Identities in plactic and related monoids

António Malheiro

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA
&
CMA FCT.UNL
centro de matemática e aplicações

NOVA id FCT
Associação para a Inovação e Desenvolvimento da FCT

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Novi Sad, June 15, 2017

(Joint work with various subsets of
{Alan J. Cain (CMA),
Georg Klein, Łukasz Kubat, Jan Okniński (Warsaw),
Fábio Silva (Lisboa)})

Acknowledgement:
This work was supported by CMA within the projects
UID/MAT/00297/2013
PTDC/MHC-FIL/2583/2014
financed by 'Fundação para a Ciência e a Tecnologia'

Young tableaux & Schensted's algorithm

Let $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ and let $\mathcal{A}_n = \{1 < 2 < 3 < \ldots < n\}$.

1	1	3	4	4
2	3	6		
5				

- Rows weakly increasing left to right.
- Columns strictly increasing top to bottom.
- Longer columns to the left.

Schensted's algorithm To insert $a \in \mathcal{A}$ into a tableau:

1. If adding a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all lower rows.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$:

- Start with an empty tableau and insert u_1, then u_2, ..., finally u_k.
- Call the resulting tableau $\mathrm{P}_{\mathrm{plac}}(u)$. For example, $\mathrm{P}_{\mathrm{plac}}(2531613443)$ is the tableau above.

Young tableaux & Schensted's algorithm

Let $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ and let $\mathcal{A}_n = \{1 < 2 < 3 < \ldots < n\}$.

1	1	3	4	4	$\leftarrow 3$
2	3	6			
5					

- Rows weakly increasing left to right.
- Columns strictly increasing top to bottom.
- Longer columns to the left.

Schensted's algorithm To insert $a \in \mathcal{A}$ into a tableau:

1. If adding a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all lower rows.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$:

- Start with an empty tableau and insert u_1, then u_2, ..., finally u_k.
- Call the resulting tableau $P_{\text{plac}}(u)$. For example, $P_{\text{plac}}(2531613443)$ is the tableau above.

Young tableaux & Schensted's algorithm

Let $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ and let $\mathcal{A}_n = \{1 < 2 < 3 < \ldots < n\}$.

1	1	3	3	4
2	3	6	$\leftarrow 4$	
5				

- Rows weakly increasing left to right.
- Columns strictly increasing top to bottom.
- Longer columns to the left.

Schensted's algorithm To insert $a \in \mathcal{A}$ into a tableau:

1. If adding a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all lower rows.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$:

- Start with an empty tableau and insert u_1, then u_2, ..., finally u_k.
- Call the resulting tableau $P_{\text{plac}}(u)$. For example, $P_{\text{plac}}(2531613443)$ is the tableau above.

Young tableaux & Schensted's algorithm

Let $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ and let $\mathcal{A}_n = \{1 < 2 < 3 < \ldots < n\}$.

1	1	3	3	4
2	3	4		
5	$\leftarrow 6$			

- Rows weakly increasing left to right.
- Columns strictly increasing top to bottom.
- Longer columns to the left.

Schensted's algorithm To insert $a \in \mathcal{A}$ into a tableau:

1. If adding a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all lower rows.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$:

- Start with an empty tableau and insert u_1, then u_2, ..., finally u_k.
- Call the resulting tableau $P_{\text{plac}}(u)$. For example, $P_{\text{plac}}(2531613443)$ is the tableau above.

Young tableaux & Schensted's algorithm

Let $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ and let $\mathcal{A}_n = \{1 < 2 < 3 < \ldots < n\}$.

1	1	3	3	4
2	3	4		
5	6			

- Rows weakly increasing left to right.
- Columns strictly increasing top to bottom.
- Longer columns to the left.

Schensted's algorithm To insert $a \in \mathcal{A}$ into a tableau:

1. If adding a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all lower rows.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$:

- Start with an empty tableau and insert u_1, then u_2, …, finally u_k.
- Call the resulting tableau $P_{\text{plac}}(u)$. For example, $P_{\text{plac}}(2531613443)$ is the tableau above.

Young tableaux & Schensted's algorithm

Let $\mathcal{A} = \{1 < 2 < 3 < \ldots\}$ and let $\mathcal{A}_n = \{1 < 2 < 3 < \ldots < n\}$.

1	1	3	3	4
2	3	4		
5	6			

- Rows weakly increasing left to right.
- Columns strictly increasing top to bottom.
- Longer columns to the left.

Schensted's algorithm To insert $a \in \mathcal{A}$ into a tableau:

1. If adding a to the end of the top row gives a tableau, this is the result.
2. Otherwise, let b the leftmost symbol of the top row such that $b > a$. Replace b with a ('bumping b').
3. Recursively insert b into the tableau formed by all lower rows.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$:

- Start with an empty tableau and insert u_1, then u_2, ..., finally u_k.
- Call the resulting tableau $\mathsf{P}_{\mathsf{plac}}(u)$. For example, $\mathsf{P}_{\mathsf{plac}}(2531613443)$ is the tableau above.

