Demystifying Reachability in Vector Addition Systems

J. Leroux, S. Schmitz

LaBRI, CNRS, LSV, ENS Cachan & INRIA

LICS 2015 in Kyoto, July 6th, 2015
Outline

vector addition systems (VAS) and their reachability problem

... solved by the KLMST algorithm of Sacerdote and Tenney (1977), Mayr (1981), Kosaraju (1982), and Lambert (1992)

decomposition theorem
the KLMST algorithm constructs an ideal decomposition of the set of runs

upper bound theorem
VAS reachability is in cubic Ackermann
Outline

vector addition systems (VAS) and their reachability problem

... solved by the KLMST algorithm of Sacerdote and Tenney (1977), Mayr (1981), Kosaraju (1982), and Lambert (1992)

decomposition theorem

the KLMST algorithm constructs an ideal decomposition of the set of runs

upper bound theorem

VAS reachability is in cubic Ackermann
vector addition systems (VAS) and their reachability problem

... solved by the KLMST algorithm of Sacerdote and Tenney (1977), Mayr (1981), Kosaraju (1982), and Lambert (1992)

decomposition theorem
the KLMST algorithm constructs an ideal decomposition of the set of runs

upper bound theorem
VAS reachability is in cubic Ackermann
Vector Addition Systems (VAS)

(Karp and Miller, 1969)

**Syntax**

- **dimension** \( d \in \mathbb{N} \)
- finite set \( A \subseteq \text{fin} \mathbb{Z}^d \) of actions \( a \in A \)

**Semantics**

- configurations \( u, v, \ldots \in \mathbb{N}^d \)
- transitions \( u \xrightarrow{a} v \in \mathbb{N}^d \times A \times \mathbb{N}^d \) with \( v = u + a \)
VECTOR ADDITION SYSTEMS (VAS)

(KARP AND MILLER, 1969)

**SYNTAX**

- dimension $d \in \mathbb{N}$
- finite set $A \subseteq \text{fin} \mathbb{Z}^d$ of actions $a \in A$

**SEMANTICS**

- configurations $u, v, ... \in \mathbb{N}^d$
- transitions $u^a \rightarrow v \in \mathbb{N}^d \times A \times \mathbb{N}^d$ with $v = u + a$
Example VAS

d = 2

A = \{ (0,2) \rightarrow (2,4) \rightarrow (3,5) \rightarrow (4,6) \rightarrow (3,4) \rightarrow (2,2) \rightarrow (0,1) = y \}
**Example VAS**

\[ d = 2 \]

\[ A = \{ \text{, } \} \]

\[ x = (0,2) \]
**Example VAS**

\[ d = 2 \]

\[ A = \{ \vec{x}, \vec{y} \} \]

**Example**

\[ x = (0,2) \rightarrow (-1,0) \in \mathbb{N}^2 \]
Example VAS

\[ d = 2 \]

\[ A = \{ , \} \]

\[ x = (0, 2) \rightarrow (1, 3) \]
Example VAS

Example

\[ d = 2 \]

\[ A = \left\{ \begin{array}{c}
\begin{array}{c}
\text{from} \\
\text{to}
\end{array}
\end{array} \right\} \]

\[ x = (0,2) \rightarrow (1,3) \rightarrow (2,4) \]
Example VAS

\[ d = 2 \]

\[ A = \left\{ \begin{array}{c}
\rightarrow \\
\end{array} \right\} \]

\[ x = (0,2) \rightarrow (1,3) \rightarrow (2,4) \rightarrow (3,5) \]
**Example VAS**

\[ d = 2 \]

\[ A = \left\{ \begin{array}{c} \uparrow, \\ \downarrow \end{array} \right\} \]

\[ x = (0,2) \rightarrow (1,3) \rightarrow (2,4) \rightarrow (3,5) \rightarrow (4,6) \]
Example VAS

\[ d = 2 \]

\[ A = \{ \begin{array}{c}
\uparrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\rightarrow \\
\uparrow \\
\end{array} \} \]

\[ x = (0,2) \rightarrow (1,3) \rightarrow (2,4) \rightarrow (3,5) \rightarrow (4,6) \rightarrow (3,4) \]
**Example VAS**

