Wreath Products of Distributive Forest Algebras

Michael Hahn[1], Andreas Krebs[2], Howard Straubing[3]

[1]Stanford University, [2]University of Tübingen, [3]Boston College

LICS 2018

Introduction

- A much-studied problem in the theory of automata is:
  Can a given regular language L be defined by a formula of some logic?

Introduction

- A much-studied problem in the theory of automata is:
  Can a given regular language L be defined by a formula of some logic?
- in other words, to
  Give an effective characterization of the precise expressive power of the logic.

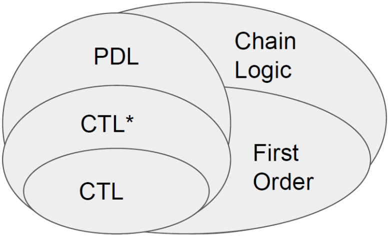
Introduction

- A much-studied problem in the theory of automata is: *Can a given regular language L be defined by a formula of some logic?*
- in other words, to *Give an effective characterization of the precise expressive power of the logic.*
- For automata over **words**, we have effective tests for definability in many temporal and predicate logics

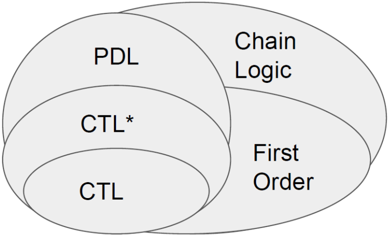
Introduction

- A much-studied problem in the theory of automata is:
  Can a given regular language L be defined by a formula of some logic?
- in other words, to
  Give an effective characterization of the precise expressive power of the logic.
- For automata over **words**, we have effective tests for definability in many temporal and predicate logics
- For automata over **trees**, situation is quite different:
  Effectively deciding expressibility in CTL, CTL* , $FO[\prec]$, PDL, ... remains open.

Logics on Finite Trees



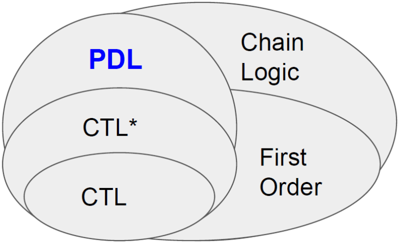
Logics on Finite Trees



For all of these logics,

- Decidability remains an open problem
- Bojańczyk, *et. al.*, (2012) proved characterizations in terms of iterated wreath products of certain forest algebras.

Logics on Finite Trees



For all of these logics,

- Decidability remains an open problem
- Bojańczyk, *et. al.*, (2012) proved characterizations in terms of iterated wreath products of certain forest algebras.

Propositional Dynamic Logic

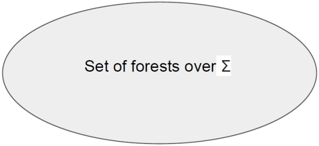
1. For any regular word language L, the language of

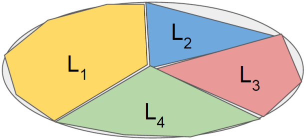
   'Forests where at least one path is in L'

   is in PDL.

Propositional Dynamic Logic

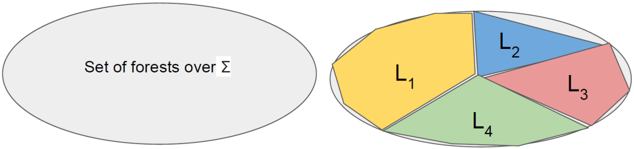
1. Let $L_1, \ldots, L_n$ PDL languages that partition the set of forests.





Propositional Dynamic Logic

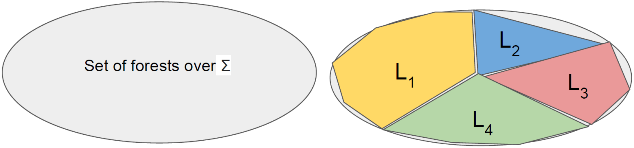
1. Let $L_1, \ldots, L_n$ PDL languages that partition the set of forests.



Let L a regular word language over $\Sigma \times \{a_1, \ldots, a_n\}$.