The plactic monoid

Schensted's algorithm computes a tableau $\mathsf{P}_{\mathsf{plac}}(u)$ for $u \in \mathcal{A}^*$.

Define $u \equiv_{\mathsf{plac}} v \iff \mathsf{P}_{\mathsf{plac}}(u) = \mathsf{P}_{\mathsf{plac}}(v)$.

Theorem (Knuth 1970)

The relation $\equiv_{\mathsf{plac}}$ is a congruence on $\mathcal{A}^*$.

- $\mathsf{plac} = \mathcal{A}^*/\equiv_{\mathsf{plac}}$ is the plactic monoid.
- $\mathsf{plac}_n = \mathcal{A}_n^*/\equiv_{\mathsf{plac}}$ is the plactic monoid of rank n.

Clearly,

$$\mathsf{plac}_1 \hookrightarrow \mathsf{plac}_2 \hookrightarrow \ldots \hookrightarrow plac_n \hookrightarrow \mathsf{plac}_{n+1} \hookrightarrow \ldots \hookrightarrow \mathsf{plac}.$$

and

$$\mathsf{plac} = \bigcup_{n \in \mathbb{N}} \mathsf{plac}_n.$$

Identities

- An identity is a formal equality $u = v$, where $u, v \in X^*$.
- A monoid M satisfies $u = v$ if substituting any element of M for each symbol in X gives an equality that holds in M.

For example,

- Any commutative monoid satisfies $xy = yx$.
- Any nilpotent group of class 2 satisfies $xyzyx = yxzxy$ [Neumann & Taylor 1963].

An identity is trivial if u and v are the same word; otherwise non-trivial.

- $xy = xy$ is trivial.
- $xy = yx$ and $xyzyx = yxzxy$ are non-trivial.

Questions

- Does plac satisfy a non-trivial identity?
- Does each plac_n satisfy a non-trivial identity?

Chinese monoid

- The Chinese monoid is also defined by an insertion algorithm.
- The Chinese monoid is related to the plactic monoid by its growth type.
- plac_2 is isomorphic to the Chinese monoid of rank 2.

Proposition (Jaszuńska & Okniński)

The Chinese monoid embeds into a direct product of copies of the bicyclic monoid and the infinite cyclic group.

Proposition (Adian)

The bicyclic monoid satisfies $xyyxxyxyyx = xyyxyxxyyx$ ('Adian's identity').

Corollary

The Chinese monoid satisfies Adian's identity.

Corollary

plac_2 satisfies Adian's identity.

Identities for plac_3

Adian's identity $xyyxxyxyyx = xyyxyxxyyx$

Proposition (Kubat & Okniński)

plac_3 satisfies $pqqpqp = pqpqqp$, where p and q are the left and right sides of Adian's identity. plac_3 does not satisfy Adian's identity.

- Use detailed calculations using normal forms.

Proposition (Izhakian)

plac_3 satisfies $xyxy^2x^2yxyxyxy^2x^2y = xyxy^2x^2yyxxyxy^2x^2y$.

- Use a (complicated) representation in the monoid of 3×3 upper-triangular tropical matrices.

Proposition (Cain, Klein, Kubat, M., Okniński)

plac_3 satisfies $pqppq = pqqpq$, where p and q are left and right sides of Adian's identity.

- Use a (simpler) representation in the monoid of 3×3 upper-triangular tropical matrices.

Identities for plac and plac_n

Theorem (Cain, Klein, Kubat, M., Okniński)

plac does not satisfy any non-trivial identity.

Proposition (CKKMO)

plac_n does not satisfy any non-trivial identity of length less than or equal to n.

Theorem (Schensted 1961)

$$\text{Number of columns in } P_{\text{plac}}(u) = \begin{cases} \text{Length of the longest weakly} \\ \text{increasing subsequence of } u; \end{cases}$$

$$\text{Number of rows in } P_{\text{plac}}(u) = \begin{cases} \text{Length of the longest strictly} \\ \text{decreasing subsequence of } u. \end{cases}$$

Identities for plac and plac_n

Theorem (Cain, Klein, Kubat, M., Okniński)

plac does not satisfy any non-trivial identity.

Proposition (CKKMO)

plac_n does not satisfy any non-trivial identity of length less than or equal to n.

Theorem (Schensted 1961)

$$\text{Number of columns in } P_{\text{plac}}(u) = \begin{cases} \text{Length of the longest weakly} \\ \text{increasing subsequence of } u; \end{cases}$$

$$\text{Number of rows in } P_{\text{plac}}(u) = \begin{cases} \text{Length of the longest strictly} \\ \text{decreasing subsequence of } u. \end{cases}$$

Identities for plac and plac_n

Theorem (Cain, Klein, Kubat, M., Okniński)

plac does not satisfy any non-trivial identity.