\[ d = 2 \]

\[ \mathbf{A} = \{ \overrightarrow{0,1}, \overrightarrow{2,2} \} \]

\[ \mathbf{x} = (0,2) \rightarrow (1,3) \rightarrow (2,4) \rightarrow (3,5) \rightarrow (4,6) \rightarrow (3,4) \rightarrow (2,2) \]
**Example VAS**

\[ d = 2 \]

\[ A = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle \} \]

\[ x = (0,2) \rightarrow (1,3) \rightarrow (2,4) \rightarrow (3,5) \rightarrow (4,6) \rightarrow (3,4) \rightarrow (2,2) \rightarrow (0,1) = y \]
**Definition (Prerun)**

A prerun is an element

\[(u, (u_1, a_1, v_1) \cdots (u_k, a_k, v_k), v)\]

from \(\text{PreRuns}_A \overset{\text{def}}{=} \mathbb{N}^d \times (\mathbb{N}^d \times A \times \mathbb{N}^d)^* \times \mathbb{N}^d\)

**Definition (Run)**

A prerun is connected (is a run) if

(source) \(u = u_1\)

(transitions) \(\forall 1 \leq j \leq k, u_j + a_j = v_j\)

(contiguity) \(\forall 1 < j \leq k, v_{j-1} = u_j\)

(target) \(v_k = v\)
The Reachability Problem

\[ \text{Runs}_{A}(x, y) \overset{\text{def}}{=} \{ \rho \in \text{PreRuns}_{A} \mid \rho \text{ is a run with source } x \text{ and target } y \} \]

**VAS Reachability**

- **Input**: \( A \subseteq_{\text{fin}} \mathbb{Z}^{d}, x, y \in \mathbb{N}^{d} \)
- **Question**: Is \( y \) reachable from \( x \) in \( A \)?
  - I.e., is \( \text{Runs}_{A}(x, y) \neq \emptyset \)?

**Theorem (Mayr, 1981; Kosaraju, 1982; Lambert, 1992; Leroux, 2011)**

*VAS Reachability is decidable.*

- by the **KLMST algorithm** (Mayr, 1981; Kosaraju, 1982; Lambert, 1992)
- by Presburger invariants (Leroux, 2011)
The Reachability Problem

Runs_A(x, y) \overset{\text{def}}{=} \{ \rho \in \text{PreRuns}_A \mid \rho \text{ is a run with source } x \text{ and target } y \}

**VAS Reachability**

- **input** $A \subseteq \text{fin} \mathbb{Z}^d$, $x, y \in \mathbb{N}^d$
- **question** Is $y$ reachable from $x$ in $A$? I.e., is $\text{Runs}_A(x, y) \neq \emptyset$?

**Theorem (Mayr, 1981; Kosaraju, 1982; Lambert, 1992; Leroux, 2011)**

VAS Reachability is decidable.

- by the KLMST algorithm (Mayr, 1981; Kosaraju, 1982; Lambert, 1992)
- by Presburger invariants (Leroux, 2011)
The Reachability Problem

\[ \text{Runs}_A(x, y) \overset{\text{def}}{=} \{ \rho \in \text{PreRuns}_A \mid \rho \text{ is a run with source } x \text{ and target } y \} \]

**VAS Reachability**

**input** \( A \subseteq \text{fin} \, \mathbb{Z}^d, x, y \in \mathbb{N}^d \)

**question** Is \( y \) reachable from \( x \) in \( A \)?

I.e., is \( \text{Runs}_A(x, y) \neq \emptyset \)?

**Theorem (Mayr, 1981; Kosaraju, 1982; Lambert, 1992; Leroux, 2011)**

VAS Reachability is decidable.

- by the **KLMST algorithm** (Mayr, 1981; Kosaraju, 1982; Lambert, 1992)
- by Presburger invariants (Leroux, 2011)
The Reachability Problem

\[ \text{Runs}_A(x,y) \overset{\text{def}}{=} \{ \rho \in \text{PreRuns}_A \mid \rho \text{ is a run with source } x \text{ and target } y \} \]

**VAS Reachability**

- **input** \( A \subseteq \text{fin } \mathbb{Z}^d, x, y \in \mathbb{N}^d \)
- **question** Is \( y \) reachable from \( x \) in \( A \)?
  - l.e., is \( \text{Runs}_A(x,y) \neq \emptyset \)?

