First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Courant Institute of Mathematical Sciences
New York University

Joint work with Joel Spencer

Indian Statistical Institute
January 1, 2016
1. Set-up, first order world, examples
   - First Order World

2. Ehrenfeucht games, Ehrenfeucht values
   - Recursive rule for determining Ehrenfeucht class

3. Probabilities of Ehrenfeucht values as fixed point of an iteration
   - Defining the natural iteration
   - Our main results
   - Outlines for proofs
   - Contraction for $\lambda \geq 1$ - 2-stage process

4. Almost sure theory for first order statements
   - Our main results
   - Rapidly determined
   - Outline of proof for an example
   - Universal trees, again!
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements
Agenda for today’s talk

1. Set-up, first order world, examples
   - First Order World

2. Ehrenfeucht games, Ehrenfeucht values
   - Recursive rule for determining Ehrenfeucht class

3. Probabilities of Ehrenfeucht values as fixed point of an iteration
   - Defining the natural iteration
   - Our main results
   - Outlines for proofs
   - Contraction for $\lambda \geq 1$ - 2-stage process

4. Almost sure theory for first order statements
   - Our main results
   - Rapidly determined
   - Outline of proof for an example
   - Universal trees, again!
The First Order World

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

First Order World

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements
The First Order World

1. Constant Symbol: root;
The First Order World

1. Constant Symbol: root;
2. Equality: $x = y$;
The First Order World

1. Constant Symbol: root;
2. Equality: $x = y$;
3. Parent: $\pi(y) = x$ ($x$ is the parent of $y$, binary predicate);
The First Order World

1. Constant Symbol: root;
2. Equality: $x = y$;
3. Parent: $\pi(y) = x$ (x is the parent of y, binary predicate);
4. Variable Symbols $x, y, z \ldots$, i.e. the nodes;
The First Order World

1. Constant Symbol: root;
2. Equality: $x = y$;
3. Parent: $\pi(y) = x$ (x is the parent of y, binary predicate);
4. Variable Symbols $x, y, z \ldots$, i.e. the nodes;
5. Boolean connectives $\lor, \land, \neg, \rightarrow, \leftrightarrow$, etc;
The First Order World

1. Constant Symbol: root;
2. Equality: \( x = y \);
3. Parent: \( \pi(y) = x \) (\( x \) is the parent of \( y \), binary predicate);
4. Variable Symbols \( x, y, z \ldots \), i.e. the nodes;
5. Boolean connectives \( \lor, \land, \neg, \implies, \iff \), etc;
6. Quantification \( \forall, \exists \), over vertices only.
The First Order World

1. Constant Symbol: root;
2. Equality: \( x = y \);
3. Parent: \( \pi(y) = x \) (\( x \) is the parent of \( y \), binary predicate);
4. Variable Symbols \( x, y, z \ldots \), i.e. the nodes;
5. Boolean connectives \( \lor, \land, \neg, \rightarrow, \leftrightarrow \), etc;
6. Quantification \( \forall, \exists \), over vertices only.

Example

\( A := \exists \) a node with one child and one grandchild.
The First Order World

1. Constant Symbol: root;
2. Equality: \( x = y \);
3. Parent: \( \pi(y) = x \) (\( x \) is the parent of \( y \), binary predicate);
4. Variable Symbols \( x, y, z \ldots \), i.e. the nodes;
5. Boolean connectives \( \lor, \land, \neg, = \Rightarrow, \Leftrightarrow \), etc;
6. Quantification \( \forall, \exists \), over vertices only.

Example

\( A := \exists \) a node with one child and one grandchild.
\( A = \{ \exists x \exists y \exists z [\pi(y) = x \land \pi(z) = y \land [\forall w [\pi(w) = y \Rightarrow z = w]] \land [\forall v [\pi(v) = x \Rightarrow y = v]] \} \).
Let's analyze this first order statement

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

First Order World

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements

Let's analyze this first order statement
Let’s analyze this first order statement

\[ A := \exists u \text{ with one child and one grandchild.} \]
Let’s analyze this first order statement

- $A := \exists u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \bullet, \bullet\}$. 

\begin{align*}
A &:= \exists u \text{ with one child and one grandchild.} \\
\text{Finite State Space } \Sigma &= \{\bullet, \bullet, \bullet\}. \\
\end{align*}
Let’s analyze this first order statement

- $A := \exists u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \bullet, \bullet\}$.
- $\bullet$: $A$ holds; $\bullet$: root has one child, $\neg A$ holds; $\bullet$: all else.
Let’s analyze this first order statement

- $A := \exists u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \bullet, \bullet\}$.
- •: $A$ holds; •: root has one child, $\neg A$ holds; ●: all else.
- Count: 0, 1, $\omega$ ($\omega$ means $\geq 2$).
Let’s analyze this first order statement

- \( A := \exists u \) with one child and one grandchild.
- **Finite** State Space \( \Sigma = \{ \bullet, \bullet, \bullet \} \).
- \( \bullet \): \( A \) holds; \( \bullet \): root has one child, \( \neg A \) holds; \( \bullet \): all else.
- Count: 0, 1, \( \omega \) (\( \omega \) means \( \geq 2 \)).
- Node colour determined by count of children of each colour.
Let’s analyze this first order statement

- $A := \exists u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \cdot, \cdot\}$.
- $\bullet$: $A$ holds; $\cdot$: root has one child, $\neg A$ holds; $\cdot$: all else.
- Count: $0, 1, \omega$ ($\omega$ means $\geq 2$).
- Node colour determined by count of children of each colour.
- $(1, -, -) \Rightarrow \bullet$
Let’s analyze this first order statement

- $A := \exists \ u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \cdot, \circ\}$.
- $\bullet$: $A$ holds; $\cdot$: root has one child, $\neg A$ holds; $\circ$: all else.
- Count: 0, 1, $\omega$ ($\omega$ means $\geq 2$).
- Node colour determined by count of children of each colour.

\[(1, -, -) \Rightarrow \bullet\]
\[(\omega, -, -) \Rightarrow \bullet\]
Let’s analyze this first order statement

- \( A := \exists u \) with one child and one grandchild.
- **Finite** State Space \( \Sigma = \{\bullet, \bullet, \bullet\} \).
- \( \bullet: A \) holds; \( \bullet: \) root has one child, \( \neg A \) holds; \( \bullet: \) all else.
- Count: 0, 1, \( \omega \) (\( \omega \) means \( \geq 2 \)).
- Node colour determined by count of children of each colour.

- \((1, -, -) \Rightarrow \bullet\)
- \((\omega, -, -) \Rightarrow \bullet\)
- \((0, 1, 0) \Rightarrow \bullet\)
Let’s analyze this first order statement

- \( A := \exists u \text{ with one child and one grandchild.} \)
- **Finite** State Space \( \Sigma = \{\bullet, \bullet, \bullet\} \).
- \( \bullet: A \text{ holds}; \bullet: \text{ root has one child, } \neg A \text{ holds}; \bullet: \text{ all else.} \)
- Count: 0, 1, \( \omega \) (\( \omega \) means \( \geq 2 \)).
- Node colour determined by count of children of each colour.

\[
\begin{align*}
(1, -, -) & \Rightarrow \bullet \\
(\omega, -, -) & \Rightarrow \bullet \\
(0, 1, 0) & \Rightarrow \bullet \\
(0, 0, 1) & \Rightarrow \bullet
\end{align*}
\]
Let’s analyze this first order statement

- \( A := \exists u \) with one child and one grandchild.
- **Finite** State Space \( \Sigma = \{\bullet, \circ, \cdot\} \).
- \( \bullet \): \( A \) holds; \( \circ \): root has one child, \( \neg A \) holds; \( \cdot \): all else.
- Count: 0, 1, \( \omega \) (\( \omega \) means \( \geq 2 \)).
- Node colour determined by count of children of each colour.
  
  \[
  \begin{align*}
  (1, - , -) & \Rightarrow \bullet \\
  (\omega , - , -) & \Rightarrow \bullet \\
  (0, 1, 0) & \Rightarrow \bullet \\
  (0, 0, 1) & \Rightarrow \bullet 
  \end{align*}
  \]

- \( x = \Pr[\bullet], y = \Pr[\circ], z = \Pr[\cdot]. \)
Let’s analyze this first order statement

- $A := \exists u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \bullet, \bullet\}$.
- $\bullet$: $A$ holds; $\bullet$: root has one child, $\neg A$ holds; $\bullet$: all else.
- Count: $0, 1, \omega$ ($\omega$ means $\geq 2$).
- Node colour determined by count of children of each colour.
  - $(1, -, -) \Rightarrow \bullet$
  - $(\omega, -, -) \Rightarrow \bullet$
  - $(0, 1, 0) \Rightarrow \bullet$
  - $(0, 0, 1) \Rightarrow \bullet$

- $x = \Pr[\bullet], y = \Pr[\bullet], z = \Pr[\bullet]$.
- $x = 1 - e^{-x\lambda} + y\lambda e^{-\lambda}, y = z\lambda e^{-\lambda}$. 
Let’s analyze this first order statement

- $A := \exists u$ with one child and one grandchild.
- **Finite** State Space $\Sigma = \{\bullet, \bullet, \bullet\}$.
- $\bullet$: $A$ holds; $\bullet$: root has one child, $\neg A$ holds; $\bullet$: all else.
- Count: 0, 1, $\omega$ ($\omega$ means $\geq 2$).
- Node colour determined by count of children of each colour.
  - $(1, -, -) \Rightarrow \bullet$
  - $(\omega, -, -) \Rightarrow \bullet$
  - $(0, 1, 0) \Rightarrow \bullet$
  - $(0, 0, 1) \Rightarrow \bullet$
- $x = \Pr[\bullet]$, $y = \Pr[\bullet]$, $z = \Pr[\bullet]$.
- $x = 1 - e^{-x\lambda} + ye^{-\lambda}$, $y = z\lambda e^{-\lambda}$.
- Solution $x = f_A(\lambda)$ **unique, nice function of** $\lambda$.  

\[ x = 1 - e^{-x\lambda} + y e^{-\lambda}, y = z\lambda e^{-\lambda}. \]

\[ x = f_A(\lambda) \text{ unique, nice function of } \lambda. \]
$1 - f_A(\lambda)$, as a function of $\lambda$
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

First Order World

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements

Figure: Probability $f_A(\lambda)$ of having no node with one child with one child.
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

First Order World

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements

Figure: Probability $f_A(\lambda)$ of having no node with one child with one child.

$$1 - f_A(\lambda), \text{ as a function of } \lambda$$
Immortality

\( B: \) The tree is infinite.
Immortality

\( B \): The tree is infinite.