Proposition (CKKMO)

plac_n does not satisfy any non-trivial identity of length less than or equal to n.

Theorem (Schensted 1961)

$$\text{Number of columns in } P_{\text{plac}}(u) = \begin{cases} \text{Length of the longest weakly} \\ \text{increasing subsequence of } u; \end{cases}$$

$$\text{Number of rows in } P_{\text{plac}}(u) = \begin{cases} \text{Length of the longest strictly} \\ \text{decreasing subsequence of } u. \end{cases}$$

Proof that plac_n satisfies no identity of length at most n

Suppose plac_n satisfies $u(x, y) = v(x, y)$ of length n.

Assume

$$u = u_1 \cdots u_{j-1} x u_{j+1} \cdots u_n$$
$$v = v_1 \cdots v_{j-1} y v_{j+1} \cdots v_n$$

Let $s = 12 \cdots n \in \mathcal{A}_n^*$,
$t = 12 \cdots (n - j)(n - j + 2) \cdots n \in \mathcal{A}_n^*$ (miss out $n - j + 1$).

So the tableaux $P_{\text{plac}}\big(u(s, t)\big)$ and $P_{\text{plac}}\big(v(s, t)\big)$ are equal.

Longest decreasing subsequences:

In $u(s, t)$: $u_1 \quad u_2 \quad \cdots \quad u_{j-1} \quad x \quad u_{j+1} \quad \cdots \quad u_{n-1} \quad u_n$

Proof that plac_n satisfies no identity of length at most n

Suppose plac_n satisfies $u(x,y) = v(x,y)$ of length n.

Assume

$$u = u_1 \cdots u_{j-1}\, x\, u_{j+1} \cdots u_n$$
$$v = v_1 \cdots v_{j-1}\, y\, v_{j+1} \cdots v_n$$

Let $s = 12\cdots n \in \mathcal{A}_n^*$,
$t = 12\cdots(n-j)(n-j+2)\cdots n \in \mathcal{A}_n^*$ (miss out $n-j+1$).

So the tableaux $\mathrm{P}_{\mathrm{plac}}\big(u(s,t)\big)$ and $\mathrm{P}_{\mathrm{plac}}\big(v(s,t)\big)$ are equal.

Longest decreasing subsequences:

	n	$n-1$		$n-j$	$n-j+1$	$n-j+2$		2	1
In $u(s,t)$:	u_1	u_2	$\cdots$	u_{j-1}	x	u_{j+1}	$\cdots$	u_{n-1}	u_n

Proof that plac_n satisfies no identity of length at most n

Suppose plac_n satisfies $u(x, y) = v(x, y)$ of length n.

Assume

$$u = u_1 \cdots u_{j-1} x u_{j+1} \cdots u_n$$
$$v = v_1 \cdots v_{j-1} y v_{j+1} \cdots v_n$$

Let $s = 12 \cdots n \in \mathcal{A}_n^*$,
$t = 12 \cdots (n-j)(n-j+2) \cdots n \in \mathcal{A}_n^*$ (miss out $n - j + 1$).

So the tableaux $\mathsf{P}_{\text{plac}}\big(u(s,t)\big)$ and $\mathsf{P}_{\text{plac}}\big(v(s,t)\big)$ are equal.

Longest decreasing subsequences:

	n	$n-1$		$n-j$	$n-j+1$	$n-j+2$		2	1
In $u(s,t)$:	u_1	u_2	$\cdots$	u_{j-1}	x	u_{j+1}	$\cdots$	u_{n-1}	u_n
In $v(s,t)$:	v_1	v_2	$\cdots$	v_{j-1}	y	v_{j+1}	$\cdots$	v_{n-1}	v_n

Proof that plac_n satisfies no identity of length at most n

Suppose plac_n satisfies $u(x,y) = v(x,y)$ of length n.

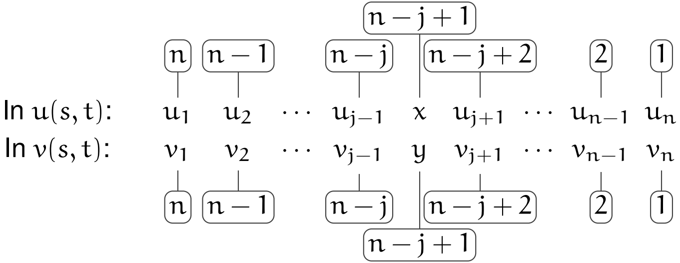
Assume

$$u = u_1 \cdots u_{j-1} x u_{j+1} \cdots u_n$$
$$v = v_1 \cdots v_{j-1} y v_{j+1} \cdots v_n$$

Let $s = 12 \cdots n \in \mathcal{A}_n^*$,
$t = 12 \cdots (n-j)(n-j+2) \cdots n \in \mathcal{A}_n^*$ (miss out $n-j+1$).

