Formal Language Techniques for Space Lower Bounds

Philipp Kuinke

February 23, 2018
Contained in Sánchez Villaamil’s Phd Thesis 2017
Treewidth
Dynamic Programming

Use treewidth structure to traverse the graph
Dynamic Programming
Dynamic Programming
Dynamic Programming

\[
\begin{array}{cccc}
  x_1 & x_2 & \ldots & x_w \\
  R & R & \ldots & G \\
  B & G & \ldots & R \\
  G & G & \ldots & B \\
  \vdots & \vdots & \ddots & \vdots \\
  G & R & \ldots & B
\end{array}
\]
Dynamic Programming
Dynamic Programming
Dynamic Programming

The runtime of dynamic programming algorithms depends on the table sizes!
Dynamic Programming

Common properties of DP-algorithms we formalize
Dynamic Programming

Common properties of DP-algorithms we formalize

1. They do a single pass over the decomposition;
Dynamic Programming

Common properties of DP-algorithms we formalize

1. They do a single pass over the decomposition;
2. they use $O(f(w) \log^{O(1)} n)$ space; and
Dynamic Programming

Common properties of DP-algorithms we formalize

1. They do a single pass over the decomposition;
2. they use $O(f(w) \log^{O(1)} n)$ space; and
3. they do not modify or rearrange the decomposition.
Dynamic Programming

**Definition (DPTM)**

A Dynamic Programming Turing Machine (DPTM) is a Turing Machine with an input read-only tape, whose head moves only in one direction and a separate working tape. It only accepts well-formed instances as inputs.
Boundaried Graphs

**Definition**

An $s$-boundaried graph $G$ is a graph with $s$ distinguished vertices, called the boundary.

\[ X \]

\[ G_1 \]
Definition

$G_1 \oplus G_2$ is the disjoint union of two $s$-boundaried graphs merged at the boundary.
Definition

$G_s$ is the set of all $s$-boundaried graphs.
Interpret Problem as a language Π, i.e. \( G \in \Pi \) if and only if \( G \) is a yes-instance.
Myhill-Nerode Families
Definition (Myhill-Nerode family)

A set $\mathcal{H} \subseteq G_s$ is an s-Myhill-Nerode family for a DP language $\Pi$ if

1. For every subset $I \subseteq H$ there exists an $s$-boundaried graph $G_I$ with bounded size, such that for every $H \in H$ it holds that $G_I \oplus H \notin \Pi$ if $H \notin I$.
2. For every $H \in H$ it holds that $H$ has bounded size.
Myhill-Nerode Families

Definition (Myhill-Nerode family)

A set $\mathcal{H} \subseteq G_s$ is an s-Myhill-Nerode family for a DP language $\Pi$ if

1. For every subset $\mathcal{I} \subseteq \mathcal{H}$ there exists an s-boundaried graph $G_{\mathcal{I}}$ with bounded size, such that for every $H \in \mathcal{H}$ it holds that

   $G_{\mathcal{I}} \oplus H \notin \Pi \iff H \in \mathcal{I}$
Myhill-Nerode Families

Definition (Myhill-Nerode family)

A set $\mathcal{H} \subseteq G_s$ is an s-Myhill-Nerode family for a DP language $\Pi$ if

1. For every subset $\mathcal{I} \subseteq \mathcal{H}$ there exists an $s$-boundaried graph $G_\mathcal{I}$ with bounded size, such that for every $H \in \mathcal{H}$ it holds that

$$G_\mathcal{I} \oplus H \notin \Pi \iff H \in \mathcal{I}$$

2. For every $H \in \mathcal{H}$ it holds that $H$ has bounded size.
Myhill-Nerode Families

Definition (Myhill-Nerode family)
A set $\mathcal{H} \subseteq \mathcal{G}_s$ is an s-Myhill-Nerode family for a DP language $\Pi$ if

1. For every subset $\mathcal{I} \subseteq \mathcal{H}$ there exists an s-boundaried graph $G_{\mathcal{I}}$ with $|G_{\mathcal{I}}| = |\mathcal{H}| \log^{O(1)} \mathcal{H}$, such that for every $H \in \mathcal{H}$ it holds that

   $$G_{\mathcal{I}} \oplus H \notin \Pi \iff H \in \mathcal{I}$$

2. For every $H \in \mathcal{H}$ it holds that $|H| = |\mathcal{H}| \log^{O(1)} \mathcal{H}$. 
Myhill-Nerode Families

\[ G_I \oplus H_1 \in \Pi \]

\[ G_I \oplus H_2 \notin \Pi \]
Lemma ([Sánchez Villaamil ’17])