\( \bullet \bullet \): Yes;

\[ x = \Pr[\bullet], \quad x = 1 - e^{-x\lambda}. \]

Solution \( x = f_B(\lambda) \) not unique when \( \lambda > 1 \).
Immortality

\[ B: \text{The tree is infinite.} \]

- •: Yes; ●: No.
Immortality

\( B: \) The tree is infinite.

- \( \bullet \): Yes; \( \bullet \): No.

- Count 0, \( \omega \).
Immortality

\( B: \) The tree is infinite.

- •: Yes; ●: No.
- Count 0, \( \omega \).
- \( (\omega, -) \Rightarrow \bullet \).
- \( (0, -) \Rightarrow \bigcirc \).
Immortality

\( B: \) The tree is infinite.

- \( \bullet \): Yes; \( \cdot \): No.
- Count 0, \( \omega \).
- (\( \omega, - \)) \( \Rightarrow \) \( \bullet \).
- (0, -) \( \Rightarrow \) \( \bullet \).
- \( x = \Pr[\bullet], \ x = 1 - e^{-x\lambda}. \)
Immortality

\( B: \) The tree is infinite.

- \( \bullet \): Yes; \( \circ \): No.
- Count 0, \( \omega \).
- \( (\omega, -) \Rightarrow \bullet. \)
- \( (0, -) \Rightarrow \circ. \)
- \( x = \Pr[\bullet], \ x = 1 - e^{-x\lambda}. \)
- Solution \( x = f_B(\lambda) \) not unique when \( \lambda > 1. \)
Immortality

\( B \): The tree is infinite.

- \( \bullet \): Yes; \( \bullet \): No.
- Count 0, \( \omega \).
- \( (\omega, -) \Rightarrow \bullet. \)
- \( (0, -) \Rightarrow \bullet. \)
- \( x = \Pr[\bullet], x = 1 - e^{-x\lambda}. \)
- Solution \( x = f_B(\lambda) \) **not unique** when \( \lambda > 1. \)

**Theorem (P., Spencer)**

*For first order \( A \), \( P[A] = f_A(\lambda) \) is always a nice function of \( \lambda \) (polynomials, exponentials, iterated exponentials etc.)*
Probability of Immortality, as a function of $\lambda$

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

First Order World

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements
Probability of Immortality, as a function of $\lambda$

Figure: Probability $p(\lambda)$ of having an infinite tree, $0 \leq \lambda \leq 3$. 

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order statements
1. Set-up, first order world, examples
   - First Order World

2. Ehrenfeucht games, Ehrenfeucht values
   - Recursive rule for determining Ehrenfeucht class

3. Probabilities of Ehrenfeucht values as fixed point of an iteration
   - Defining the natural iteration
   - Our main results
   - Outlines for proofs
   - Contraction for $\lambda \geq 1$ - 2-stage process

4. Almost sure theory for first order statements
   - Our main results
   - Rapidly determined
   - Outline of proof for an example
   - Universal trees, again!
Ehrenfeucht games

Definition

1. Trees $T_1, T_2$, roots $R_1, R_2$, $\# \text{ moves} = k$. 
Definition

1. Trees $T_1, T_2$, roots $R_1, R_2$, # moves $= k$.
2. Spoiler picks any one tree and a node from it. Duplicator chooses a node from the other tree.
Ehrenfeucht games

Definition

1. Trees \( T_1, T_2 \), roots \( R_1, R_2 \), \# moves = \( k \).
2. Spoiler picks any one tree and a node from it. Duplicator chooses a node from the other tree.
3. \( (x_i, y_i) \in T_1 \times T_2, 1 \leq i \leq k \), pairs of nodes selected.
## Definition

1. **Trees** $T_1, T_2$, roots $R_1, R_2$, # moves = $k$.
2. **Spoiler** picks any one tree and a node from it. **Duplicator** chooses a node from the other tree.
3. $(x_i, y_i) \in T_1 \times T_2$, $1 \leq i \leq k$, pairs of nodes selected.
4. **Duplicator** wins EHR[$T_1, T_2, k$] if
   - $x_i = R_1 \iff y_i = R_2$,
Ehrenfeucht games

Definition

1. Trees $T_1, T_2$, roots $R_1, R_2$, $\# \text{ moves} = k$.  
2. Spoiler picks any one tree and a node from it. Duplicator chooses a node from the other tree.  
3. $(x_i, y_i) \in T_1 \times T_2, 1 \leq i \leq k$, pairs of nodes selected.  
4. Duplicator wins $EHR[T_1, T_2, k]$ if  
   a. $x_i = R_1 \iff y_i = R_2$,  
   b. $\pi(x_j) = x_i \iff \pi(y_j) = y_i$.  

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Recursive rule for determining Ehrenfeucht class

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order
Ehrenfeucht games

Definition

1. Trees $T_1, T_2$, roots $R_1, R_2$, $\# \text{ moves} = k$.
2. Spoiler picks any one tree and a node from it. Duplicator chooses a node from the other tree.
3. $(x_i, y_i) \in T_1 \times T_2, 1 \leq i \leq k$, pairs of nodes selected.
4. Duplicator wins EHR[$T_1, T_2, k$] if
   - $x_i = R_1 \iff y_i = R_2$,
   - $\pi(x_j) = x_i \iff \pi(y_j) = y_i$,
   - $x_i = x_j \iff y_i = y_j$. 
Ehrenfeucht games

**Definition**

1. Trees $T_1, T_2$, roots $R_1, R_2$, \# moves $= k$.
2. **Spoiler** picks any one tree and a node from it. **Duplicator** chooses a node from the other tree.
3. $(x_i, y_i) \in T_1 \times T_2, 1 \leq i \leq k$, pairs of nodes selected.
4. **Duplicator wins** $\text{EHR}[T_1, T_2, k]$ if
   - $x_i = R_1 \iff y_i = R_2$,
   - $\pi(x_j) = x_i \iff \pi(y_j) = y_i$,
   - $x_i = x_j \iff y_i = y_j$.

**Definition**

$T_1 \equiv_k T_2$ if Duplicator wins $\text{EHR}[T_1, T_2, k]$. 
Equivalence classes and Ehrenfeucht value

**Theorem**

Fix $k$. $\Sigma = \Sigma_k$ finite set of equivalence classes under $\equiv_k$.
Equivalence classes and Ehrenfeucht value

Theorem

\[ \text{Fix } k. \Sigma = \Sigma_k \text{ finite set of equivalence classes under } \equiv_k. \]

Definition

\[ \text{If } T \in \sigma, \sigma \in \Sigma, \text{ then } \sigma \text{ Ehrenfeucht value / class of } T. \]
Equivalence classes and Ehrenfeucht value

Theorem

Fix $k$. $\Sigma = \Sigma_k$ finite set of equivalence classes under $\equiv_k$.

Definition

If $T \in \sigma, \sigma \in \Sigma$, then $\sigma$ Ehrenfeucht value / class of $T$.

Theorem

If $T_1 \equiv_k T_2$ then

$$T_1 \models A \iff T_2 \models A$$

for F.O. $A$ of depth $k$. 
First order properties and probabilities for Galton-Watson trees in the Poisson regime
Moumanti Podder
Set-up, first order world, examples
Ehrenfeucht games, Ehrenfeucht values
Recursive rule for determining Ehrenfeucht class
Probabilities of Ehrenfeucht values as fixed point of an iteration
Almost sure theory for first order

Important observation about Ehrenfeucht values

Fix $k$. Set $C = \{0, 1, \ldots, k-1, \omega\}$, $\omega$ meaning $\geq k$.

Let $n_\sigma \in C$ denote the number of children of root $R$ in class $\sigma \in \Sigma$.

$\exists$ a rule such that $\vec{n} = \{n_\sigma : \sigma \in \Sigma\}$ completely determines the Ehrenfeucht value $\tau$ of $R$.

Call this set of rules $EHR_k$.

$EHR_k : C \times \Sigma \rightarrow \Sigma$. 

Important observation about Ehrenfeucht values

Fix $k$. Set $C = \{0, 1, \ldots, k - 1, \omega\}$, $\omega$ meaning $\geq k$. 