**Theorem (Mayr, 1981; Kosaraju, 1982; Lambert, 1992; Leroux, 2011)**

*VAS Reachability is decidable.*

- by the **KLMST algorithm** (Mayr, 1981; Kosaraju, 1982; Lambert, 1992)
- by Presburger invariants (Leroux, 2011)
Marked Graph

Example

\[ A = \{ a = (1,1,-1), b = (-1,0,1) \} \]

\[ s^{in} = (1,0,1) \quad s^{out} = (1,\omega,1) \]

Example (Initial graph)

\[ x \quad y \]

\[ (\omega, \ldots, \omega) \]

\[ \forall a \in A \]
**Marked Graph Sequence**

- associated set of runs \( \Omega_\sigma \subseteq \text{Runs}_A(x, y) \)
- perfectness condition (aka \( \theta \) condition): decidable semantic condition ensuring \( \Omega_\sigma \neq \emptyset \)
- effective decomposition of imperfect sequences: 
  \[
  \Omega_\sigma = \bigcup_{\sigma' \in \text{decompose}(\sigma)} \Omega_{\sigma'}
  \]
**Marked Graph Sequence**

- associated set of runs $\Omega_\sigma \subseteq \text{Runs}_A(x, y)$
- perfectness condition (aka $\theta$ condition): decidable semantic condition ensuring $\Omega_\sigma \neq \emptyset$
- effective decomposition of imperfect sequences: $\Omega_\sigma = \bigcup_{\sigma' \in \text{decompose}(\sigma)} \Omega_{\sigma'}$
Marked Graph Sequence

- associated set of runs $\Omega_{\sigma} \subseteq \text{Runs}_A(x, y)$

- **perfectness** condition (aka $\theta$ condition): decidable semantic condition ensuring $\Omega_{\sigma} \neq \emptyset$

- effective decomposition of imperfect sequences: $\Omega_{\sigma} = \bigcup_{\sigma' \in \text{decompose}(\sigma)} \Omega_{\sigma'}$
**Marked Graph Sequence**

- associated set of runs $\Omega_\sigma \subseteq \text{Runs}_A(x, y)$
- perfectness condition (aka $\theta$ condition): decidable semantic condition ensuring $\Omega_\sigma \neq \emptyset$
- effective decomposition of imperfect sequences: $\Omega_\sigma = \bigcup_{\sigma' \in \text{decompose}(\sigma)} \Omega_{\sigma'}$
**KLMST Algorithm (Schematically)**

Construct a sequence $S_0, S_1, \ldots$ of finite sets of marked witness graph sequences with $\forall n$

$$\Omega_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} \Omega_{\sigma} = \text{Runs}_A(x, y)$$

- init $S_0$ is s.t. $\text{Runs}_A(x, y) = \Omega_0$

- $\forall n \ni$ if $S_n = \{\sigma\} \uplus S$ and $\neg \text{perfect}(\sigma)$
  $$S_{n+1} \overset{\text{def}}{=} S \cup (\text{decompose}(\sigma))$$

- otherwise stop: $\text{Runs}_A(x, y)$ is not empty

terminates via a ranking function $r$

$$\forall \sigma' \in \text{decompose}(\sigma) \cdot r(\sigma) > r(\sigma')$$
KLMST Algorithm (Schematically)

Construct a sequence $S_0, S_1, \ldots$ of finite sets of marked witness graph sequences with $\forall n$

$$\Omega_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} \Omega_\sigma = \text{Runs}_A(x, y)$$

init $S_0$ is s.t. $\text{Runs}_A(x, y) = \Omega_0$

$\forall n \quad \text{if } S_n = \{\sigma\} \cup S \text{ and } \neg \text{perfect(}\sigma\text{)}$

$$S_{n+1} \overset{\text{def}}{=} S \cup (\text{decompose(}\sigma\text{)}$$

$\triangleright$ otherwise stop: $\text{Runs}_A(x, y) \neq \emptyset$

terminates via a ranking function $r$

$$\forall \sigma' \in \text{decompose(}\sigma\text{)} . \ r(\sigma) > r(\sigma')$$
**KLMST Algorithm (Schematically)**