Let $\epsilon > 0$ and $\Pi$ be a DP decision problem such that for every $s$ there exists an $s$-Myhill-Nerode family $\mathcal{H}$ for $\Pi$ of size $c^s$ and width $tw(\mathcal{H}) = s$. Then no DPTM can decide $\Pi$ using space $O((c - \epsilon)^k \log n)$, where $n$ is the size of the input and $k$ the treewidth of the input.
DPTM bounds

Lemma ([Sánchez Villaamil ’17])

Let $\epsilon > 0$ and $\Pi$ be a DP decision problem such that for every $s$ there exists an $s$-Myhill-Nerode family $\mathcal{H}$ for $\Pi$ of size $c^s/f(s)$, where $f(s) = s^{O(1)} \cap \Theta(1)$ and width $tw(\mathcal{H}) = s + o(s)$. Then no DPTM can decide $\Pi$ using space $O((c - \epsilon)^k \log^{O(1)} n)$, where $n$ is the size of the input and $k$ the treewidth of the input.
3-Coloring

- **Input:** A Graph $G$
- **$k$:** The treewidth of $G$
- **Question:** Can $G$ be colored with 3 colors?
Coloring Gadget
Coloring Gadget
Coloring Gadget
Coloring Gadget
Coloring Gadget
Coloring Gadget

\[ \mathcal{U} \]

\[ u_1 \]

\[ u_2 \]
The Graph $\Gamma_X$
Enforcing Colorings with $H_X$
No-Instances

This is not 3-colorable.
No-Instances

This is **not** 3-colorable.
Yes-Instances

This is 3-colorable.
Yes-Instances

This is 3-colorable.
Myhill-Nerode Families

\[
\Gamma_X \oplus H_X \not\in \Pi
\]
Myhill-Nerode Families

\[ \Gamma_X \oplus H_X \notin \Pi \]

\[ \Gamma_X \oplus H_{X'} \in \Pi, \text{ for } (X \neq X') \]
Myhill-Nerode Families

- \( \Gamma_X \oplus H_X \not\in \Pi \)

- \( \Gamma_X \oplus H_{X'} \in \Pi \), for \( X \neq X' \)

- \( G_I = \oplus_{H_X \in I} \Gamma_X \)
Myhill-Nerode Families

- $\Gamma_X \oplus H_X \not\in \Pi$

- $\Gamma_X \oplus H_{X'} \in \Pi$, for ($X \neq X'$)

- $G_{\mathcal{I}} = \bigoplus_{H_X \in \mathcal{I}} \Gamma_X$

- $G_{\mathcal{I}} \oplus H_X \in \Pi \iff H_X \not\in \Pi$
Myhill-Nerode Families
Myhill-Nerode Families
Myhill-Nerode Families
Myhill-Nerode Families
Myhill-Nerode Families

\[ G_I \oplus H_X \in \Pi \iff H_X \notin \Pi \]
We can generate $3^w/6$ such graphs.
We can generate a Myhill-Nerode family of index $3^w/6$. 
We cannot use $O((3 - \epsilon)^w \cdot \log n)$ space for a dynamic programming algorithm.
Theorem ([Sánchez Villaamil '17])

No DPTM solves 3-COLORING on a treewidth-decomposition of width $w$ with space bounded by $O((3 - \epsilon)^w \cdot \log^{O(1)} n)$. 

Obtained result
Further results

Theorem ([Sánchez Villaamil ’17])

No DPTM solves **Vertex Cover** on a treewidth-decomposition of width \( w \) with space bounded by \( O((2 - \epsilon)^w \cdot \log^{O(1)} n) \).

Theorem ([Sánchez Villaamil ’17])

No DPTM solves **Dominating Set** on a treewidth-decomposition of width \( w \) with space bounded by \( O((3 - \epsilon)^w \cdot \log^{O(1)} n) \).
Not Captured

- Compression.
- Algebraic techniques.
- Preprocessing to compute optimal traversal.
- Branching instead of DP
The end