Important observation about Ehrenfeucht values

1. Fix $k$. Set $C = \{0, 1, \ldots k - 1, \omega\}$, $\omega$ meaning $\geq k$.
2. Let $n_\sigma \in C$ denote the number of children of root $R$ in class $\sigma \in \Sigma$. 
Important observation about Ehrenfeucht values

1. Fix $k$. Set $C = \{0, 1, \ldots, k - 1, \omega\}$, $\omega$ meaning $\geq k$.

2. Let $n_\sigma \in C$ denote the number of children of root $R$ in class $\sigma \in \Sigma$.

3. There exists a rule such that $\vec{n} = \{n_\sigma : \sigma \in \Sigma\}$ completely determines the Ehrenfeucht value $\tau$ of $R$. 

Recursive rule for determining Ehrenfeucht class

Probabilities of Ehrenfeucht values as fixed point of an iteration

Almost sure theory for first order
Important observation about Ehrenfeucht values

1. Fix $k$. Set $C = \{0, 1, \ldots k - 1, \omega\}$, $\omega$ meaning $\geq k$.
2. Let $n_\sigma \in C$ denote the number of children of root $R$ in class $\sigma \in \Sigma$.
3. $\exists$ a rule such that $\vec{n} = \{n_\sigma : \sigma \in \Sigma\}$ completely determines the Ehrenfeucht value $\tau$ of $R$.
4. Call this set of rules $EHR_k$. 
Important observation about Ehrenfeucht values

1. Fix $k$. Set $C = \{0, 1, \ldots k - 1, \omega\}$, $\omega$ meaning $\geq k$.
2. Let $n_\sigma \in C$ denote the number of children of root $R$ in class $\sigma \in \Sigma$.
3. $\exists$ a rule such that $\vec{n} = \{n_\sigma : \sigma \in \Sigma\}$ completely determines the Ehrenfeucht value $\tau$ of $R$.
4. Call this set of rules $EHR_k$.
5. $EHR_k : C^\Sigma \rightarrow \Sigma$. 
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples
  • First Order World

Ehrenfeucht games, Ehrenfeucht values
  • Recursive rule for determining Ehrenfeucht class

Probabilities of Ehrenfeucht values as fixed point of an iteration
  • Defining the natural iteration
  • Our main results
  • Outlines for proofs
  • Contraction for $\lambda \geq 1$ - 2-stage process

Almost sure theory for first order statements
  • Our main results
  • Rapidly determined
  • Outline of proof for an example
  • Universal trees, again!
Solution as a fixed point: defining the iteration
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$. 

---

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for $\lambda \geq 1$ - 2-stage process
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $\text{Poisson}(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$. 
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $Poisson(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\vec{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$,
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $Poisson(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\vec{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$. 

**Set-up, first order world, examples**

- Ehrenfeucht games, Ehrenfeucht values
- Probabilities of Ehrenfeucht values as fixed point of an iteration
- Defining the natural iteration
- Our main results
- Outlines for proofs
- Contraction for $\lambda \geq 1$ - 2-stage process
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $Poisson(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\bar{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$.

**Definition**

*Start with any $\bar{x} \in D$.***
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $\text{Poisson}(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\vec{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$.

**Definition**

Start with any $\vec{x} \in D$. Define $\Psi_\lambda : D \rightarrow D$:
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $\text{Poisson}(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\vec{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$.

Definition

Start with any $\vec{x} \in D$. Define $\Psi_\lambda : D \to D$:

1. $v$ has Poisson mean $\lambda$ children.
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $\text{Poisson}(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\vec{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$.

**Definition**

Start with any $\vec{x} \in D$. Define $\Psi_\lambda : D \to D$:

1. $v$ has Poisson mean $\lambda$ children.
2. Each child of $v$ has $\sigma \in \Sigma$ i.i.d. with distribution $\vec{x}$. 
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $\text{Poisson}(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\vec{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$.

**Definition**

Start with any $\vec{x} \in D$. Define $\Psi_\lambda : D \to D$:

1. $v$ has Poisson mean $\lambda$ children.
2. Each child of $v$ has $\sigma \in \Sigma$ i.i.d. with distribution $\vec{x}$.
3. State $\tau$ of $v$ determined by $\text{EHR}_k$. 
Solution as a fixed point: defining the iteration

- $D$ set of all probability distributions on $\Sigma$.
- $T_\lambda$ random G-W tree with $Poisson(\lambda)$ offspring, $P_\lambda$ probability under $T_\lambda$.
- $\tilde{x}(\lambda) = \{x_\sigma(\lambda) : \sigma \in \Sigma\}$, where $P_\lambda(\sigma) = x_\sigma(\lambda)$.

**Definition**

Start with any $\tilde{x} \in D$. Define $\psi_\lambda : D \rightarrow D$:

1. $v$ has Poisson mean $\lambda$ children.
2. Each child of $v$ has $\sigma \in \Sigma$ i.i.d. with distribution $\tilde{x}$.
3. State $\tau$ of $v$ determined by $EHR_k$.
4. $\psi_\lambda(\tilde{x})$ is induced distribution $\tilde{y}$ on $v$. 
Natural interpretation of $\Psi^s_\lambda$

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for $\lambda \geq 1$ - 2-stage process
Natural interpretation of $\Psi^s_\lambda$

1. Start with any $\vec{x} \in D$. 

---

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for $\lambda \geq 1 - 2$-stage process
Natural interpretation of $\psi^s_\lambda$

1. Start with any $\vec{x} \in D$.
2. Construct G-W Poisson($\lambda$) tree rooted at $v$,
Natural interpretation of $\Psi^s_\lambda$

1. Start with any $\vec{x} \in D$.
2. Construct G-W $\text{Poisson}(\lambda)$ tree rooted at $v$, up to depth $s$. 
Natural interpretation of $\Psi_s^\lambda$

1. Start with any $\vec{x} \in D$.
2. Construct G-W $Poisson(\lambda)$ tree rooted at $v$, up to depth $s$.
3. Assign Ehrenfeucht value to each node at depth $s$ according to $\vec{x}$. 
Natural interpretation of $\psi^s_\lambda$

1. Start with any $\vec{x} \in D$.
2. Construct G-W $\text{Poisson}(\lambda)$ tree rooted at $v$, up to depth $s$.
3. Assign Ehrenfeucht value to each node at depth $s$ according to $\vec{x}$.
4. Determine Ehrenfeucht values of nodes at depth $s - 1$ by $EHR_k$.
1. Start with any $\vec{x} \in D$.
2. Construct G-W $\text{Poisson}(\lambda)$ tree rooted at $v$, up to depth $s$.
3. Assign Ehrenfeucht value to each node at depth $s$ according to $\vec{x}$.
4. Determine Ehrenfeucht values of nodes at depth $s - 1$ by $EHR_k$, then those at depth $s - 2$ by $EHR_k$. 

Natural interpretation of $\Psi^s_\lambda$
Natural interpretation of $\Psi^s_\lambda$

1. Start with any $\vec{x} \in D$.
2. Construct G-W $\text{Poisson}(\lambda)$ tree rooted at $v$, up to depth $s$.
3. Assign Ehrenfeucht value to each node at depth $s$ according to $\vec{x}$.
4. Determine Ehrenfeucht values of nodes at depth $s - 1$ by $EHR_k$, then those at depth $s - 2$ by $EHR_k$, and so on.
Natural interpretation of $\Psi^s_\lambda$

1. Start with any $\vec{x} \in D$.
2. Construct G-W $\text{Poisson}(\lambda)$ tree rooted at $v$, up to depth $s$.
3. Assign Ehrenfeucht value to each node at depth $s$ according to $\vec{x}$.
4. Determine Ehrenfeucht values of nodes at depth $s - 1$ by $EHR_k$, then those at depth $s - 2$ by $EHR_k$, and so on.
5. The Ehrenfeucht value of the root $v$ follows $\Psi^s_\lambda(\vec{x})$. 
How the Poisson regime helps

- Initial distribution $\vec{x} \in D$. 

Initial distribution $\vec{x} \in D$. 

As $v$ has Poisson $(\lambda)$ many children, $n_{\sigma} \sim \text{Poisson}(\lambda x_{\sigma})$, $\forall \sigma \in \Sigma$, and $\{n_{\sigma} : \sigma \in \Sigma\}$ mutually independent.
How the Poisson regime helps

- Initial distribution $\vec{x} \in D$.
- As $v$ has $\text{Poisson}(\lambda)$ many children,
How the Poisson regime helps

- Initial distribution $\vec{x} \in D$.
- As $\nu$ has $\text{Poisson}(\lambda)$ many children,
  \[ n_\sigma \sim \text{Poisson}(\lambda x_\sigma), \quad \forall \sigma \in \Sigma, \quad \text{and} \]
  \[ \{n_\sigma : \sigma \in \Sigma\} \text{ mutually independent}. \]
How the Poisson regime helps

- Initial distribution \( \vec{x} \in D \).
- As \( \nu \) has \( \text{Poisson}(\lambda) \) many children,

\[
n_\sigma \sim \text{Poisson}(\lambda x_\sigma), \quad \forall \sigma \in \Sigma, \quad \text{and} \quad \{n_\sigma : \sigma \in \Sigma\} \quad \text{mutually independent}.
\]

- For \( 0 \leq u \leq k - 1 \),

\[
P[n_\sigma = u] = e^{-\lambda x_\sigma} \frac{(\lambda x_\sigma)^u}{u!}.
\]
How the Poisson regime helps

- Initial distribution $\vec{x} \in D$.
- As $\nu$ has $\text{Poisson}(\lambda)$ many children,
  $$n_\sigma \sim \text{Poisson}(\lambda x_\sigma), \quad \forall \sigma \in \Sigma,$$  and
  $$\{n_\sigma : \sigma \in \Sigma\} \text{ mutually independent}.$$