Construct a sequence $S_0, S_1, \ldots$ of finite sets of marked witness graph sequences with $\forall n$

$$\Omega_n \stackrel{\text{def}}{=} \bigcup_{\sigma \in S_n} \Omega_{\sigma} = \text{Runs}_A(x, y)$$

- **init** $S_0$ is s.t. $\text{Runs}_A(x, y) = \Omega_0$
- $\forall n \triangleright$ if $S_n = \{\sigma\} \cup S$ and $\neg \text{perfect}(\sigma)$
  $$S_{n+1} \stackrel{\text{def}}{=} S \cup (\text{decompose}(\sigma))$$  
  - otherwise stop: $\text{Runs}_A(x, y) \neq \emptyset$

terminates via a ranking function $r$

$$\forall \sigma' \in \text{decompose}(\sigma) . r(\sigma) > r(\sigma')$$
KLMST Algorithm (Schematically)

Construct a sequence $S_0, S_1, \ldots$ of finite sets of marked witness graph sequences with $\forall n$

$$\Omega_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} \Omega_\sigma = \text{Runs}_A(x, y)$$

**init** $S_0$ is s.t. $\text{Runs}_A(x, y) = \Omega_0$

$\forall n \quad \text{if } S_n = \{\sigma\} \cup S \text{ and } \neg \text{perfect}(\sigma)$

$$S_{n+1} \overset{\text{def}}{=} S \cup (\text{decompose}(\sigma))$$

$\quad \text{otherwise stop: } \text{Runs}_A(x, y) \neq \emptyset$

terminates via a ranking function $r$

$$\forall \sigma' \in \text{decompose}(\sigma) . \ r(\sigma) > r(\sigma')$$
Mysteries

1. Conceptual complexity
   - the complexity of the proofs (especially of Mayr, 1981)
   - wrap the result in mystery use of their original ideas has been made

2. Computational complexity
   - \textsc{ExpSpace}-hard problem (Lipton, 1976),
   - Ackermann lower bound on the KLMST algorithm (Müller, 1985)
   - no known upper bound
Mysteries

1. Conceptual complexity
   “the complexity of the two proofs (especially of Mayr, 1981) wrapped the result in mystery and no use of their original ideas has been made” (Lambert, 1992)

2. Computational complexity
   - EXPSPACE-hard problem (Lipton, 1976),
   - Ackermann lower bound on the KLMST algorithm (Müller, 1985)
   - No known upper bound
Mysteries

1. conceptual complexity

“the complexity of the two proofs (especially of Mayr, 1981) wrapped the result in mystery and no use of their original ideas has been made” (Lambert, 1992)

2. computational complexity

- ExSpace-hard problem (Lipton, 1976),
- Ackermann lower bound on the KLMST algorithm (Müller, 1985)
- no known upper bound
Mystery 1: Conceptual Complexity

Theorem (Decomposition Theorem)

The KLMST algorithm computes the ideal decomposition of

\[ \downarrow \text{Runs}_A(x, y) \overset{\text{def}}{=} \{ \rho' \in \text{PreRuns}_A \mid \exists \rho \in \text{Runs}_A(x, y). \rho' \sqsubseteq \rho \} \]

Deciphering the statement (upcoming slides)

- definition of a well quasi order (wqo) over preruns (Jančar, 1990)
- wqo ideals (Finkel and Goubault-Larrecq, 2009, 2012)
Theorem (Decomposition Theorem)

The KLMST algorithm computes the ideal decomposition of

\[
\downarrow \text{Runs}_A(x,y) \overset{\text{def}}{=} \{ \rho' \in \text{PreRuns}_A \mid \exists \rho \in \text{Runs}_A(x,y) . \rho' \sqsubseteq \rho \}
\]

Deciphering the statement (upcoming slides)

- definition of a well quasi order (wqo) over preruns (Jančar, 1990)
- wqo ideals (Finkel and Goubault-Larrecq, 2009, 2012)
**Theorem (Decomposition Theorem)**