So the tableaux $P_{\mathsf{plac}}\big(u(s,t)\big)$ and $P_{\mathsf{plac}}\big(v(s,t)\big)$ are equal.

Longest decreasing subsequences:



Proof that plac_n satisfies no identity of length at most n

Suppose plac_n satisfies $u(x,y) = v(x,y)$ of length n.

Assume

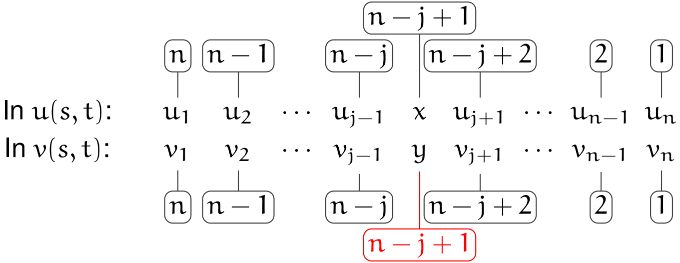
$$\begin{aligned} u &= u_1 \cdots u_{j-1}\, x\, u_{j+1} \cdots u_n \\ v &= v_1 \cdots v_{j-1}\, y\, v_{j+1} \cdots v_n \end{aligned}$$

Let $s = 12\cdots n \in \mathcal{A}_n^*$,

$t = 12 \cdots (n-j)(n-j+2)\cdots n \in \mathcal{A}_n^*$ (miss out $n-j+1$).

So the tableaux $\mathrm{P}_{\mathrm{plac}}\big(u(s,t)\big)$ and $\mathrm{P}_{\mathrm{plac}}\big(v(s,t)\big)$ are equal.

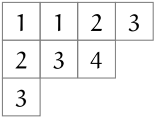
Longest decreasing subsequences:



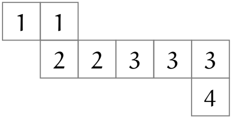
‘Plactic-like’ monoids

A family of monoids whose elements can be viewed as combinatorial objects:

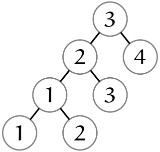
Plactic monoid
Young tableaux



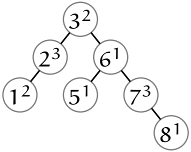
Hypoplactic monoid
Quasi-ribbon tableaux



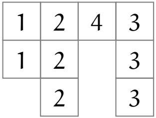
Sylvester monoid
Binary search trees



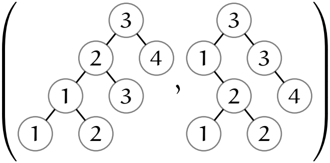
Taiga monoid
BSTs with multiplicities



Stalactite monoid
Stalactite tableaux

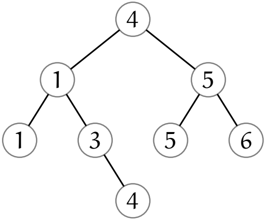


Baxter monoid
Pairs of twin binary search trees



Binary search trees and leaf insertion

Binary search tree (BST):



To insert a into a BST T:

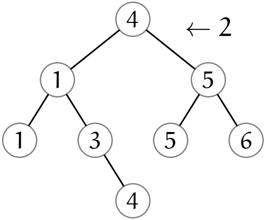
- Add a as a leaf node in the unique position that yields a BST.

For a word $u = u_k u_{k-1} \cdots u_1 \in \mathcal{A}^*$.

- Start with an empty BST and insert u_1, then u_2, …, finally u_k.
- Call the resulting BST $\mathrm{P}_{\mathrm{sylv}}(u)$. For example, $\mathrm{P}_{\mathrm{sylv}}(214536154)$ is the tree above.

Binary search trees and leaf insertion

Binary search tree (BST):



To insert a into a BST T:

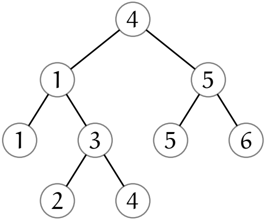
- Add a as a leaf node in the unique position that yields a BST.

For a word $u = u_k u_{k-1} \cdots u_1 \in \mathcal{A}^*$.

- Start with an empty BST and insert u_1, then u_2, ..., finally u_k.
- Call the resulting BST $P_{\text{sylv}}(u)$. For example, $P_{\text{sylv}}(214536154)$ is the tree above.

Binary search trees and leaf insertion

Binary search tree (BST):



To insert a into a BST T:

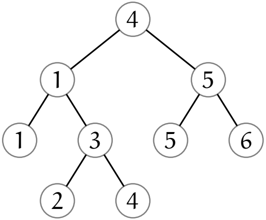
- Add a as a leaf node in the unique position that yields a BST.

For a word $u = u_k u_{k-1} \cdots u_1 \in \mathcal{A}^*$.

- Start with an empty BST and insert u_1, then u_2, …, finally u_k.
- Call the resulting BST $P_{\mathrm{sylv}}(u)$. For example, $P_{\mathrm{sylv}}(214536154)$ is the tree above.