- For $0 \leq u \leq k - 1$,
  $$P[n_\sigma = u] = e^{-\lambda x_\sigma} \frac{(\lambda x_\sigma)^u}{u!}.$$

- For $u = \omega$,
  $$P[n_\sigma = u] = 1 - e^{-\lambda x_\sigma} \sum_{j=0}^{k-1} \frac{(\lambda x_\sigma)^j}{j!}.$$
Solution as a fixed point: the results

Theorem (P., Spencer)

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)

Solution as a fixed point: the results

Theorem (P., Spencer)
Solution as a fixed point: the results

Theorem (P., Spencer)

1. $\tilde{x}(\lambda)$ is a fixed point of $\Psi_\lambda$, i.e.

$$\Psi_\lambda(\tilde{x}(\lambda)) = \tilde{x}(\lambda).$$
Solution as a fixed point: the results

**Theorem (P., Spencer)**

1. \( \tilde{x}(\lambda) \) is a fixed point of \( \psi_\lambda \), i.e.
   \[
   \psi_\lambda(\tilde{x}(\lambda)) = \tilde{x}(\lambda).
   \]

2. \( \psi_\lambda \) is a contraction.
Solution as a fixed point: the results

Theorem (P., Spencer)

1. $\bar{x}(\lambda)$ is a fixed point of $\psi_\lambda$, i.e.
   $$\psi_\lambda(\bar{x}(\lambda)) = \bar{x}(\lambda).$$

2. $\psi_\lambda$ is a contraction.

3. As a result, the fixed point is unique.
Solution as a fixed point: the results

Theorem (P., Spencer)

1. $\vec{x}(\lambda)$ is a fixed point of $\Psi_\lambda$, i.e.
   \[ \Psi_\lambda(\vec{x}(\lambda)) = \vec{x}(\lambda). \]

2. $\Psi_\lambda$ is a contraction.

3. As a result, the fixed point is unique.

4. $\vec{x}(\lambda)$ is a real analytic function of $\lambda$. 
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for \( \lambda \geq 1 \) - 2-stage process

Outline of proof - Slide 1

1. **Fixed point:** \( v \) has \( Poisson(\lambda) \) children,

\[ \text{Fixed point: } v \ \text{has } Poisson(\lambda) \text{ children,} \]
Fixed point: \( v \) has \( Poisson(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_{\sigma}(\lambda) \), true under \( Poisson(\lambda) \) regime.
Fixed point: \( v \) has \( \text{Poisson}(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_\sigma(\lambda) \), true under \( \text{Poisson}(\lambda) \) regime. So \( \Psi_\lambda \) must preserve \( \bar{x}(\lambda) \).
Outline of proof - Slide 1

1. **Fixed point:** \( v \) has \( \text{Poisson}(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_\sigma(\lambda) \), true under \( \text{Poisson}(\lambda) \) regime. So \( \Psi_\lambda \) must preserve \( \bar{x}(\lambda) \).

2. **Contraction for** \( \lambda < 1 \):
Outline of proof - Slide 1

1. **Fixed point**: $v$ has $\text{Poisson}(\lambda)$ children, each child has state $\sigma \in \Sigma$ with probability $x_\sigma(\lambda)$, true under $\text{Poisson}(\lambda)$ regime. So $\Psi_\lambda$ must preserve $\vec{x}(\lambda)$.

2. **Contraction for $\lambda < 1$**: Will show, for $\vec{x}, \vec{y} \in D$:

   $$\|\psi_\lambda(\vec{x}) - \psi_\lambda(\vec{y})\|_{TV} \leq \lambda \cdot \|\vec{x} - \vec{y}\|_{TV}.$$
Outline of proof - Slide 1

1. **Fixed point**: \( \nu \) has \( \text{Poisson}(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_\sigma(\lambda) \), true under \( \text{Poisson}(\lambda) \) regime. So \( \Psi_\lambda \) must preserve \( \vec{x}(\lambda) \).

2. **Contraction for** \( \lambda < 1 \): Will show, for \( \vec{x}, \vec{y} \in D \):

\[
||\Psi_\lambda(\vec{x}) - \Psi_\lambda(\vec{y})||_{TV} \leq \lambda \cdot ||\vec{x} - \vec{y}||_{TV}.
\]

**Proof.**

- Let \( \nu \) have \( s \) children \( \nu_1, \ldots, \nu_s \).
Outline of proof - Slide 1

1. **Fixed point**: $v$ has $\text{Poisson}(\lambda)$ children, each child has state $\sigma \in \Sigma$ with probability $x_\sigma(\lambda)$, true under $\text{Poisson}(\lambda)$ regime. So $\Psi_\lambda$ must preserve $\vec{x}(\lambda)$.

2. **Contraction for $\lambda < 1$**: Will show, for $\vec{x}, \vec{y} \in D$:

$$||\Psi_\lambda(\vec{x}) - \Psi_\lambda(\vec{y})||_{TV} \leq \lambda \cdot ||\vec{x} - \vec{y}||_{TV}.$$

### Proof.

- Let $v$ have $s$ children $v_1, \ldots, v_s$.

- In picture 1, $v_i$ gets state $X_i \in \Sigma$, $X_i \sim \vec{x}$;
First order properties and probabilities for Galton-Watson trees in the Poisson regime

1 Fixed point: \( v \) has \( \text{Poisson}(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_{\sigma}(\lambda) \), true under \( \text{Poisson}(\lambda) \) regime. So \( \Psi_{\lambda} \) must preserve \( \vec{x}(\lambda) \).

2 Contraction for \( \lambda < 1 \): Will show, for \( \vec{x}, \vec{y} \in D \):

\[
\| \Psi_{\lambda}(\vec{x}) - \Psi_{\lambda}(\vec{y}) \|_{TV} \leq \lambda \cdot \| \vec{x} - \vec{y} \|_{TV}.
\]

Proof.

- Let \( v \) have \( s \) children \( v_1, \ldots, v_s \).
- In picture 1, \( v_i \) gets state \( X_i \in \Sigma, \ X_i \sim \vec{x} \); in picture 2, \( v_i \) gets state \( Y_i \in \Sigma, \ Y_i \sim \vec{y} \).
Outline of proof - Slide 1

1. **Fixed point:** \( v \) has \( \text{Poisson}(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_\sigma(\lambda) \), true under \( \text{Poisson}(\lambda) \) regime. So \( \Psi_\lambda \) must preserve \( \vec{x}(\lambda) \).

2. **Contraction for \( \lambda < 1 \):** Will show, for \( \vec{x}, \vec{y} \in D \):

\[
\| \Psi_\lambda(\vec{x}) - \Psi_\lambda(\vec{y}) \|_{TV} \leq \lambda \cdot \| \vec{x} - \vec{y} \|_{TV}.
\]

**Proof.**

- Let \( v \) have \( s \) children \( v_1, \ldots, v_s \).
- In picture 1, \( v_i \) gets state \( X_i \in \Sigma, \ X_i \sim \vec{x} \); in picture 2, \( v_i \) gets state \( Y_i \in \Sigma, \ Y_i \sim \vec{y} \).
- \((X_i, Y_i), 1 \leq i \leq s\) mutually independent.
Outline of proof - Slide 1

1. **Fixed point:** $v$ has $Poisson(\lambda)$ children, each child has state $\sigma \in \Sigma$ with probability $x_\sigma(\lambda)$, true under $Poisson(\lambda)$ regime. So $\Psi_\lambda$ must preserve $\bar{x}(\lambda)$.

2. **Contraction for $\lambda < 1$:** Will show, for $\bar{x}, \bar{y} \in D$:

$$||\Psi_\lambda(\bar{x}) - \Psi_\lambda(\bar{y})||_{TV} \leq \lambda \cdot ||\bar{x} - \bar{y}||_{TV}.$$

**Proof.**

- Let $v$ have $s$ children $v_1, \ldots, v_s$.
- In picture 1, $v_i$ gets state $X_i \in \Sigma$, $X_i \sim \bar{x}$; in picture 2, $v_i$ gets state $Y_i \in \Sigma$, $Y_i \sim \bar{y}$.
- $(X_i, Y_i), 1 \leq i \leq s$ mutually independent. But $X_i, Y_i$ coupled so that
Outline of proof - Slide 1

1. **Fixed point:** \( v \) has \( \text{Poisson}(\lambda) \) children, each child has state \( \sigma \in \Sigma \) with probability \( x_\sigma(\lambda) \), true under \( \text{Poisson}(\lambda) \) regime. So \( \Psi_\lambda \) must preserve \( \vec{x}(\lambda) \).

2. **Contraction for \( \lambda < 1 \):** Will show, for \( \vec{x}, \vec{y} \in D \):
   \[
   \|\Psi_\lambda(\vec{x}) - \Psi_\lambda(\vec{y})\|_{TV} \leq \lambda \cdot \|\vec{x} - \vec{y}\|_{TV}.
   \]

**Proof.**

- Let \( v \) have \( s \) children \( v_1, \ldots, v_s \).
- In picture 1, \( v_i \) gets state \( X_i \in \Sigma, \ X_i \sim \vec{x} \); in picture 2, \( v_i \) gets state \( Y_i \in \Sigma, \ Y_i \sim \vec{y} \).
- \((X_i, Y_i), 1 \leq i \leq s\) mutually independent. But \( X_i, Y_i \) coupled so that
  \[
P[X_i \neq Y_i] = \|\vec{x} - \vec{y}\|_{TV}.
  \]
Proof continued...

\[ X_v \sim \Psi_\lambda(\vec{x}), \quad Y_v \sim \Psi_\lambda(\vec{y}). \]

\[ ||\Psi_\lambda(\vec{x}) - \Psi_\lambda(\vec{y})||_{TV} \leq P[X_v \neq Y_v] \leq \infty \sum_{s=0}^\infty P[Poi(\lambda) = s] \cdot ||\vec{x} - \vec{y}||_{TV}. \]

\[ \lambda \cdot ||\vec{x} - \vec{y}||_{TV}. \]
Proof continued...