The KLMST algorithm computes the ideal decomposition of

$$\downarrow \text{Runs}_A(x, y) \overset{\text{def}}{=} \{ \rho' \in \text{PreRuns}_A \mid \exists \rho \in \text{Runs}_A(x, y), \rho' \sqsubseteq \rho \}$$

**Deciphering the statement (upcoming slides)**

- definition of a well quasi order (wqo) over preruns (Jančar, 1990)
- wqo ideals (Finkel and Goubault-Larrecq, 2009, 2012)
Mystery 1: Conceptual Complexity

**Theorem (Decomposition Theorem)**

The KLMST algorithm computes the ideal decomposition of

$$\downarrow \text{Runs}_A(x, y) \overset{\text{def}}{=} \{ \rho' \in \text{PreRuns}_A \mid \exists \rho \in \text{Runs}_A(x, y). \rho' \sqsubseteq \rho \}$$

**Significance**

- entails decidability of VAS Reachability:
  \[ \text{Runs}_A(x, y) = \emptyset \text{ iff } \downarrow \text{Runs}_A(x, y) = \emptyset \]
- generalises Habermehl et al. (2010)’s result on the computability of downward-closures of VAS languages
- **template** for decidability proofs in extensions (unordered data nets, branching VAS, pushdown VAS, ...)
**Well quasi orders**

- **Downward closure**
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S. x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supsetneq D_1 \supsetneq \cdots\) of downwards-closed subsets of \(X\) is finite.

- **Examples**
  - finite sets with equality
  - \(\mathbb{N}\) with the naturals ordering (Dickson’s Lemma)
  - \(A \times B\) with the product ordering
  - \(A^*\) with scattered subword ordering (Higman’s Lemma)

over a quasi-order \((X, \leq)\)
Well quasi orders

- **Downward closure**
  
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S. \ x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supsetneq D_1 \supsetneq \cdots\) of downwards-closed subsets of \(X\) is finite.

- **Examples**
  - finite sets with equality
  - \(\mathbb{N}\) with the naturals ordering
  - \(A \times B\) with the product ordering (Dickson’s Lemma)
  - \(A^*\) with scattered subword ordering (Higman’s Lemma)
**Well Quasi Orders**

- **Downward closure**
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S. x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supsetneq D_1 \supsetneq \cdots\) of downwards-closed subsets of \(X\) is finite.

**Examples**
- finite sets with equality
- \(\mathbb{N}\) with the naturals ordering
- \(A \times B\) with the product ordering (Dickson’s Lemma)
- \(A^*\) with scattered subword ordering (Higman’s Lemma)

over a quasi-order \((X, \leq)\)
**Well quasi orders**

- **Downward closure**
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S. x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supset D_1 \supset \cdots\) of downwards-closed subsets of \(X\) is finite.

- **Examples**
  - finite sets with equality
  - \(\mathbb{N}\) with the naturals ordering (Dickson’s Lemma)
  - \(A \times B\) with the product ordering
  - \(A^*\) with scattered subword ordering (Higman’s Lemma)
WELL QUASI ORDERS

- **Downward closure**
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S. \ x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supseteq D_1 \supseteq \cdots\) of downwards-closed subsets of \(X\) is finite.

- **Examples**
  - finite sets with equality
  - \(\mathbb{N}\) with the naturals ordering
  - \(A \times B\) with the product ordering (Dickson’s Lemma)
  - \(A^*\) with scattered subword ordering (Higman’s Lemma)
Well quasi orders

- **Downward closure**
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S. x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supseteq D_1 \supseteq \cdots\) of downwards-closed subsets of \(X\) is finite.

- **Examples**
  - finite sets with equality
  - \(\mathbb{N}\) with the naturals ordering
  - \(A \times B\) with the product ordering (for \(A, B\) two wqos) (Dickson’s Lemma)
  - \(A^*\) with scattered subword ordering (Higman’s Lemma)
WELL QUASI ORDERS

- **Downward closure**
  \[ \downarrow S \overset{\text{def}}{=} \{ x \in X \mid \exists s \in S . x \leq s \} \]

- **Descending Chain Property**
  A quasi-order \((X, \leq)\) is a well quasi order if every descending chain \(D_0 \supseteq D_1 \supseteq \cdots\) of downwards-closed subsets of \(X\) is finite.