Binary search trees and leaf insertion

Binary search tree (BST):



To insert a into a BST T:

- Add a as a leaf node in the unique position that yields a BST.

For a word $u = u_k u_{k-1} \cdots u_1 \in \mathcal{A}^*$.

- Start with an empty BST and insert u_1, then u_2, ..., finally u_k.
- Call the resulting BST $P_{\text{sylv}}(u)$. For example, $P_{\text{sylv}}(214536154)$ is the tree above.

The sylvester monoid

The insertion algorithm computes a tree $\mathrm{P}_{\mathrm{sylv}}(u)$ for $u \in \mathcal{A}^*$.

Define $u \equiv_{\mathrm{sylv}} v \iff \mathrm{P}_{\mathrm{sylv}}(u) = \mathrm{P}_{\mathrm{sylv}}(v)$.

Theorem (Hivert et al. 2005)

The relation $\equiv_{\mathrm{sylv}}$ is a congruence on $\mathcal{A}^*$.

- sylv $= \mathcal{A}^*/{\equiv_{\mathrm{sylv}}}$ is the sylvester monoid.
- $\mathrm{sylv}_n = \mathcal{A}_n^*/{\equiv_{\mathrm{sylv}}}$ is the sylvester monoid of rank n.

Theorem (Cain, M.)

sylv satisfies the identity $xyxy = yxxy$.
This is the unique shortest non-trivial identity satisfied by sylv.

- 'unique' up to renaming variables and swapping the two sides.

The sylvester monoid

The insertion algorithm computes a tree $\mathsf{P}_{\mathsf{sylv}}(u)$ for $u \in \mathcal{A}^*$.

Define $u \equiv_{\mathsf{sylv}} v \iff \mathsf{P}_{\mathsf{sylv}}(u) = \mathsf{P}_{\mathsf{sylv}}(v)$.

Theorem (Hivert et al. 2005)

The relation $\equiv_{\mathsf{sylv}}$ is a congruence on $\mathcal{A}^*$.

- sylv $= \mathcal{A}^*/{\equiv_{\mathsf{sylv}}}$ is the sylvester monoid.
- $\mathsf{sylv}_n = \mathcal{A}_n^*/{\equiv_{\mathsf{sylv}}}$ is the sylvester monoid of rank n.

Theorem (Cain, M.)

sylv satisfies the identity $xyxy = yxxy$.
This is the unique shortest non-trivial identity satisfied by sylv.

- 'unique' up to renaming variables and swapping the two sides.

Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$

$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$

Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

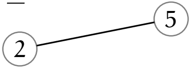
$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$

$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$

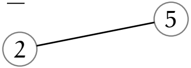
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



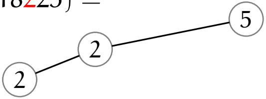
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



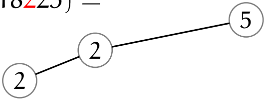
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



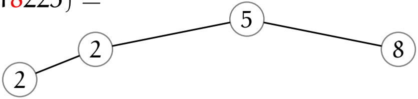
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



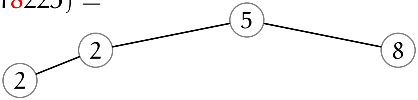
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



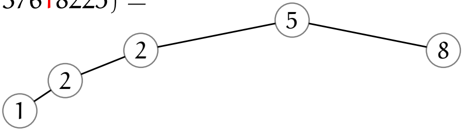
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



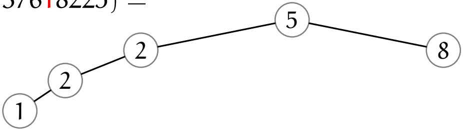
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



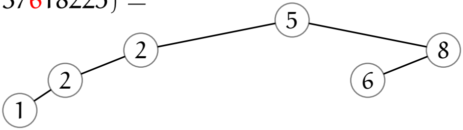
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



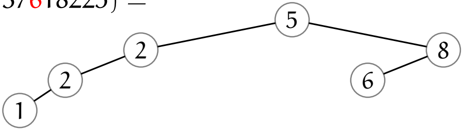
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37{\color{red}6}18225) =$



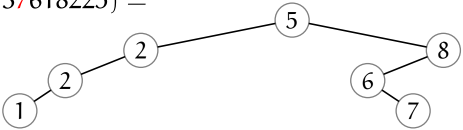
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37{\color{red}6}18225) =$



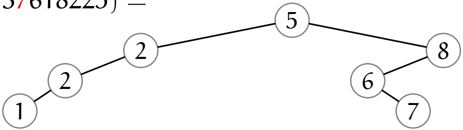
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



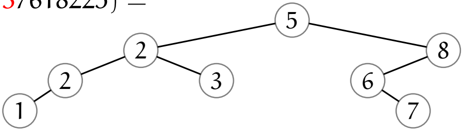
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