If $X_v$, $Y_v$ states of node $v$ in pictures 1, 2,
Outline of proof - Slide 2

Proof continued...

- If $X_v, Y_v$ states of node $v$ in pictures 1, 2, then $X_v \sim \Psi_\lambda(\vec{x}), Y_v \sim \Psi_\lambda(\vec{y})$. 

\[ |\Psi_\lambda(\vec{x}) - \Psi_\lambda(\vec{y})|_{TV} \leq P[X_v \neq Y_v] \leq \infty \sum_{s=0}^{\infty} P[\text{Poi}(\lambda) = s] \cdot \sum_{i=1}^{s} P[X_i \neq Y_i] = \infty \sum_{s=0}^{\infty} P[\text{Poi}(\lambda) = s] \cdot |\vec{x} - \vec{y}|_{TV} = \lambda \cdot |\vec{x} - \vec{y}|_{TV}. \]
Proof continued...

- If $X_v, Y_v$ states of node $v$ in pictures 1, 2, then $X_v \sim \psi_\lambda(\vec{x}), Y_v \sim \psi_\lambda(\vec{y})$.

\[ \|\psi_\lambda(\vec{x}) - \psi_\lambda(\vec{y})\|_{TV} \leq P[X_v \neq Y_v] \]

\[ \leq \sum_{s=0}^{\infty} P[\text{Poi}(\lambda) = s] \sum_{i=1}^{s} P[X_i \neq Y_i] \]

\[ = \sum_{s=0}^{\infty} P[\text{Poi}(\lambda) = s] \cdot s \cdot \|\vec{x} - \vec{y}\|_{TV} \]

\[ = \lambda \cdot \|\vec{x} - \vec{y}\|_{TV}. \]
Before we can cover $\lambda \geq 1$: *universal* trees

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for $\lambda \geq 1$ - 2-stage process

Fix $k$. $\text{Rad}[0] = 0$, $\text{Rad}[i+1] = 3\text{Rad}[i] + 1$.  

$\exists$ finite universal tree $\text{UNIV}_k$ such that:

Theorem (P., Spencer) If for some $v \in T$, $T(v) \sim \text{UNIV}_k$, the Ehrenfeucht value of $T$ is determined completely by $T|_{\text{Rad}[k]}$ (T truncated at depth $\text{Rad}[k]$).

Definition Fix $k$. $T$ called $s$-universal if $T|_s$ determines its Ehrenfeucht value.

Remark $v \in T$ with $T(v) \sim \text{UNIV}_k \Rightarrow T_{\text{Rad}[k]}$-universal.
Before we can cover $\lambda \geq 1$: universal trees

- Fix $k$.
Before we can cover $\lambda \geq 1$: universal trees

- Fix $k$. $Rad[0] = 0$, $Rad[i + 1] = 3R[i] + 1$. 

Fix $k$. $Rad[0] = 0$, $Rad[i + 1] = 3R[i] + 1$. 

$\exists$ finite universal tree $UNIV_k$ such that:

*Theorem (P., Spencer)*

If for some $v \in T$, $T(v) \sim UNIV_k$, the Ehrenfeucht value of $T$ is determined completely by $T_{Rad[k]}(T$ truncated at depth $Rad[k])$.

*Definition*

Fix $k$. $T$ called $s$-universal if $T_{s}$ determines its Ehrenfeucht value.

*Remark*

$\exists v \in T$ with $T(v) \sim UNIV_k \Rightarrow T_{Rad[k]}$-universal.
Before we can cover $\lambda \geq 1$: *universal* trees

- Fix $k$. $\text{Rad}[0] = 0$, $\text{Rad}[i + 1] = 3R[i] + 1$.
- $\exists$ finite *universal* tree $\text{UNIV}_k$ such that:

  $\text{Rad}[0] = 0$, $\text{Rad}[i + 1] = 3R[i] + 1$.  

  $\exists$ finite *universal* tree $\text{UNIV}_k$ such that:

  $\exists$ finite *universal* tree $\text{UNIV}_k$ such that:
Before we can cover $\lambda \geq 1$: universal trees

- Fix $k$. $Rad[0] = 0$, $Rad[i + 1] = 3R[i] + 1$.
- $\exists$ finite universal tree $UNIV_k$ such that:

**Theorem (P., Spencer)**

If for some $v \in T$, $T(v) \cong UNIV_k$, the Ehrenfeucht value of $T$ is determined completely by $T|_{Rad[k]}$ ($T$ truncated at depth $Rad[k]$).
Before we can cover $\lambda \geq 1$: universal trees

- Fix $k$. $\text{Rad}[0] = 0$, $\text{Rad}[i + 1] = 3R[i] + 1$.
- $\exists$ finite universal tree $\text{UNIV}_k$ such that:

### Theorem (P., Spencer)

If for some $v \in T$, $T(v) \cong \text{UNIV}_k$, the Ehrenfeucht value of $T$ is determined completely by $T|_{\text{Rad}[k]}$ ($T$ truncated at depth $\text{Rad}[k]$).

### Definition

Fix $k$. $T$ called $s$-universal if $T|_s$ determines its Ehrenfeucht value.
Before we can cover $\lambda \geq 1$: universal trees

- Fix $k$. $\text{Rad}[0] = 0$, $\text{Rad}[i + 1] = 3\text{R}[i] + 1$.
- $\exists$ finite universal tree $\text{UNIV}_k$ such that:

**Theorem (P., Spencer)**

*If for some $v \in T$, $T(v) \cong \text{UNIV}_k$, the Ehrenfeucht value of $T$ is determined completely by $T|_{\text{Rad}[k]}$ ($T$ truncated at depth $\text{Rad}[k]$).*

**Definition**

*Fix $k$. $T$ called $s$-universal if $T|_s$ determines its Ehrenfeucht value.*

**Remark**

$v \in T$ with $T(v) \cong \text{UNIV}_k \implies T\text{ Rad}[k]$-universal.
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.

Recall $UNIV_k$. Let $D_0$ be its depth.
The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot Rad[k]$. 

Proof of contraction for $\lambda \geq 1$
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot Rad[k]$.
3. Will generate $T|_s$ in two stages.
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot \text{Rad}[k]$.
3. Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$. 

Proof of contraction for $\lambda \geq 1$
The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot \text{Rad}[k]$.
3. Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$.
4. Hang from each node at depth $s_0$, independently, a random G-W tree of depth $\leq D_0$. 
Proof of contraction for \( \lambda \geq 1 \)

The Two-Stage Process:

1. Recall \( UNIV_k \). Let \( D_0 \) be its depth.
2. Set \( s = s_0 + D_0, \quad s_0 \geq 2 \cdot \text{Rad}[k] \).
3. Will generate \( T|_s \) in two stages. First fix an arbitrary \( T_0 \) of depth \( \leq s_0 \).
4. Hang from each node at depth \( s_0 \), independently, a random G-W tree of depth \( \leq D_0 \). We call this \( \text{Ext}(T_0) \), of depth \( \leq s \).
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot Rad[k]$.
3. Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$.
4. Hang from each node at depth $s_0$, independently, a random G-W tree of depth $\leq D_0$. We call this $Ext(T_0)$, of depth $\leq s$.
5. Let $Y = \#$ nodes at depth $s$. 

Proof of contraction for $\lambda \geq 1$
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot \text{Rad}[k]$.
3. Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$.
4. Hang from each node at depth $s_0$, independently, a random G-W tree of depth $\leq D_0$. We call this $\text{Ext}(T_0)$, of depth $\leq s$.
5. Let $Y = \#$ nodes at depth $s$. Start with any $\vec{x} \in D$. 
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, where $s_0 \geq 2 \cdot Rad[k]$.
3. Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$.
4. Hang from each node at depth $s_0$, independently, a random G-W tree of depth $\leq D_0$. We call this $Ext(T_0)$, of depth $\leq s$.
5. Let $Y = \# \text{ nodes at depth } s$. Start with any $\vec{x} \in D$.
6. Assign to each node at depth $s$ of $Ext(T_0)$, independently, an Ehrenfeucht value according to $\vec{x}$. 
Proof of contraction for $\lambda \geq 1$

The Two-Stage Process:

1. Recall $UNIV_k$. Let $D_0$ be its depth.
2. Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot Rad[k]$.
3. Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$.
4. Hang from each node at depth $s_0$, independently, a random G-W tree of depth $\leq D_0$. We call this $Ext(T_0)$, of depth $\leq s$.
5. Let $Y = \#$ nodes at depth $s$. Start with any $\vec{x} \in D$.
6. Assign to each node at depth $s$ of $Ext(T_0)$, independently, an Ehrenfeucht value according to $\vec{x}$.
7. The Ehrenfeucht value this assigns to the root $v$ of $T_0$ follows distribution $\Psi^s_\lambda(\vec{x}, T_0)$. 

Recall $UNIV_k$. Let $D_0$ be its depth. 

Set $s = s_0 + D_0$, $s_0 \geq 2 \cdot Rad[k]$.