Propositional Dynamic Logic

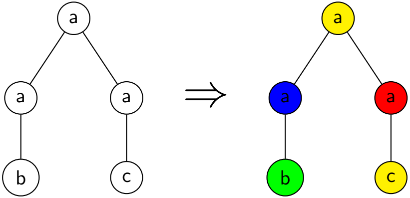
1. Let $L_1, \ldots, L_n$ PDL languages that partition the set of forests.



Let L a regular word language over $\Sigma \times \{a_1, \ldots, a_n\}$.
Then the following language $\mathcal{L}$ is in PDL:

Given a forest f, add to each node a label L_i if the forest below it is in L_i.

Set $f \in \mathcal{L}$ iff the resulting forest has a path in L.



(follows Bojańczyk, *et. al.*, (2012))

Propositional Dynamic Logic

Open Question

Given a regular forest language, can we decide whether it is definable in PDL?

Propositional Dynamic Logic

Open Question

Given a regular forest language, can we decide whether it is definable in PDL?

Following Bojańczyk, *et. al.*, (2012) and Straubing (2013), we attack this problem using algebraic characterizations in terms of **wreath products** of **forest algebras**.

Background: Forest Algebras

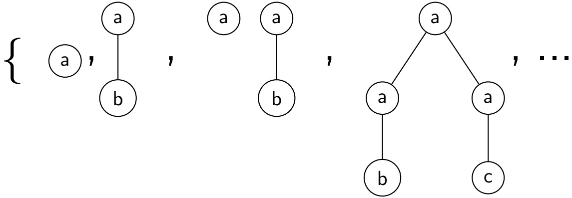
Definition (Bojańczyk and Walukiewicz, 2008)

A tuple (H, V) is called a *forest algebra* if:

1. H and V are monoids
2. There is a faithful action $V \times H \to H$
3. There is a map $I_{\cdot} : H \to V$ such that $I_h h' = h +_H h'$.
4. For each $h \in H$, there is $v \in V$ such that $h = v \cdot_V 0_H$.

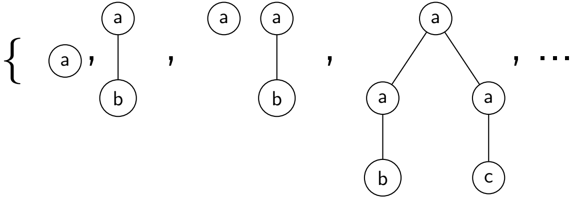
Example (Free Forest Algebra)

- H = set of finite forests over some finite alphabet Σ

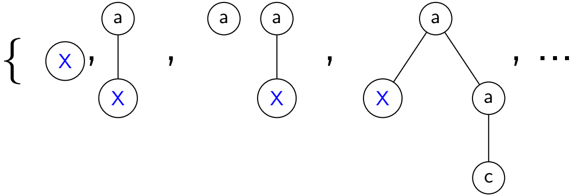


Example (Free Forest Algebra)

- H = set of finite forests over some finite alphabet Σ

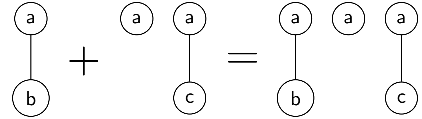


- V = set of such forests where one leaf is a variable

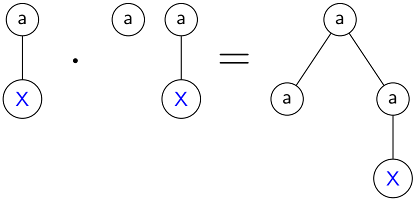


Example (Free Forest Algebra)

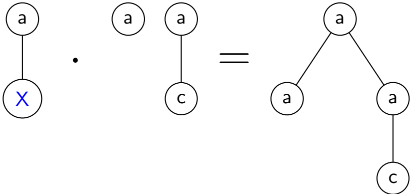
H is a monoid:



V is a monoid:



Action:



Forest Algebras and Languages

- A *forest language* is a set of forests – that is, a subset of H_Σ.
- A forest language $L \subset H_\Sigma$ is *recognized* by (H, V) iff there is a morphism

$$\phi : (H_\Sigma, V_\Sigma) \to (H, V)$$

  such that

$$L = \phi^{-1}(\phi(L))$$

Proposition

Regular forest languages are exactly those recognized by finite forest algebras.