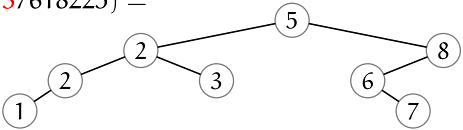
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



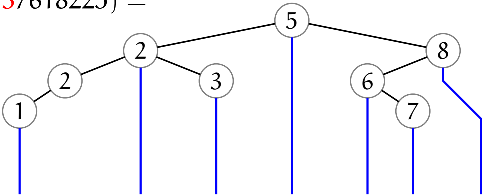
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



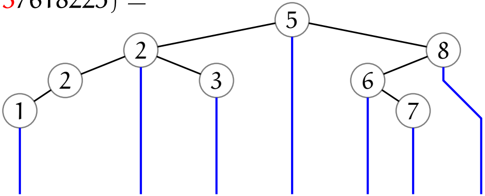
Idea of proof of identity for sylv

- Want to show that $\mathrm{P}_{\mathrm{sylv}}(stst) = \mathrm{P}_{\mathrm{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathrm{P}_{\mathrm{sylv}}(pr) = \mathrm{P}_{\mathrm{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathrm{P}_{\mathrm{sylv}}(22853671 \quad 37618225) =$



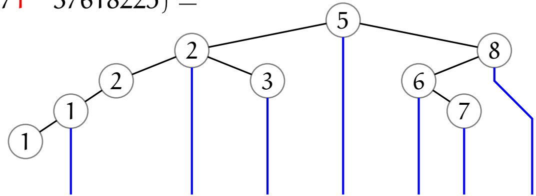
$\mathrm{P}_{\mathrm{sylv}}(67125832 \quad 37618225) =$



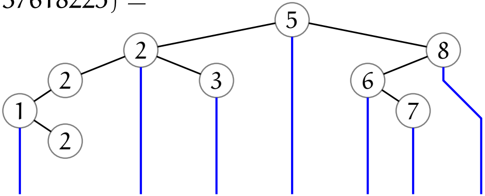
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



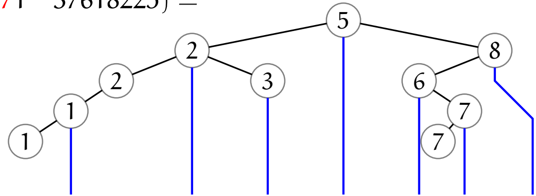
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



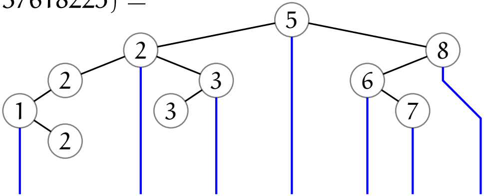
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



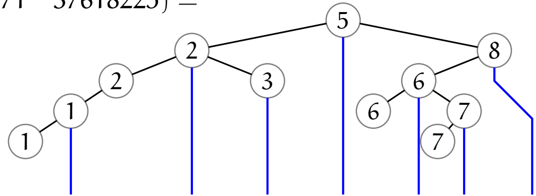
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



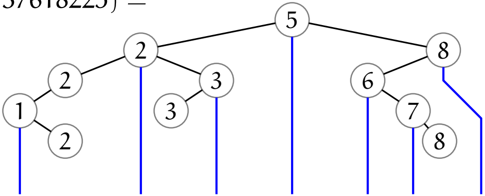
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



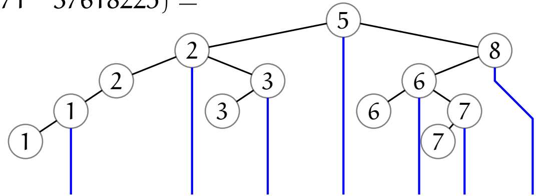
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



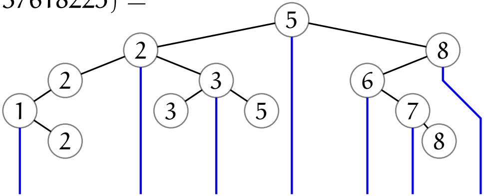
Idea of proof of identity for sylv

- Want to show that $\mathrm{P}_{\mathrm{sylv}}(stst) = \mathrm{P}_{\mathrm{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathrm{P}_{\mathrm{sylv}}(pr) = \mathrm{P}_{\mathrm{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathrm{P}_{\mathrm{sylv}}(22853671 \quad 37618225) =$



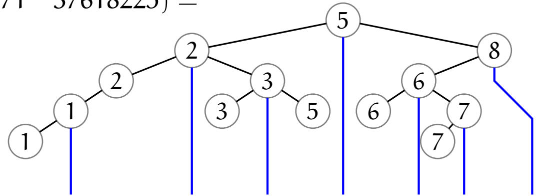
$\mathrm{P}_{\mathrm{sylv}}(67125832 \quad 37618225) =$