Will generate $T|_s$ in two stages. First fix an arbitrary $T_0$ of depth $\leq s_0$.

Hang from each node at depth $s_0$, independently, a random G-W tree of depth $\leq D_0$. We call this $Ext(T_0)$, of depth $\leq s$.

Let $Y = \#$ nodes at depth $s$. Start with any $\vec{x} \in D$.

Assign to each node at depth $s$ of $Ext(T_0)$, independently, an Ehrenfeucht value according to $\vec{x}$.

The Ehrenfeucht value this assigns to the root $v$ of $T_0$ follows distribution $\Psi^s_\lambda(\vec{x}, T_0)$.
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for \( \lambda \geq 1 \) - 2-stage process
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for \( \lambda \geq 1 \) - 2-stage process

Pictorial representation

Figure: Structure of Ext(\( T_0 \)).
Now adopt similar argument as when $\lambda < 1$
Now adopt similar argument as when $\lambda < 1$

1. Start with $\vec{x}, \vec{y} \in D$. 

---

First order properties and probabilities for Galton-Watson trees in the Poisson regime
Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration
Defining the natural iteration
Our main results
Outlines for proofs

Contraction for $\lambda \geq 1$ - 2-stage process
Now adopt similar argument as when $\lambda < 1$

1. Start with $\vec{x}, \vec{y} \in D$.
2. Let nodes at depth $s$ of $\text{Ext}(T_0)$ be $u_1, \ldots u_Y$. 
Now adopt similar argument as when $\lambda < 1$

1. Start with $\vec{x}, \vec{y} \in D$.
2. Let nodes at depth $s$ of $\text{Ext}(T_0)$ be $u_1, \ldots, u_Y$.
3. Picture 1: assign Ehrenfeucht values $X_1, \ldots, X_Y$ to $u_1, \ldots, u_Y$ according to $\vec{x}$. 
Now adopt similar argument as when $\lambda < 1$

1. Start with $\vec{x}, \vec{y} \in D$.
2. Let nodes at depth $s$ of $\text{Ext}(T_0)$ be $u_1, \ldots u_Y$.
3. Picture 1: assign Ehrenfeucht values $X_1, \ldots X_Y$ to $u_1, \ldots u_Y$ according to $\vec{x}$.
4. Picture 2: assign Ehrenfeucht values $Z_1, \ldots Z_Y$ to $u_1, \ldots u_Y$ according to $\vec{y}$.
Now adopt similar argument as when $\lambda < 1$

1. Start with $\vec{x}, \vec{y} \in D$.
2. Let nodes at depth $s$ of $Ext(T_0)$ be $u_1, \ldots u_Y$.
3. Picture 1: assign Ehrenfeucht values $X_1, \ldots X_Y$ to $u_1, \ldots u_Y$ according to $\vec{x}$.
4. Picture 2: assign Ehrenfeucht values $Z_1, \ldots Z_Y$ to $u_1, \ldots u_Y$ according to $\vec{y}$.
5. $(X_i, Z_i)$ independent over $1 \leq i \leq Y$. 

But $(X_i, Z_i)$ coupled so that $P[X_i \neq Z_i] = ||\vec{x} - \vec{y}||_{TV}$. 

---

First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for $\lambda \geq 1$ - 2-stage process
Now adopt similar argument as when $\lambda < 1$

1. Start with $\vec{x}, \vec{y} \in D$.
2. Let nodes at depth $s$ of $\text{Ext}(T_0)$ be $u_1, \ldots, u_Y$.
3. Picture 1: assign Ehrenfeucht values $X_1, \ldots, X_Y$ to $u_1, \ldots, u_Y$ according to $\vec{x}$.
4. Picture 2: assign Ehrenfeucht values $Z_1, \ldots, Z_Y$ to $u_1, \ldots, u_Y$ according to $\vec{y}$.
5. $(X_i, Z_i)$ independent over $1 \leq i \leq Y$. But $(X_i, Z_i)$ coupled so that

$$P[X_i \neq Z_i] = \|\vec{x} - \vec{y}\|_{TV}.$$
Here’s the catch, though...
Here’s the catch, though...

1. Recall $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots, T(v_t)$ i.i.d G-W trees up to depth $D_0$. 

Probabilities of Ehrenfeucht values as fixed point of an iteration
Here’s the catch, though...

1. Recall \( v_1, \ldots v_t \) nodes at depth \( s_0 \). \( T(v_1), \ldots T(v_t) \) i.i.d G-W trees up to depth \( D_0 \).
2. If \( T(v_i) \cong \text{UNIV}_k \) for any \( 1 \leq i \leq t \),
Here’s the catch, though...

1. Recall $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots T(v_t)$ i.i.d G-W trees up to depth $D_0$.
2. If $T(v_i) \cong UNIV_k$ for any $1 \leq i \leq t$, Ehrenfeucht value of $Ext(T_0)$ determined by $Ext(T_0)|_{Rad[k]}$, 

1. \begin{itemize}
   \item Recall $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots T(v_t)$ i.i.d G-W trees up to depth $D_0$.
   \item If $T(v_i) \cong UNIV_k$ for any $1 \leq i \leq t$, Ehrenfeucht value of $Ext(T_0)$ determined by $Ext(T_0)|_{Rad[k]}$.
\end{itemize}
Here’s the catch, though...

1. Recall $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots, T(v_t)$ i.i.d G-W trees up to depth $D_0$.

2. If $T(v_i) \cong UNIV_k$ for any $1 \leq i \leq t$, Ehrenfeucht value of $Ext(T_0)$ determined by $Ext(T_0)|_{Rad[k]}$, hence by $T_0$, which is fixed.
Here’s the catch, though...

1. Recall $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots, T(v_t)$ i.i.d G-W trees up to depth $D_0$.

2. If $T(v_i) \cong \text{UNIV}_k$ for any $1 \leq i \leq t$, Ehrenfeucht value of $\text{Ext}(T_0)$ determined by $\text{Ext}(T_0)|_{\text{Rad}[k]}$, hence by $T_0$, which is fixed.

3. Let $GOOD = \bigcup_{i=1}^{t} \{ T(v_i) \cong \text{UNIV}_k \}$.
Here’s the catch, though...

1. Recall $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots, T(v_t)$ i.i.d G-W trees up to depth $D_0$.

2. If $T(v_i) \cong UNIV_k$ for any $1 \leq i \leq t$, Ehrenfeucht value of $Ext(T_0)$ determined by $Ext(T_0)|_{Rad[k]}$, hence by $T_0$, which is fixed.

3. Let $GOOD = \bigcup_{i=1}^t \{ T(v_i) \cong UNIV_k \}$. So, under $GOOD$, Ehrenfeucht value of root $v$ independent of $\vec{x}, \vec{y}$.
Here’s the catch, though...

1. Recall \( v_1, \ldots, v_t \) nodes at depth \( s_0 \). \( T(v_1), \ldots, T(v_t) \) i.i.d G-W trees up to depth \( D_0 \).

2. If \( T(v_i) \cong UNIV_k \) for any \( 1 \leq i \leq t \), Ehrenfeucht value of \( Ext(T_0) \) determined by \( Ext(T_0)|_{Rad[k]} \), hence by \( T_0 \), which is fixed.

3. Let \( GOOD = \bigcup_{i=1}^t \{ T(v_i) \cong UNIV_k \} \). So, under \( GOOD \), Ehrenfeucht value of root \( v \) independent of \( \bar{x}, \bar{y} \).

4. Only need to consider \( BAD = GOOD^c \).
So what happens under BAD?

1. Picture 1: \( X_v \) = Ehrenfeucht value of root \( v \).
   \[ X_v \sim \psi^s_\lambda(\vec{x}, T_0). \]
So what happens under $BAD$?

1. Picture 1: $X_v =$ Ehrenfeucht value of root $v$.
   $X_v \sim \psi^s_\lambda(\vec{x}, T_0)$.

   $Z_v \sim \psi^s_\lambda(\vec{y}, T_0)$. 

$X_v = \text{Ehrenfeucht value of root } v.$

$X_v \sim \psi^s_\lambda(\vec{x}, T_0).$

$Z_v = \text{Ehrenfeucht value of root } v.$

$Z_v \sim \psi^s_\lambda(\vec{y}, T_0).$
So what happens under BAD?

1. Picture 1: $X_v =$ Ehrenfeucht value of root $v$.
   $X_v \sim \psi^s_\lambda(\vec{x}, T_0)$.

   $Z_v \sim \psi^s_\lambda(\vec{y}, T_0)$.

3. When $Y = y$, $\{X_i \neq Z_i\}$ for at least one $i$ happens with probability at most $y \cdot P[X_1 \neq Z_1] = y \cdot \|\vec{x} - \vec{y}\|_{TV}$.
So what happens under BAD?

1. **Picture 1**: \(X_v\) = Ehrenfeucht value of root \(v\).
   \(X_v \sim \psi^s_\lambda(\vec{x}, T_0)\).

2. **Picture 2**: \(Z_v\) = Ehrenfeucht value of root \(v\).
   \(Z_v \sim \psi^s_\lambda(\vec{y}, T_0)\).

3. When \(Y = y\), \(\{X_i \neq Z_i\}\) for at least one \(i\) happens with probability at most \(y \cdot P[X_1 \neq Z_1] = y \cdot ||\vec{x} - \vec{y}||_{TV}\).