- **Examples**
  - finite sets with equality
  - \(\mathbb{N}\) with the naturals ordering
  - \(A \times B\) with the product ordering (Dickson’s Lemma)
  - \(A^*\) with scattered subword ordering (Higman’s Lemma)
Prerun Embedding

Construct the ordering $\preceq$ over preruns inductively; recall

$$\text{PreRuns}_A \overset{\text{def}}{=} \mathbb{N}^d \times (\mathbb{N}^d \times A \times \mathbb{N}^d)^* \times \mathbb{N}^d$$

**Example (Run Embedding $\preceq$)**

$$(3,3) \preceq (2,1) \preceq (3,2) \preceq (2,0) \preceq (3,1)$$

$$(1,0) \preceq (2,1)$$

**Lemma (Jančar, 1990)**

$(\text{PreRuns}_A, \preceq)$ is a wqo.
**PRERUN EMBEDDING**

Construct the ordering $\preceq$ over preruns inductively; recall

$$\text{PreRuns}_A \overset{\text{def}}{=} \mathbb{N}^d \times (\mathbb{N}^d \times A \times \mathbb{N}^d)^* \times \mathbb{N}^d$$

**EXAMPLE (RUN EMBEDDING $\preceq$)**

$$(3,3) \rightarrow (2,1) \rightarrow (3,2) \rightarrow (2,0) \rightarrow (3,1)$$

$$(1,0) \rightarrow (2,1)$$

**LEMMA (JANČAR, 1990)**

$(\text{PreRuns}_A, \preceq)$ is a wqo.
**Ideals as Canonical Bases**

**Lemma (Canonical Ideal Decomposition; Bonnet, 1975)**

Every downward-closed subset $D \subseteq X$ of a wqo $(X, \preceq)$ is the union of a unique finite family of incomparable (for the inclusion) ideals.

**Ideals as Canonical Bases**

**Lemma (Canonical Ideal Decomposition; Bonnet, 1975)**

Every downward-closed subset $D \subseteq X$ of a wqo $(X, \leq)$ is the union of a unique finite family of incomparable (for the inclusion) ideals.

A representation for (particular) prerun ideals with

$$I_\sigma \supseteq \downarrow \Omega_\sigma$$

**Theorem (Perfectness as Ideal Adherence)**

If $\sigma$ is perfect then $I_\sigma = \downarrow \Omega_\sigma$. 
KLMST Algorithm (Ideal View)

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

$$D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_\sigma \supseteq \downarrow \text{Runs}_A(x, y)$$

\text{init } D_0 \overset{\text{def}}{=} \text{PreRuns}_A

$\forall n \Rightarrow$ if $D_n = I \sqcup D$ and not perfect($I$),

$D_{n+1} \overset{\text{def}}{=} D \cup \text{decompose}(I)$

\text{otherwise stop:}

$D_n = \downarrow \text{Runs}_A(x, y)$
KLMST Algorithm (Ideal View)

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

$$D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_{\sigma} \supseteq \downarrow \text{Runs}_A(x, y)$$

init $D_0 \overset{\text{def}}{=} \text{PreRuns}_A$

$\forall n \quad \text{if } D_n = I \sqcup D \text{ and } \neg \text{perfect}(I)$,

$D_{n+1} \overset{\text{def}}{=} D \cup \text{decompose}(I)$

otherwise stop:

$D_n = \downarrow \text{Runs}_A(x, y)$
KLMST Algorithm (Ideal View)

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

$$D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_{\sigma} \supseteq \downarrow \text{Runs}_A(x, y)$$

\[ \begin{align*}
\text{init } & \quad D_0 \overset{\text{def}}{=} \text{PreRuns}_A \\
\forall n & \quad \begin{align*}
& \quad \text{if } D_n = I \sqcup D \text{ and } \\
& \quad \quad \neg \text{perfect}(I), \\
& \quad \quad D_{n+1} \overset{\text{def}}{=} D \cup \text{decompose}(I) \\
& \quad \quad \text{otherwise stop:} \\
& \quad \quad D_n = \downarrow \text{Runs}_A(x, y)
\end{align*}
\]
KLMST Algorithm (Ideal View)