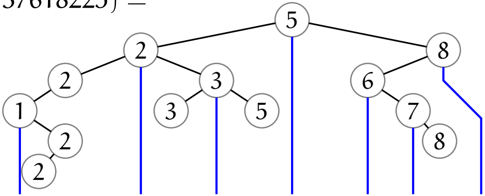
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



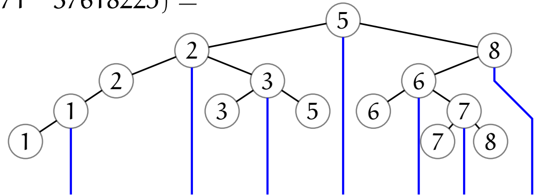
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



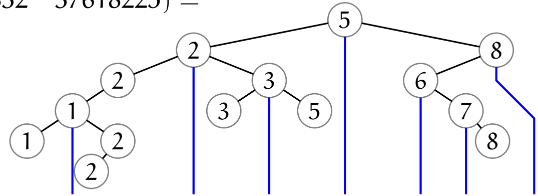
Idea of proof of identity for sylv

- Want to show that $\mathrm{P}_{\mathrm{sylv}}(stst) = \mathrm{P}_{\mathrm{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathrm{P}_{\mathrm{sylv}}(pr) = \mathrm{P}_{\mathrm{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathrm{P}_{\mathrm{sylv}}(22853671 \quad 37618225) =$



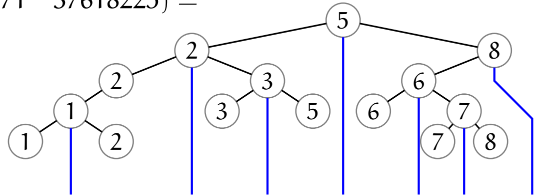
$\mathrm{P}_{\mathrm{sylv}}(67125832 \quad 37618225) =$



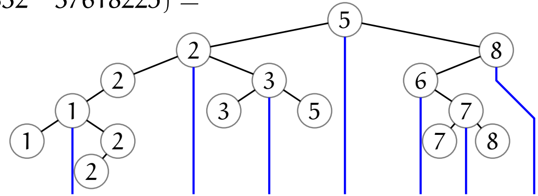
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



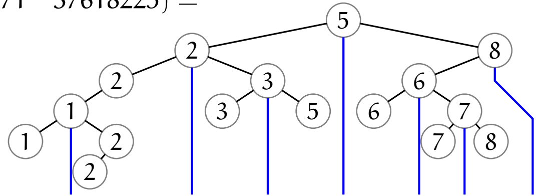
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



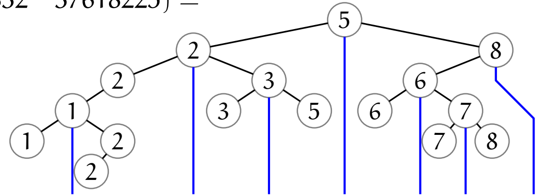
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



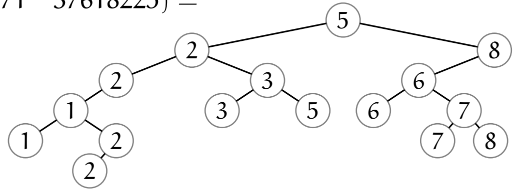
$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



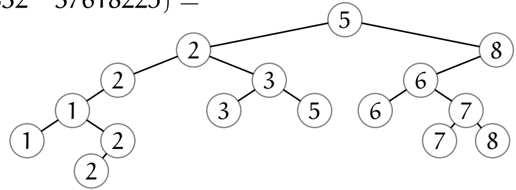
Idea of proof of identity for sylv

- Want to show that $\mathsf{P}_{\mathsf{sylv}}(stst) = \mathsf{P}_{\mathsf{sylv}}(tsst)$ for all $s, t \in \mathcal{A}^*$.
- Suffices to prove that $\mathsf{P}_{\mathsf{sylv}}(pr) = \mathsf{P}_{\mathsf{sylv}}(qr)$ for all $p, q, r \in \mathcal{A}^*$, where p, q, r contain the same number of each symbol.

$\mathsf{P}_{\mathsf{sylv}}(22853671 \quad 37618225) =$



$\mathsf{P}_{\mathsf{sylv}}(67125832 \quad 37618225) =$



Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):

1	2			
	3	4	4	
			5	7

To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, …, finally u_k.
- Call the resulting QRT $\mathrm{P}_{\mathsf{hypo}}(u)$. For example, $\mathrm{P}_{\mathsf{hypo}}(15344723)$is the QRT above.

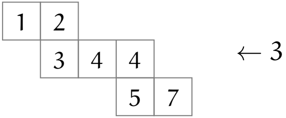
Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $\mathrm{P}_{\mathsf{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):



To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, ..., finally u_k.
- Call the resulting QRT $P_{\text{hypo}}(u)$. For example, $P_{\text{hypo}}(15344723)$is the QRT above.