4. But for \(\{X_v \neq Z_v\}\), we also require BAD to hold.
So what happens under \textit{BAD}?

1. Picture 1: $X_v = \text{Ehrenfeucht value of root } v$. 
   $X_v \sim \psi^s_\lambda(\vec{x}, T_0)$.

2. Picture 2: $Z_v = \text{Ehrenfeucht value of root } v$. 
   $Z_v \sim \psi^s_\lambda(\vec{y}, T_0)$.

3. When $Y = y$, \{ $X_i \neq Z_i$ \} for at least one $i$ happens with probability at most $y \cdot P[X_1 \neq Z_1] = y \cdot ||\vec{x} - \vec{y}||_{TV}$.

4. But for \{ $X_v \neq Z_v$ \}, we also require \textit{BAD} to hold. Thus

$$||\psi^s_\lambda(\vec{x}, T_0) - \psi^s_\lambda(\vec{y}, T_0)||_{TV} \leq P[X_v \neq Z_v]$$

$$\leq \sum_{y=0}^{\infty} y ||\vec{x} - \vec{y}||_{TV} 1_{BAD} P[Y = y]$$

$$= E[Y 1_{BAD}] ||\vec{x} - \vec{y}||_{TV}.$$
Exponential bound on $P[BAD]$
Exponential bound on $P[\text{BAD}]$

**Lemma**

$$P[\text{BAD}] \leq e^{-t\beta}, \text{ } t \text{ the number of nodes at depth } s_0 \text{ of } T_0.$$
Lemma

\[ P[\text{BAD}] \leq e^{-t\beta}, \ t \text{ the number of nodes at depth } s_0 \text{ of } T_0. \]

Proof.

1. \( \exists v : T(v) \cong \text{UNIV}_k \Rightarrow T \text{ Rad}[k]-universal. \)
Exponential bound on $P[\text{BAD}]$

**Lemma**

$$P[\text{BAD}] \leq e^{-t\beta}, \text{ } t \text{ the number of nodes at depth } s_0 \text{ of } T_0.$$ 

**Proof.**

1. $\exists v : T(v) \cong \text{UNIV}_k \Rightarrow T \text{ Rad}[k]$-universal.
2. Let $P[T(v) \cong \text{UNIV}_k] = 1 - e^{-\beta}$. 
Exponential bound on $P[BAD]$

**Lemma**

$$P[BAD] \leq e^{-t\beta}, \text{ } t \text{ the number of nodes at depth } s_0 \text{ of } T_0.$$  

**Proof.**

1. $\exists v : T(v) \cong UNIV_k \Rightarrow T \text{ Rad}[k]-universal.$
2. Let $P[T(v) \cong UNIV_k] = 1 - e^{-\beta}.$
3. $v_1, \ldots, v_t$ nodes at depth $s_0$. $T(v_1), \ldots, T(v_t)$ i.i.d. G-W up to depth $D_0$. Hence

$$P[BAD] \leq \prod_{i=1}^{t} P[T(v_i) \not\cong UNIV_k] = e^{-t\beta}. $$
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples

Ehrenfeucht games, Ehrenfeucht values

Probabilities of Ehrenfeucht values as fixed point of an iteration

Defining the natural iteration

Our main results

Outlines for proofs

Contraction for $\lambda \geq 1$ - 2-stage process

So finally...

1. $E[Y_1^{BAD}] \leq c_1 E[Y] E[Y_1^{BAD}]$.

2. $E[Y] = t \cdot \lambda D_0 (t)$.

3. Thus $E[||\Psi_s^{\lambda}(\vec{x}, T_0) - \Psi_s^{\lambda}(\vec{y}, T_0)||_{TV}] \leq E[Y_1^{BAD}] ||\vec{x} - \vec{y}||_{TV} \leq c_1 E[Y] E[Y_1^{BAD}] ||\vec{x} - \vec{y}||_{TV}$. 

4. For large $t$, $t \cdot e^{-t \beta}$ very small.
So finally...

1. Can show: $E[Y\mathbf{1}_{BAD}] \leq c_1 E[Y]E[\mathbf{1}_{BAD}]$. 
So finally...

1. Can show: $E[Y 1_{BAD}] \leq c_1 E[Y] E[1_{BAD}]$.
2. $E[Y] = t \cdot \lambda^{D_0}$ (i.i.d copies of G-W trees up to depth $D_0$).
3. Thus

$$E \left[ \| \psi_s(\vec{x}, T_0) - \psi_s(\vec{y}, T_0) \|_{TV} \right] \leq E[Y 1_{BAD}] \| \vec{x} - \vec{y} \|_{TV} \leq c_1 E[Y] E[1_{BAD}] \| \vec{x} - \vec{y} \|_{TV}$$

$$= c_1 \cdot t \lambda^{D_0} \cdot e^{-t\beta} \cdot \| \vec{x} - \vec{y} \|_{TV}$$
So finally...

1. Can show: \( E[Y \mathbb{1}_{BAD}] \leq c_1 E[Y] E[\mathbb{1}_{BAD}] \).
2. \( E[Y] = t \cdot \lambda^{D_0} \) (\( t \) i.i.d copies of G-W trees up to depth \( D_0 \)).
3. Thus

\[
E \left[ \| \Psi_{\lambda}^s(\tilde{x}, T_0) - \Psi_{\lambda}^s(\tilde{y}, T_0) \|_{TV} \right] \leq E[Y \mathbb{1}_{BAD}] \| \tilde{x} - \tilde{y} \|_{TV} \\
\leq c_1 E[Y] E[\mathbb{1}_{BAD}] \| \tilde{x} - \tilde{y} \|_{TV} \\
= c_1 \cdot t \lambda^{D_0} \cdot e^{-t\beta} \cdot \| \tilde{x} - \tilde{y} \|_{TV}
\]

4. For large \( t \), \( t \cdot e^{-t\beta} \) very small.
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

1. Set-up, first order world, examples
   - First Order World

2. Ehrenfeucht games, Ehrenfeucht values
   - Recursive rule for determining Ehrenfeucht class

3. Probabilities of Ehrenfeucht values as fixed point of an iteration
   - Defining the natural iteration
   - Our main results
   - Outlines for proofs
   - Contraction for $\lambda \geq 1$ - 2-stage process

4. Almost sure theory for first order statements
   - Our main results
   - Rapidly determined
   - Outline of proof for an example
   - Universal trees, again!
Our results on almost sure theory for First Order Logic

Theorem (P., Spencer)

Fix $k \in \mathbb{N}$. Fix a finite tree $T_0$. $A[T_0] := \{\exists v : T(v) \sim T_0\} \lor \{T \text{ is finite}\}$. In $T_\lambda$, $A[T_0]$ is almost surely true. $= \Rightarrow$ conditioned on $T_\lambda$ infinite, $\exists v : T(v) \sim T_0$.

Schema $A = \{A[T_0] : \forall T_0 \text{ finite tree}\}$ gives almost sure theory for infinite trees.

Lemma (Consequence of theorem) Conditioned on $T_\lambda$ infinite, Ehrenfeucht value determined by local neighbourhood $B(R, \text{Rad}[k])$ of root. $P[A[T_0]] = P[A^*]$, $A^*$ depends on $B(R, \text{Rad}[k])$. 


### Theorem (P., Spencer)

- \( \text{Fix } k \in \mathbb{N} \).
Our results on almost sure theory for First Order Logic

Theorem (P., Spencer)

- Fix \( k \in \mathbb{N} \).
- Fix a finite tree \( T_0 \).

\[
A[T_0] := \{ \exists v : T(v) \cong T_0 \} \lor \{ T \text{ is finite} \}.
\]
Our results on almost sure theory for First Order Logic

**Theorem (P., Spencer)**

- Fix $k \in \mathbb{N}$.
- Fix a finite tree $T_0$.
  $$A[T_0] := \{\exists v : T(v) \cong T_0\} \lor \{T \text{ is finite}\}.$$  
- In $T_\lambda$, $A[T_0]$ is almost surely true. $\implies$ conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong T_0$. 

- **Schema** 
  $$A = \{A[T_0] : \forall T_0 \text{ finite tree}\}$$ gives almost sure theory for infinite trees.

- **Lemma (Consequence of theorem)**
  Conditioned on $T_\lambda$ infinite, Ehrenfeucht value determined by local neighbourhood $B(R, \text{Rad}[k])$ of root.
  $$P[A[T_0]] = P[A^*], A^* \text{ depends on } B(R, \text{Rad}[k]).$$
Our results on almost sure theory for First Order Logic

**Theorem (P., Spencer)**

- **Fix** $k \in \mathbb{N}$.
- **Fix a finite tree** $T_0$.
  
  \[ A[T_0] := \{ \exists v : T(v) \cong T_0 \} \lor \{ T \text{ is finite} \} \]

- **In** $T_\lambda$, $A[T_0]$ is almost surely true. $\implies$ conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong T_0$.

- **Schema** $\mathcal{A} = \{ A[T_0] : \forall T_0 \text{ finite tree} \}$ gives almost sure theory for infinite trees.
Our results on almost sure theory for First Order Logic

Theorem (P., Spencer)

- Fix $k \in \mathbb{N}$.
- Fix a finite tree $T_0$.
  
  $A[T_0] := \{ \exists v : T(v) \cong T_0 \} \lor \{ T \text{ is finite} \}$.
- In $T_\lambda$, $A[T_0]$ is almost surely true. $\implies$ conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong T_0$.
- Schema $\mathcal{A} = \{ A[T_0] : \forall T_0 \text{ finite tree} \}$ gives almost sure theory for infinite trees.