Wreath Product

Definition (Bojańczyk, *et. al.*, (2012))

Let (H_1, V_1), (H_2, V_2) be forest algebras.

$$(H_1, V_1) \wr (H_2, V_2) := (H_1 \times H_2, V_1^{H_2} \times V_2)$$

Wreath Product

Definition (Bojańczyk, *et. al.*, (2012))

Let (H_1, V_1), (H_2, V_2) be forest algebras.

$$(H_1, V_1) \wr (H_2, V_2) := (H_1 \times H_2, V_1^{H_2} \times V_2)$$

Action:

$$(f, v)(h_1, h_2) := (f(h_2)h_1, vh_2)$$

Wreath Product

Definition (Bojańczyk, *et. al.*, (2012))

Let (H_1, V_1), (H_2, V_2) be forest algebras.

$$(H_1, V_1) \wr (H_2, V_2) := (H_1 \times H_2, V_1^{H_2} \times V_2)$$

Action:

$$(f, v)(h_1, h_2) := (f(h_2)h_1, vh_2)$$

Multiplication:

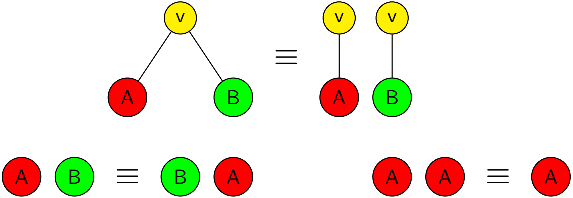
$$(f, v)(f', v') := (f'', vv')$$

with $f''(h) := (f(v'h)) \cdot (f'(h))$.

Distributive Forest Algebras

Definition (Bojańczyk, *et. al.*, (2012))

A forest algebra (H, V) is called *distributive* if any morphism into it respects the identities



- Distributive forest algebras can only distinguish forests with different *sets of paths*

Characterisation of PDL

Theorem (Bojańczyk, *et. al.*, (2012))

For a regular forest language $\mathcal{L} \subset H_\Sigma$*, the following are equivalent:*

1. $\mathcal{L}$ *is definable in PDL*
2. *There are finite distributive forest algebras* $(H_1, V_1), ..., (H_n, V_n)$ *such that* $(H_1, V_1) \wr ... \wr (H_n, V_n)$ *recognizes* $\mathcal{L}$

Characterisation of PDL

Theorem (Bojańczyk, *et. al.*, (2012))

For a regular forest language $\mathcal{L} \subset H_\Sigma$, the following are equivalent:

1. *$\mathcal{L}$ is definable in PDL*
2. *There are finite distributive forest algebras $(H_1, V_1), ..., (H_n, V_n)$ such that $(H_1, V_1) \wr ... \wr (H_n, V_n)$ recognizes $\mathcal{L}$*

- *'There is a path in L'* is recognized by a distributive algebra
- Wreath product simulates composition operation

Characterisation of PDL

Theorem (Bojańczyk, *et. al.*, (2012))

For a regular forest language $\mathcal{L} \subset H_\Sigma$, the following are equivalent:

1. *$\mathcal{L}$ is definable in PDL*
2. *There are finite distributive forest algebras $(H_1, V_1), ..., (H_n, V_n)$ such that $(H_1, V_1) \wr ... \wr (H_n, V_n)$ recognizes $\mathcal{L}$*

- *'There is a path in L'* is recognized by a distributive algebra
- Wreath product simulates composition operation

Similar characterizations for other classes:

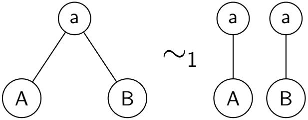
- *CTL*: only one specific distributive algebra is allowed
- *CTL**: only aperiodic distributive algebras are allowed
- ...

Iterated Distributive Laws

Definition (cf. Straubing (2013))

For each $k \geq 1$, define a congruence $\sim_k$ on $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$:

1. Base case: $\sim_1$ is the congruence generated by:

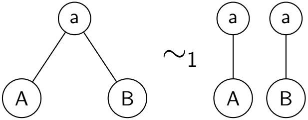


Iterated Distributive Laws

Definition (cf. Straubing (2013))

For each $k \geq 1$, define a congruence $\sim_k$ on $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$:

1. Base case: $\sim_1$ is the congruence generated by:



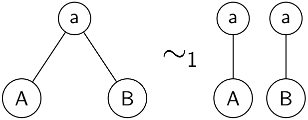
Distributive forest algebras respect $\sim_1$.

