Expressive Completeness of Two-Variable First-Order Logic with Counting for First-Order Logic Queries on Rooted Unranked Trees

Jelle Hellings†  Marc Gyssens‡  Jan Van den Bussche‡  Dirk Van Gucht§

† Department of Computing and Software
McMaster University
Hamilton, Ontario, Canada
https://jhellings.nl

‡ Data Science Institute
Hasselt University
Diepenbeek, Belgium

§ Luddy School of Informatics, Computing, and Engineering
Indiana University
Bloomington, Indiana, USA
The Result

Theorem (Theorem 37)

Let \( \varphi \) be an unary first-order query.

There exists an FO\(^2\)+C query \( \psi \) that is equivalent to \( \varphi \) on trees.
The Result

Theorem (Theorem 37)

Let $\varphi$ be an unary first-order query. There exists an $\text{FO}^2+C$ query $\psi$ that is equivalent to $\varphi$ on trees.

- **Unary first-order queries** on graphs express **node predicates**: operations to restrict the considered nodes within more complex graph queries.
- $\text{FO}^2+C$: first-order logic, restricted to two variables, with counting quantifiers such as

  $$\exists v (\exists_3 =_3 w \text{ edge}(v, w)), \quad \forall v (\exists_5 \leq_5 w \text{ edge}(v, w)).$$

- **Trees**: node-labeled, unranked, and unordered.
  - **Unranked** Nodes do not have a fixed number of children.
  - **Unordered** Siblings are not ordered.

**Extensions** Edge-labeled trees, forests, ....
Related Work

- Similar results are known on strings with a successor relationship.

- Marx and de Rijke considered ordered trees with a descendant- and sibling-axis. They showed that unary FO\(^2\) queries are equivalent to Core XPath.

- ten Cate and Marx showed that binary FO queries are equivalent to Core XPath 2.0.

- Marx showed that binary first-order queries are equivalent to Conditional XPath (Conditional XPath is an algebraization of FO\(^3\) with a limited transitive closures).

- Hellings et al. showed that unary Conditional XPath queries are equivalent to a variant of FO\(^2\) with fixpoints.
FO$^2$+C Queries on Trees

- **Root with three children:**
  \[
  \exists v (\text{root}(v) \land \exists w \text{edge}(v, w) \land C_1 \land C_2 \land C_3).
  \]

- **One has two children (all leaves):**
  \[
  \exists v \exists w \text{edge}(v, w) \land \exists v \exists w \text{edge}(w, v) \land \exists w \text{leaf}(w).
  \]

- **One is a leaf:**
  \[
  \exists w \text{edge}(v, w) \land \text{leaf}(w).
  \]

- **One has three children (all leaves):**
  \[
  \exists w \text{edge}(v, w) \land \exists v \exists w \text{edge}(w, v) \land \exists w \text{leaf}(w).
  \]
FO$^2+C$ Queries on Trees

- Root with three children:
  \[(\exists^1 v \ (\text{root}(v) \land (\exists^3 w \ \text{edge}(v, w)) \land C_1 \land C_2 \land C_3)).\]
FO²+C Queries on Trees

- Root with three children:
  \[(\exists^=1 v \ (\text{root}(v) \land (\exists^=3 w \ \text{edge}(v, w))) \land C_1 \land C_2 \land C_3).\]

- One has two children (all leaves):

  \[
  C_1 := \exists^=1 w \ (\text{edge}(v, w) \land (\exists^=2 v \ \text{edge}(w, v))) \land (\exists^=2 v \ \text{edge}(w, v) \land \text{leaf}(v))).
  \]

- One is a leaf:

  \[
  C_2 := \exists^=1 w \ (\text{edge}(v, w) \land \text{leaf}(w)).
  \]

- One has three children (all leaves):

  \[
  C_3 := \exists^=1 w \ (\text{edge}(v, w) \land (\exists^=3 v \ \text{edge}(w, v))) \land (\exists^=3 v \ \text{edge}(w, v) \land \text{leaf}(v))).
  \]
Lemma
Let $\varphi$ be a unary first-order query, let $\mathcal{T} = (N, E)$ be an unlabeled tree, and let $n \in N$.

1. There exists an unary $\text{FO}^2+\text{C}$ query $tq_{\mathcal{T}}$ such that

$$\models_{\mathcal{T}} tq_{\mathcal{T}} \neq \emptyset$$

if and only if trees $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic.

2. There exists an unary $\text{FO}^2+\text{C}$ query $tn_{\mathcal{T}}$ such that

$$\models_{\mathcal{T}} tn_{\mathcal{T}} = \models_{\mathcal{T}} \varphi.$$

Main challenge Argue that we can conceptually restrict $T$ to a finite set.
Lemma

Let $\varphi$ be a unary first-order query, let $\mathcal{T} = (N, E)$ be an unlabeled tree, and let $n \in \mathbb{N}$.