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

$$D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_\sigma \supseteq \downarrow \text{Runs}_A(x, y)$$

init $D_0 \overset{\text{def}}{=} \text{PreRuns}_A$

$\forall n \quad \text{if } D_n = I \sqcup D \text{ and } \neg \text{perfect}(I),$ 

$D_{n+1} \overset{\text{def}}{=} D \cup \text{decompose}(I)$

\begin{itemize}
  \item otherwise stop:
  \end{itemize}

$D_n = \downarrow \text{Runs}_A(x, y)$
KLMST Algorithm (Ideal View)

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

$$D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_{\sigma} \supseteq \downarrow \text{Runs}_A(x, y)$$

init $D_0 \overset{\text{def}}{=} \text{PreRuns}_A$

$\forall n \hspace{1cm} \text{if } D_n = I \sqcup D \text{ and } \neg \text{perfect}(I)$,

$$D_{n+1} \overset{\text{def}}{=} D \sqcup \text{decompose}(I)$$

$\text{otherwise stop:}$

$$D_n = \downarrow \text{Runs}_A(x, y)$$
KLMST Algorithm (Ideal View)

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

$$D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_{\sigma} \supseteq \downarrow \text{Runs}_A(x, y)$$

init $D_0 \overset{\text{def}}{=} \text{PreRuns}_A$

$\forall n \quad \text{if } D_n = I \sqcup D$ and $\neg \text{perfect}(I)$,

$$D_{n+1} \overset{\text{def}}{=} D \sqcup \text{decompose}(I)$$

$\quad$ otherwise stop:

$$D_n = \downarrow \text{Runs}_A(x, y)$$
**KLMST Algorithm (Ideal View)**

Construct a sequence $D_0, D_1, \ldots$ of downwards-closed sets, represented as finite sets of ideals, with $\forall n$

\[
D_n \overset{\text{def}}{=} \bigcup_{\sigma \in S_n} I_\sigma \supseteq \downarrow \text{Runs}_A(x, y)
\]

- **init** $D_0 \overset{\text{def}}{=} \text{PreRuns}_A$
- $\forall n \quad \text{if } D_n = I \sqcup D \text{ and } \neg \text{perfect}(I)$,
  
  $D_{n+1} \overset{\text{def}}{=} D \cup \text{decompose}(I)$

- otherwise stop:
  
  $D_n = \downarrow \text{Runs}_A(x, y)$
**Mystery 2: Computational Complexity**

**Theorem (Upper Bound Theorem)**

VAS Reachability is in $F_{\omega^3}$.

- uses a length function theorem for ranking functions (S., 2014)
- ...which provides bounds in fast-growing complexity classes $(F_{\alpha})_{\alpha}$

Diagram:
- **Elementary**
- **Primitive Recursive**
- **Multiply Recursive**
- $F_3 = \text{Tower}$
- $F_{\omega} = \text{Ack}$
- $F_{\omega^\omega} = \text{HAck}$
Mystery 2: Computational Complexity

Theorem (Upper Bound Theorem)

VAS Reachability is in $F_\omega^3$.

- **uses a length function theorem** for ranking functions (S., 2014)
- ...which provides bounds in **fast-growing complexity classes** $(F_\alpha)_\alpha$
Concluding Remarks

- ideals as an algorithmic tool to work with downward-closed sets
- new understanding of the KLMST decomposition
- extension to other models (BVASS, PDVAS, ...)?
- immense complexity gap: EXPSPACE vs. $F_{\omega^3}$
- only known tight case: PSPACE-complete $d = 2$ + control states (as presented by Blondin et al., 2015, this morning)
Concluding Remarks

- Ideals as an algorithmic tool to work with downward-closed sets
- New understanding of the KLMST decomposition extension to other models (BVASS, PDVAS,...)?
- Immense complexity gap: \( \text{ExpSpace} \) vs. \( F_{\omega^3} \)
- Only known tight case: \( \text{PSPACE-complete} \) \( d = 2 + \) control states (as presented by Blondin et al., 2015, this morning)
Concluding Remarks

- Ideals as an algorithmic tool to work with downward-closed sets
- New understanding of the KLMST decomposition
  - Extension to other models (BVASS, PDVAS, ...)?
- Immense complexity gap: \( \text{ExpSpace} \) vs. \( F_{\omega^3} \)
  - Only known tight case: \( \text{PSpace} \)-complete \( d = 2 + \) control states (as presented by Blondin et al., 2015, this morning)


A Bit of Magic...