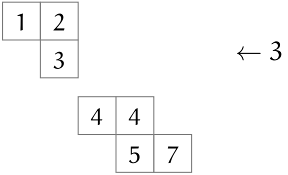
Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $P_{\text{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):



To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, ..., finally u_k.
- Call the resulting QRT $P_{\text{hypo}}(u)$. For example, $P_{\text{hypo}}(15344723)$is the QRT above.

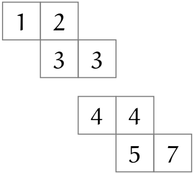
Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $P_{\text{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):



To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, . . . , finally u_k.
- Call the resulting QRT $P_{\text{hypo}}(u)$. For example, $P_{\text{hypo}}(15344723)$is the QRT above.

Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $P_{\text{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):

1	2			
	3	3		
		4	4	
			5	7

To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, …, finally u_k.
- Call the resulting QRT $P_{\text{hypo}}(u)$. For example, $P_{\text{hypo}}(15344723)$is the QRT above.

Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $P_{\text{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):

1	2			
	3	3		
		4	4	
			5	7

To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, ..., finally u_k.
- Call the resulting QRT $\mathrm{P}_{\mathrm{hypo}}(u)$. For example, $\mathrm{P}_{\mathrm{hypo}}(15344723)$is the QRT above.

Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $\mathrm{P}_{\mathrm{hypo}}(u)$ $\Longleftrightarrow$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):

1	2			
	3	3		
		4	4	
			5	7

To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, ..., finally u_k.
- Call the resulting QRT $\mathrm{P}_{\text{hypo}}(u)$. For example, $\mathrm{P}_{\text{hypo}}(15344723)$is the QRT above.

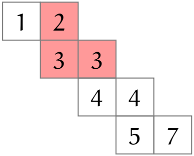
Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $\mathrm{P}_{\text{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):



To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, …, finally u_k.
- Call the resulting QRT $\mathrm{P}_{\mathrm{hypo}}(u)$. For example, $\mathrm{P}_{\mathrm{hypo}}(15344723)$ is the QRT above.

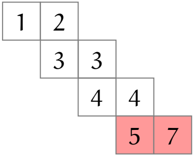
Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $\mathrm{P}_{\mathrm{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

Quasi-ribbon tableaux & insertion

Quasi-ribbon tableau (QRT):



To insert a symbol a into a quasi-ribbon tableau T:

- Break the tableau two parts: $T_{\leqslant}$ is up to and including the bottom-right-most symbol r such that $r \leqslant a$; the remainder is $T_{>}$.
- Add a to the right of r.
- Attach $T_{>}$ to the bottom of a.

For a word $u = u_1 u_2 \cdots u_k \in \mathcal{A}^*$.

- Start with an empty QRT and insert u_1, then u_2, ..., finally u_k.
- Call the resulting QRT $P_{\text{hypo}}(u)$. For example, $P_{\text{hypo}}(1534 4723)$ is the QRT above.

Lemma

If $i < j$ are symbols in u and there is no k in u with $i < k < j$, then:

i and j are on the different rows of $P_{\text{hypo}}(u)$ $\iff$ In u, some symbol i is to the right of some symbol j

The hypoplactic monoid

Define $u \equiv_{\text{hypo}} v \iff \mathsf{P}_{\text{hypo}}(u) = \mathsf{P}_{\text{hypo}}(v)$.

Theorem (Novelli 2000)

The relation $\equiv_{\text{hypo}}$ is a congruence on A^*.

- hypo $= \mathcal{A}^*/\equiv_{\text{hypo}}$ is the hypoplactic monoid.
- $\text{hypo}_n = \mathcal{A}_n^*/\equiv_{\text{hypo}}$ is the hypoplactic monoid of rank n.

Theorem (Cain, M.)

hypo satisfies the identities

$$xyxy = xyyx = yxxy = yxyx;$$
$$xxyx = xyxx.$$

These are the unique shortest non-trivial identities satisfied by hypo.

- A QRT is determined by the number of each symbol it contains and which symbols are on the same rows.
- These are the length-4 identities where the two sides preserve these properties.

Summary table

Monoid	*Symbol*	*Identity*	*In rank n*
Plactic	plac	None	?
Hypoplactic	hypo	$xyxy = yxyx$	Y
Sylvester	sylv	$xyxy = yxxy$	Y
Baxter	baxt	$yxxyxy = yxyxxy$	Y
Stalactic	stal	$xyx = yxx$	Y
Taiga	taig	$xyx = yxx$	Y
Left patience sorting	lPS	None	N
Right patience sorting	rPS	None	Y

Question

Does plac_n satisfy a non-trivial identity for $n \geqslant 4$?

- Conjectured hierarchy of identities for plac_n, length $2 \times 5^{n-1}$.
- Lots of random examples checked in plac_4 using Sage.