xrightarrow{\mathbf{a}_1} c'_1) \preceq (c_2 \xrightarrow{\mathbf{a}_2} c'_2) \stackrel{\text{def}}{\Leftrightarrow} c_1 \sqsubseteq c_2 \wedge \mathbf{a}_1 = \mathbf{a}_2 \wedge c'_1 \sqsubseteq c'_2.$$

For runs [Jančar 1990]:  $\rho \preceq \rho'$  if  $\rho = c_0 \mathbf{a}_1 c_1 \dots \mathbf{a}_k c_k$  and  $\rho'$  is some  $\rho' = \rho_0 \mathbf{a}_1 \rho_1 \dots \mathbf{a}_k \rho_k$  with  $c_j \sqsubseteq \text{src}(\rho_j)$  and  $c_j \sqsubseteq \text{tgt}(\rho_j)$  for all  $j = 0, \dots, k$ .

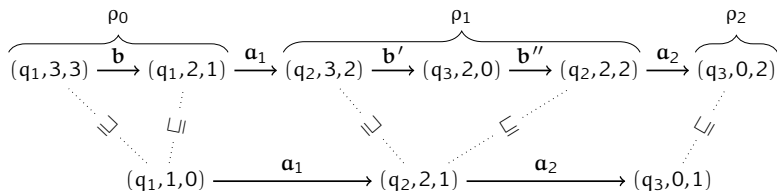


**Lem. (WQO)** any sequence  $\rho_0, \rho_1, \dots$  contains an increasing  $\rho_i \preceq \rho_j$

**Pumping Lemma.** Let  $\rho \preceq \rho'$  and let  $\mathbf{u}, \mathbf{v} \in \mathbb{N}^d$  such that  $\text{src}(\rho') = \text{src}(\rho) + \mathbf{u}$  and  $\text{tgt}(\rho') = \text{tgt}(\rho) + \mathbf{v}$ . Then



# ORDERING RUNS OF A PN/VASS



**Lem. (WQO)** any sequence  $\rho_0, \rho_1, \dots$  contains an increasing  $\rho_i \preceq \rho_j$

**Pumping Lemma.** Let  $\rho \preceq \rho'$  and let  $\mathbf{u}, \mathbf{v} \in \mathbb{N}^d$  such that  $\text{src}(\rho') = \text{src}(\rho) + \mathbf{u}$  and  $\text{tgt}(\rho') = \text{tgt}(\rho) + \mathbf{v}$ . Then

$$\forall n \in \mathbb{N} : (\text{src}(\rho) + n \cdot \mathbf{u}) \xrightarrow{*} (\text{tgt}(\rho) + n \cdot \mathbf{v}).$$

In above example this yields  $(q_1, 2n + 1, 3n) \xrightarrow{*} (q_3, 0, n + 1)$

# APPLICATION OF THE PUMPING LEMMA

**Thm.** If  $f: \mathbb{N} \rightarrow \mathbb{N}$  is weakly PN computable and unbounded then  $f(x)$  is in  $\Omega(x)$

**Example.**  $x \mapsto \lfloor \log x \rfloor$  or  $x \mapsto \lfloor \sqrt{x} \rfloor$  are not weakly PN computable

**Proof.** Take all runs

$$\rho_x : (q_{\text{init}}, x, \mathbf{0}, 0) \xrightarrow{*} (q_{\text{final}}, x', \mathbf{y}'_x, f(x))$$

1. Find  $x_1 < x_2$  with  $\rho_{x_1} \preceq \rho_{x_2}$  and  $f(x_1) < f(x_2)$  (since  $\preceq$  is wqo)
2. Pumping lemma gives

$$\forall n \in \mathbb{N}: \quad (q_{\text{init}}, x_1 + n\Delta_x, \mathbf{0}, 0) \xrightarrow{*} (q_{\text{final}}, \dots, \dots, f(x_1) + n\Delta_f)$$

3. Hence  $f(x_1 + n\Delta_x) \geq f(x_1) + n\Delta_f$  by (SA)

QED

# APPLICATION OF THE PUMPING LEMMA

**Thm.** If  $f: \mathbb{N} \rightarrow \mathbb{N}$  is weakly PN computable and unbounded then  $f(x)$  is in  $\Omega(x)$

**Example.**  $x \mapsto \lfloor \log x \rfloor$  or  $x \mapsto \lfloor \sqrt{x} \rfloor$  are not weakly PN computable

**Proof.** Take all runs

$$\rho_x : (q_{\text{init}}, x, \mathbf{0}, 0) \xrightarrow{*} (q_{\text{final}}, x', \mathbf{y}'_x, f(x))$$