Iterated Distributive Laws

Definition (cf. Straubing (2013))

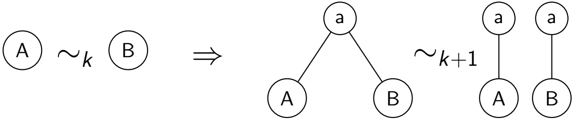
For each $k \geq 1$, define a congruence $\sim_k$ on $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$:

1. Base case: $\sim_1$ is the congruence generated by:



   Distributive forest algebras respect $\sim_1$.

2. Inductive case: $\sim_{k+1}$ is the congruence generated by:

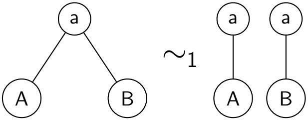


Iterated Distributive Laws

Definition (cf. Straubing (2013))

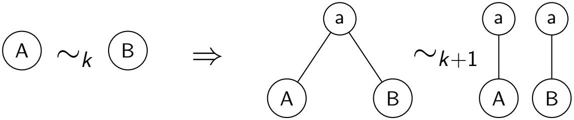
For each $k \geq 1$, define a congruence $\sim_k$ on $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$:

1. Base case: $\sim_1$ is the congruence generated by:



   Distributive forest algebras respect $\sim_1$.

2. Inductive case: $\sim_{k+1}$ is the congruence generated by:



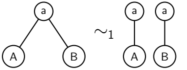
   $k+1$-fold wreath products of distr. algebras respect $\sim_{k+1}$.

Iterated Distributive Laws

Definition (cf. Straubing (2013))

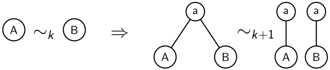
For each $k \geq 1$, define a congruence $\sim_k$ on $\Sigma^\Delta = (H_\Sigma, V_\Sigma)$:

1. Base case: $\sim_1$ is the congruence generated by:



   Distributive forest algebras respect $\sim_1$.

2. Inductive case: $\sim_{k+1}$ is the congruence generated by:



   $k+1$-fold wreath products of distr. algebras respect $\sim_{k+1}$.

A converse to this statement would yield a characterization of PDL!

Iterated Distributive Laws

Definition

(H, V) is *k-distributive* if any morphism into it respects $\sim_k$.

Iterated Distributive Laws

Definition

(H, V) is *k-distributive* if any morphism into it respects $\sim_k$.

Proposition

- Given (H, V), it is decidable whether there is k such that (H, V) is k-distributive.
- Any PDL language is recognized by a k-distributive forest algebra, for some k.

Iterated Distributive Laws

Definition

(H, V) is *k-distributive* if any morphism into it respects $\sim_k$.

Proposition

- Given (H, V), it is decidable whether there is k such that (H, V) is k-distributive.
- Any PDL language is recognized by a k-distributive forest algebra, for some k.

Open Question

Conversely, are all languages recognized by finite k-distributive algebras in PDL?

Iterated Distributive Laws

Definition

(H, V) is *k-distributive* if any morphism into it respects $\sim_k$.

Proposition

- Given (H, V), it is decidable whether there is k such that (H, V) is k-distributive.
- Any PDL language is recognized by a k-distributive forest algebra, for some k.

Open Question

Conversely, are all languages recognized by finite k-distributive algebras in PDL?

An affirmative answer would settle decidability of PDL!

Our Main Result

We solve this for $k = 2$:

Theorem

1. *Let (H, V) be finite and 2-distributive. Then every language recognized by (H, V) is definable in PDL.*
   Further, a product of 4 distributive algebras is enough.
2. *It is decidable whether a regular language is recognized by some 2-distributive forest algebra.*

Proof of the Main Result

Theorem

Let (H, V) be finite and 2-distributive. Then every language recognized by (H, V) is definable in PDL.
Further, a product of 4 distributive algebras is enough.

Proof of the Main Result

Theorem

Let (H, V) be finite and 2-distributive. Then every language recognized by (H, V) is definable in PDL.
Further, a product of 4 distributive algebras is enough.

- Given a 2-distributive forest algebra (H, V),
- we seek distributive algebras $(H_1, V_1), \ldots, (H_4, V_4)$ such that

$$(H, V) \prec (H_1, V_1) \wr (H_2, V_2) \wr (H_3, V_3) \wr (H_4, V_4)$$

Proof of the Main Result

Theorem

Let (H, V) be finite and 2-distributive. Then every language recognized by (H, V) is definable in PDL.
Further, a product of 4 distributive algebras is enough.