Lemma (Consequence of theorem)

Conditioned on $T_\lambda$ infinite, Ehrenfeucht value determined by local neighbourhood $B(R, \text{Rad}[k])$ of root.
Our results on almost sure theory for First Order Logic

Theorem (P., Spencer)

- Fix $k \in \mathbb{N}$.
- Fix a finite tree $T_0$.
  \[ A[T_0] := \{ \exists v : T(v) \cong T_0 \} \lor \{ T \text{ is finite} \}. \]
- In $T_\lambda$, $A[T_0]$ is almost surely true. $\implies$ conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong T_0$.
- Schema $\mathcal{A} = \{ A[T_0] : \forall T_0 \text{ finite tree} \}$ gives almost sure theory for infinite trees.

Lemma (Consequence of theorem)

Conditioned on $T_\lambda$ infinite, Ehrenfeucht value determined by local neighbourhood $B(R, \text{Rad}[k])$ of root.
\[ P[A[T_0]] = P[A^*], A^* \text{ depends on } B(R, \text{Rad}[k]). \]
Outline of proof: Rapidly determined properties

1. Recall fictitious continuation $X_1, X_2, \ldots$.
Outline of proof: Rapidly determined properties

1. Recall fictitious continuation $X_1, X_2, \ldots$.
2. Quite surely: Exponentially small failure probability.
Outline of proof: Rapidly determined properties

1. Recall fictitious continuation $X_1, X_2, \ldots$
2. **Quite surely:** Exponentially small failure probability.

**Definition**

A **rapidly determined** if quite surely $A$ tautologically determined by $X_1, \ldots X_s$, $s \in \mathbb{N}$. 

Recall fictitious continuation $X_1, X_2, \ldots$.
Outline of proof: Rapidly determined properties

1. Recall fictitious continuation $X_1, X_2, \ldots$.
2. **Quite surely:** Exponentially small failure probability.

**Definition**

A **rapidly determined** if quite surely $A$ tautologically determined by $X_1, \ldots X_s, s \in \mathbb{N}$.

$P[A \text{ not determined by } X_1 \ldots X_s] \leq e^{-\beta s}$, $\beta$ independent of $s$. 
Outline of proof: Rapidly determined properties

1. Recall fictitious continuation $X_1, X_2, \ldots$.
2. **Quite surely:** Exponentially small failure probability.

**Definition**

A **rapidly determined** if quite surely A tautologically determined by $X_1, \ldots X_s$, $s \in \mathbb{N}$. 

$P[A \text{ not determined by } X_1 \ldots X_s] \leq e^{-\beta s}$, $\beta$ independent of $s$.

**Theorem (P., Spencer)**

$A[T_0]$ rapidly determined for every fixed $T_0$. 
Proof for example: one child and one grandchild

1. $i$: $i$ has one child with one child. (makes sense because of fictitious continuation).
Proof for example: one child and one grandchild

1. $I_i$: $i$ has one child with one child. (makes sense because of fictitious continuation).

2. $Y = \sum_{i \leq s_\epsilon} I_i$,
Proof for example: one child and one grandchild

1. $I_i$: $i$ has one child with one child. (makes sense because of fictitious continuation).
2. $Y = \sum_{i \leq s} I_i, \quad (\lambda \epsilon < 1 - \epsilon)$. 
Proof for example: one child and one grandchild

1. \( I_i \): \( i \) has one child with one child. (makes sense because of fictitious continuation).

2. \( Y = \sum_{i \leq s} I_i \), \( (\lambda \epsilon < 1 - \epsilon) \).

3. \( E[Y] = s\epsilon \cdot (\lambda e^{-\lambda})^2 \).
Proof for example: one child and one grandchild

1. \( l_i: \) \( i \) has one child with one child. (makes sense because of fictitious continuation).
2. \( Y = \sum_{i \leq s} l_i, \) \( \lambda \varepsilon < 1 - \varepsilon \).
3. \( E[Y] = s \varepsilon \cdot (\lambda e^{-\lambda})^2. \)
4. Martingale \( Y_0, Y_1, \ldots, Y_s: \) \( Y_i = E[Y|X_1, \ldots, X_i], \)
   \( Y_0 = E[Y]. \)
Proof for example: one child and one grandchild

1. \( I_i: i \) has one child with one child. (makes sense because of fictitious continuation).
2. \( Y = \sum_{i \leq s} I_i, \quad (\lambda \epsilon < 1 - \epsilon). \)
3. \( E[Y] = s \epsilon \cdot (\lambda e^{-\lambda})^2. \)
4. Martingale \( Y_0, Y_1, \ldots, Y_s: Y_i = E[Y|X_1, \ldots, X_i], Y_0 = E[Y]. \)
5. Can show: quite surely \( Y_s = Y. \)
Proof for example: one child and one grandchild

1. $I_i$: $i$ has one child with one child. (makes sense because of fictitious continuation).

2. $Y = \sum_{i \leq s} I_i$, \quad ($\lambda \epsilon < 1 - \epsilon$).

3. $E[Y] = s\epsilon \cdot (\lambda e^{-\lambda})^2$.

4. Martingale $Y_0, Y_1, \ldots, Y_s$: $Y_i = E[Y|X_1, \ldots, X_i]$, $Y_0 = E[Y]$.

5. Can show: quite surely $Y_s = Y$.

6. Lipschitz: $|Y_i - Y_{i-1}| \leq 2$. Only $I_i, I_{\pi(i)}$ affected by revealing $X_i$. 
Proof for example: one child and one grandchild

1. $I_i$: $i$ has one child with one child. (makes sense because of fictitious continuation).
2. $Y = \sum_{i \leq s} I_i$, $(\lambda \epsilon < 1 - \epsilon)$.
3. $E[Y] = s\epsilon \cdot (\lambda e^{-\lambda})^2$.
4. Martingale $Y_0, Y_1, \ldots, Y_s$: $Y_i = E[Y|X_1, \ldots, X_i]$, $Y_0 = E[Y]$.
5. Can show: quite surely $Y_s = Y$.
6. Lipschitz: $|Y_i - Y_{i-1}| \leq 2$. Only $I_i, I_{\pi(i)}$ affected by revealing $X_i$.
7. Azuma’s inequality $\Rightarrow P[Y < \xi s] \leq e^{-\varphi s}$, $\xi, \varphi$ independent of $s$.

Therefore: $A$ is rapidly determined.
Why Ehrenfeucht value determined by neighbourhood of root
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree $\text{UNIV}_k$. 

Remark (A parting remark) Nice way of visualizing $\text{UNIV}_k$: as a Christmas tree. Hang sufficiently many strings, of sufficiently long length, from the root. Hang a "ball" at the end of each string, each (somewhat refined) Ehrenfeucht class having $k$ representative balls.
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree \( UNIV_k \).
2. Conditioned on \( T_\lambda \) infinite, \( \exists v : T(v) \cong UNIV_k \).
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree $UNIV_k$.
2. Conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong UNIV_k$.
3. Hence Ehrenfeucht value determined by $B(R, Rad[k])$. 

Remark (A parting remark) Nice way of visualizing $UNIV_k$: as a Christmas tree. Hang sufficiently many strings, of sufficiently long length, from the root. Hang a "ball" at the end of each string, each (somewhat refined) Ehrenfeucht class having $k$ representative balls.
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree \( UNIV_k \).
2. Conditioned on \( T_\lambda \) infinite, \( \exists v : T(v) \cong UNIV_k \).
3. Hence Ehrenfeucht value determined by \( B(R, \text{Rad}[k]) \).

**Remark (A parting remark)**

*Nice way of visualizing UNIV\(_k\): as a Christmas tree.*
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree \( \text{UNIV}_k \).
2. Conditioned on \( T_\lambda \) infinite, \( \exists v : T(v) \cong \text{UNIV}_k \).
3. Hence Ehrenfeucht value determined by \( B(R, \text{Rad}[k]) \).

Remark (A parting remark)

*Nice way of visualizing \( \text{UNIV}_k \): as a Christmas tree. Hang sufficiently many strings,*
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree $UNIV_k$.
2. Conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong UNIV_k$.
3. Hence Ehrenfeucht value determined by $B(R, Rad[k])$.

**Remark (A parting remark)**

*Nice way of visualizing $UNIV_k$: as a Christmas tree. Hang sufficiently many strings, of sufficiently long length, from the root.*
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree $UNIV_k$.
2. Conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong UNIV_k$.
3. Hence Ehrenfeucht value determined by $B(R, Rad[k])$.

**Remark (A parting remark)**

*Nice way of visualizing $UNIV_k$: as a Christmas tree. Hang sufficiently many strings, of sufficiently long length, from the root. Hang a “ball” at the end of each string,*
Why Ehrenfeucht value determined by neighbourhood of root

1. Recall **finite** universal tree $UNIV_k$.
2. Conditioned on $T_\lambda$ infinite, $\exists v : T(v) \cong UNIV_k$.
3. Hence Ehrenfeucht value determined by $B(R, Rad[k])$.

Remark (A parting remark)

Nice way of visualizing $UNIV_k$: as a Christmas tree. Hang sufficiently many strings, of sufficiently long length, from the root. Hang a “ball” at the end of each string, each (somewhat refined) Ehrenfeucht class having $k$ representative balls.
First order properties and probabilities for Galton-Watson trees in the Poisson regime

Moumanti Podder

Set-up, first order world, examples
Ehrenfeucht games, Ehrenfeucht values
Probabilities of Ehrenfeucht values as fixed point of an iteration
Almost sure theory for first order statements

Our main results
Rapidly determined

Outline of proof for an example
Universal trees, again!

Thank you!