1. Find  $x_1 < x_2$  with  $\rho_{x_1} \preceq \rho_{x_2}$  and  $f(x_1) < f(x_2)$  (since  $\preceq$  is wqo)
2. Pumping lemma gives

$$\forall n \in \mathbb{N}: \quad (q_{\text{init}}, x_1 + n\Delta_x, \mathbf{0}, 0) \xrightarrow{*} (q_{\text{final}}, \dots, f(x_1) + n\Delta_f)$$

3. Hence  $f(x_1 + n\Delta_x) \geq f(x_1) + n\Delta_f$  by (SA)

QED

# WHAT ABOUT GVAS?

GVAS = Grammar-controlled VAS

$G = (V, \mathbf{A}, R, S)$  is a CFG over finite “alphabet”  $\mathbf{A} \subseteq \mathbb{Z}^d$  for some dimension  $d$

Example:  $S ::= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \mid \begin{pmatrix} -1 \\ 2 \end{pmatrix} S \begin{pmatrix} 2 \\ -1 \end{pmatrix}$

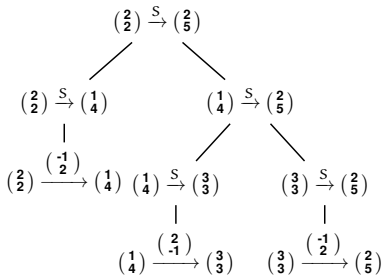
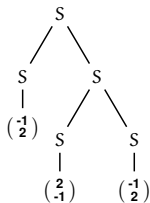
**Def.**  $w \xRightarrow{*}_G w'$ , where  $w, w' \in (V \cup \mathbf{A})^*$ , if ...

**Def.**  $x \xrightarrow{u} y \stackrel{\text{def}}{\iff} \exists w \in \mathbf{A}^* : u \xRightarrow{*} w \wedge x \xrightarrow{w} y$

Similar to Pushdown-VASS with some subtle differences

# DERIVATION TREES AND FLOW TREES

Take  $S ::= SS \mid \begin{pmatrix} -1 \\ 2 \end{pmatrix} \mid \begin{pmatrix} 2 \\ -1 \end{pmatrix}$ .



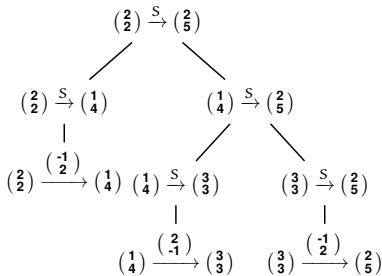
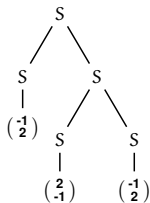
$s = \sigma[s_1, \dots, s_k] \leq_G t = \theta[t_1, \dots, t_\ell]$  iff

- $\sigma \leq \theta$  and
- $\exists$  subtree  $t' = \theta'[t'_1, \dots, t'_{\ell'}]$  of  $t$  st.  $\sigma \leq \theta'$ ,  $\ell' = k$  and  $s_j \leq_G t'_j$  for every  $1 \leq j \leq k$ .

**Fact.**  $\leq_G$  is a wqo

# DERIVATION TREES AND FLOW TREES

Take  $S ::= SS \mid \begin{pmatrix} -1 \\ 2 \end{pmatrix} \mid \begin{pmatrix} 2 \\ -1 \end{pmatrix}$ .



$s = \sigma[s_1, \dots, s_k] \leq_G t = \theta[t_1, \dots, t_\ell]$  iff

- $\sigma \leq \theta$  and
- $\exists$  subtree  $t' = \theta'[t'_1, \dots, t'_{\ell'}]$  of  $t$  st.  $\sigma \leq \theta'$ ,  $\ell' = k$  and  $s_j \leq_G t'_j$  for every  $1 \leq j \leq k$ .

**Fact.**  $\leq_G$  is a wqo

# AMALGAMATION OF FLOW TREES

**Thm.** If  $s \leq_G t_1, t_2$  then exist  $t_1, t_2 \leq_G s'$  s.t.

$$\text{dir}(s') - \text{dir}(s) = \text{dir}(t_1) - \text{dir}(s) + \text{dir}(t_2) - \text{dir}(s).$$

NB: entails pumping as special case

**Coro 1.** Reachable sets can be written  $\sum_i \mathbf{b}_i + \mathbf{P}_i$  with periodic  $\mathbf{P}_i$ ,  
i.e.,  $\mathbf{x}, \mathbf{y} \in \mathbf{P} \implies \mathbf{x} + \mathbf{y} \in \mathbf{P}$ .

**Coro 2.** Weakly computable functions are in  $\Omega(\chi)$ .