- Given a 2-distributive forest algebra (H, V),
- we seek distributive algebras $(H_1, V_1), \ldots, (H_4, V_4)$ such that

$$(H, V) \prec (H_1, V_1) \wr (H_2, V_2) \wr (H_3, V_3) \wr (H_4, V_4)$$

We solve two sub-problems related to the left and right factors:

1. To obtain a right factor, we study the problem of separating forest languages by looking at paths only
2. We then compute a 'minimal' left factor and show that it is distributive.

Proof of the Main Result

Theorem

Let (H, V) be finite and 2-distributive. Then every language recognized by (H, V) is definable in PDL.
Further, a product of 4 distributive algebras is enough.

- Given a 2-distributive forest algebra (H, V),
- we seek distributive algebras $(H_1, V_1), \ldots, (H_4, V_4)$ such that

$$(H, V) \prec (H_1, V_1) \wr (H_2, V_2) \wr (H_3, V_3) \wr (H_4, V_4)$$

We solve two sub-problems related to the left and right factors:

1. To obtain a right factor, we study the problem of separating forest languages by looking at paths only
2. We then compute a 'minimal' left factor and show that it is distributive.

- Approach is parallel to much work on logic over words

Separation Lemma

- For a forest f, let $\pi(f)$ be the set of paths in the forest.

Lemma (Separation Lemma)

Let $\mathcal{L}_1, \mathcal{L}_2 \subseteq H_\Sigma$ be regular forest languages such that

$$\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2) = \emptyset$$

Separation Lemma

- For a forest f, let $\pi(f)$ be the set of paths in the forest.

Lemma (Separation Lemma)

Let $\mathcal{L}_1, \mathcal{L}_2 \subseteq H_\Sigma$ be regular forest languages such that

$$\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2) = \emptyset$$

Then there is a PDL language X such that

$$\mathcal{L}_1 \subseteq X \subseteq (H_\Sigma - \mathcal{L}_2)$$

Further, X can be written as the wreath product of *three* distributive algebras.

Separation Lemma

- For a forest f, let $\pi(f)$ be the set of paths in the forest.

Lemma (Separation Lemma)

Let $\mathcal{L}_1, \mathcal{L}_2 \subseteq H_\Sigma$ be regular forest languages such that

$$\pi(\mathcal{L}_1) \cap \pi(\mathcal{L}_2) = \emptyset$$

Then there is a PDL language X such that

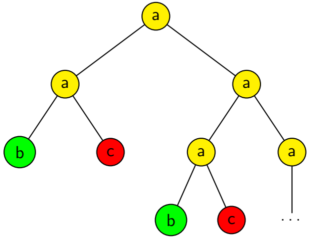
$$\mathcal{L}_1 \subseteq X \subseteq (H_\Sigma - \mathcal{L}_2)$$

Further, X can be written as the wreath product of *three* distributive algebras.

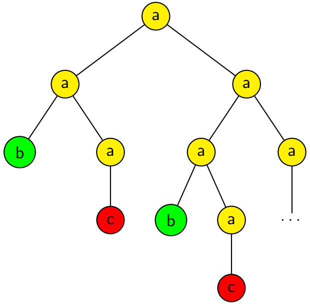
- Similar results are often used to prove decidability for logics over **words**

Separation Lemma: Example

$\mathcal{L}_1$: ‘Each path ends in siblings labeled b and c’

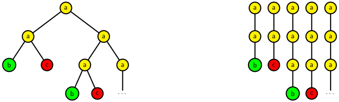


$\mathcal{L}_2$: ‘Each path ends in siblings labeled b and a[c]’

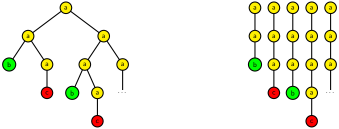


Separation Lemma: Example

$\mathcal{L}_1$: ‘Each path ends in siblings labeled b and c’



$\mathcal{L}_2$: ‘Each path ends in siblings labeled b and a[c]’



- Elements of $\mathcal{L}_1, \mathcal{L}_2$ can be told apart just from looking at the sets of paths.
- No *finite* distributive forest algebra can separate them.
- Nonetheless, PDL separates them.