**Theorem (Length Function Theorem, S., 2014)**

$(g, n)$-controlled decreasing sequences $\alpha_0, \alpha_1, \ldots$ of ordinals $< \alpha$ with $n \geq |\alpha|$ are of length bounded by $g^\alpha(n)$ in the Hardy hierarchy.

**Claim (KLMST control, using Figueira et al., 2011)**

In the KLMST algorithm, the sequence of ranks along any branch is controlled by $(H^{\omega^{d+1}}, |A|)$. As a result, the KLMST algorithm runs in space $(H^{\omega^{d+1}})^{\omega^3} (|A|)$, which is in $F_{\omega^3}$. 
**Definition (Ordinal Norm)**

For an ordinal $\alpha < \varepsilon_0$, define the norm $|\alpha|$ as the maximal coefficient appearing in its Cantor normal form $\alpha = \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_n} \cdot c_n$:

$$|\alpha| \overset{\text{def}}{=} \max_{1 \leq j \leq n} (c_j, |\alpha_j|).$$

**Definition (Controlled Sequence)**

Let $g: \mathbb{N} \rightarrow \mathbb{N}$ strictly monotone and $n \in \mathbb{N}$. A sequence of ordinals $\alpha_0, \alpha_1, \ldots$ is $(g, n)$-controlled if

$$\forall i. |\alpha_i| \leq g^i(n).$$

In particular, $|\alpha_0| \leq n$. 
A BIT OF MAGIC: HARDY FUNCTIONS

Ordinal-indexed hierarchy of functions $h^\alpha : \mathbb{N} \to \mathbb{N}$

**Definition**
Fix $h: \mathbb{N} \to \mathbb{N}$ strictly monotone:

\[
\begin{align*}
h^0(x) & \overset{\text{def}}{=} x & h^{\alpha+1}(x) & \overset{\text{def}}{=} h^\alpha(h(x)) & h^\lambda(x) & \overset{\text{def}}{=} h^{\lambda(x)}(x)
\end{align*}
\]

where $\lambda(0) < \lambda(1) < \cdots < \lambda$ is the standard fundamental sequence for the limit ordinal $\lambda$, e.g. $\omega(x) = x + 1$, $\omega^2(x) = \omega \cdot (x + 1)$, $\omega^\omega(x) = \omega^{x+1}$.

**Example**
For instance for $H(x) \overset{\text{def}}{=} x + 1$:

\[
\begin{align*}
H^\omega(x) & = 2x + 1 & H^{\omega^2}(x) & = 2^{x+1}(x + 1) - 1 \\
H^{\omega^3}(x) & \approx 2 \cdot \ldots \cdot 2 \text{ } x \text{ times} & H^{\omega^\omega}(x) & \approx \text{ackermann}(x)
\end{align*}
\]
For $\alpha \geq 3$:

$$\mathcal{F}_{<\alpha} \overset{\text{def}}{=} \bigcup_{\beta < \omega^\alpha} \text{FDTIME}(H^\beta(n)), \quad F_\alpha \overset{\text{def}}{=} \bigcup_{p \in \mathcal{F}_{<\alpha}} \text{DTIME}(H^{\omega^\alpha}(p(n)))$$

**References**

- A Bit of Magic

**A Bit of Magic: Fast-Growing Complexity**

**Elementary**

**Primitive Recursive**

**Multiply Recursive**

$F_3 = \text{Tower}$

$F_\omega = \text{Ack}$

$F_{\omega^\omega} = \text{HAck}$
For $\alpha \geq 3$:

$$\mathcal{F}_{<\alpha} \overset{\text{def}}{=} \bigcup_{\beta < \omega^\alpha} \text{FDTIME}(H^\beta(n)),$$

$$F_\alpha \overset{\text{def}}{=} \bigcup_{p \in \mathcal{F}_{<\alpha}} \text{DTIME}(H^{\omega^\alpha}(p(n))).$$