Proof of the Main Result

1. Use separation lemma to construct right factor
2. Remaining problem is then to find a distributive left-hand factor (H_1, V_1) such that

$$(H, V) \prec (H_1, V_1) \wr (H_2, V_2) \wr (H_3, V_3) \wr (H_4, V_4) \qquad (1)$$

Proof of the Main Result

1. Use separation lemma to construct right factor
2. Remaining problem is then to find a distributive left-hand factor (H_1, V_1) such that

$$(H, V) \prec (H_1, V_1) \wr (H_2, V_2) \wr (H_3, V_3) \wr (H_4, V_4) \qquad (1)$$

3. Construct the 'minimal' left-hand factor (H_1, V_1)
   - In the case of groups, the solution is the kernel group
   - For monoids, the analog is a category (Tilson, 1987)
   - For forest algebras, result is *forest category* (Straubing, 2018)

Proof of the Main Result

1. Use separation lemma to construct right factor
2. Remaining problem is then to find a distributive left-hand factor (H_1, V_1) such that

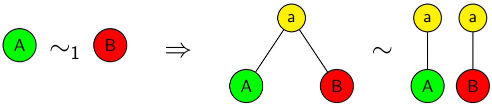
$$(H, V) \prec (H_1, V_1) \wr (H_2, V_2) \wr (H_3, V_3) \wr (H_4, V_4) \quad (1)$$

3. Construct the 'minimal' left-hand factor (H_1, V_1)
   - In the case of groups, the solution is the kernel group
   - For monoids, the analog is a category (Tilson, 1987)
   - For forest algebras, result is *forest category* (Straubing, 2018)
4. Can show that this factor can be replaced by a distributive forest algebra

Decidability of 2-Distributivity

Theorem

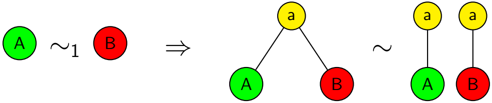
It is decidable whether a regular language is recognized by some 2-distributive forest algebra.



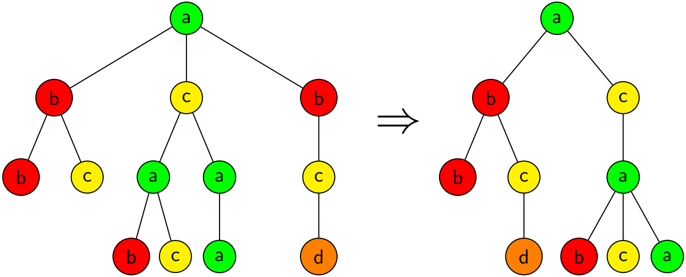
Decidability of 2-Distributivity

Theorem

It is decidable whether a regular language is recognized by some 2-distributive forest algebra.



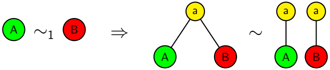
1. Transform regular forest languages into a normal form by mapping forests to representatives that only depends on the set of paths



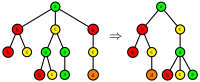
Decidability of 2-Distributivity

Theorem

It is decidable whether a regular language is recognized by some 2-distributive forest algebra.



1. Transform regular forest languages into a normal form



2. This transformation preserves regularity of languages
3. Find pairs A, B by checking intersection of images under this transformation

Discussion

- Decidability remains open for prominent tree logics

Discussion

- Decidability remains open for prominent tree logics
- 2-distributive finite forest algebras describe a decidable subset of PDL
- Generalizing this to $k > 2$ would settle decidability of PDL

Discussion

- Decidability remains open for prominent tree logics
- 2-distributive finite forest algebras describe a decidable subset of PDL
- Generalizing this to $k > 2$ would settle decidability of PDL
- Builds on algebraic characterization that is similar to related logics for which decidability is open
  - Our results may shed light on this larger family of open problems

Discussion

- Decidability remains open for prominent tree logics
- 2-distributive finite forest algebras describe a decidable subset of PDL
- Generalizing this to $k > 2$ would settle decidability of PDL
- Builds on algebraic characterization that is similar to related logics for which decidability is open
  - Our results may shed light on this larger family of open problems
- Our proof method relates to results from the study of word languages
  - Shows how classical theory generalizes to trees and forests