1. There exists an unary $\text{FO}^2+C$ query $tq_{\mathcal{T}}$ such that

   $$[tq_{\mathcal{T}}]_{\mathcal{T}'} \neq \emptyset$$

   if and only if trees $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic.

2. There exists an unary $\text{FO}^2+C$ query $tn_{\mathcal{T}}$ such that

   $$[tn_{\mathcal{T}}]_{\mathcal{T}} = [\varphi]_{\mathcal{T}}.$$

3. Let $\mathbb{T}$ be the set of all trees. The query $\varphi$ is equivalent to $\text{FO}^2+C$ query

   $$Q_\varphi := \bigvee_{\mathcal{T}' \in \mathbb{T}} \left( (\exists v \ (tq_{\mathcal{T}'}) \land tn_{\mathcal{T}'}) \right).$$

Main challenge: Argue that we can conceptually restrict $\mathbb{T}$ to a finite set.
FO²+C Queries on Trees

Lemma
Let $\varphi$ be a unary first-order query, let $\mathcal{T} = (N, E)$ be an unlabeled tree, and let $n \in N$.

1. There exists an unary FO²+C query $tq_\mathcal{T}$ such that
   $$\llbracket tq_\mathcal{T} \rrbracket_{\mathcal{T}'} \neq \emptyset$$
   if and only if trees $\mathcal{T}$ and $\mathcal{T}'$ are isomorphic.

2. There exists an unary FO²+C query $tn_\mathcal{T}$ such that
   $$\llbracket tn_\mathcal{T} \rrbracket_{\mathcal{T}} = \llbracket \varphi \rrbracket_{\mathcal{T}}.$$

3. Let $\mathcal{T}$ be the set of all trees. The query $\varphi$ is equivalent to FO²+C query
   $$Q_\varphi := \bigvee_{\mathcal{T}' \in \mathcal{T}} \left( \exists v \ (tq_{\mathcal{T}'}) \right) \land tn_{\mathcal{T}'}.$$

Main challenge Argue that we can conceptually restrict $\mathcal{T}$ to a finite set.
Hanf Locality

Let $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ be a tree and let $n \in \mathcal{N}$.

**Definition**
The $d$-neighborhood around $n$ is the set of nodes (subtree) reachable from $n$ via a path of at-most $d$ edges.

**Definition**
Two trees are $(d, m)$-equivalent if they have the same amount (up-till-$m$) of each $d$-neighborhood.
Hanf Locality

Let $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ be a tree and let $n \in \mathcal{N}$.

**Definition**
The *d-neighborhood* around $n$ is the set of nodes (subtree) reachable from $n$ via a path of at-most $d$ edges.

**Definition**
Two trees are *(d, m)-equivalent* if they have the *same amount* (up-till-$m$) of each $d$-neighborhood.
Let $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ be a tree and let $n \in \mathcal{N}$.

**Definition**
The *d-neighborhood* around $n$ is the set of nodes (subtree) reachable from $n$ via a path of at-most $d$ edges.

**Definition**
Two trees are \((d, m)\)-equivalent if they have the same amount (up-till-\(m\)) of each $d$-neighborhood.
Let $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ be a tree and let $n \in \mathcal{N}$.

**Definition**
The *d-neighborhood* around $n$ is the set of nodes (subtree) reachable from $n$ via a path of at-most $d$ edges.

**Definition**
Two trees are \((d, m)\)-equivalent if they have the same amount (up-till-$m$) of each $d$-neighborhood.

**Lemma (Fagin et al.)**
*If every node has at-most $f$ children, then there is a finite number of distinct $d$-neighborhoods (up-to-isomorphisms).*
Hanf Locality

Let $\mathcal{T} = (\mathcal{N}, \mathcal{E})$ be a tree and let $n \in \mathcal{N}$.

**Definition**

The *$d$-neighborhood* around $n$ is the set of nodes (subtree) reachable from $n$ via a path of at-most $d$ edges.

**Definition**

Two trees are *(d, m)-equivalent* if they have the *same amount* (up-till-$m$) of each $d$-neighborhood.

**Theorem (Fagin et al.)**

Let $r$ be a positive integer. If every node has at-most $f$ children, then there exists $d, m$ that only depend on $r, f$ such that if two trees are *(d, m)-equivalent*, then they are indistinguishable by $r$-round EF-games.
Hanf Locality

Hanf locality: we can restrict the depth of trees we consider.

Limitations of Hanf Locality

We consider unranked trees!

All four nodes have distinct $d$-neighborhoods, $d \geq 1$. 
Hanf Locality

Hanf locality: we can restrict the depth of trees we consider.

Limitations of Hanf Locality

We consider unranked trees!

All four nodes have distinct $d$-neighborhoods, $d \geq 1$.

Our main technical contribution

For trees, we need a stronger locality notion that takes into account branching. 

_Paper_: provide such a notion and show how it relates to FO$^2$+C and first-order expressivity.
Bounded Equivalence on *Nodes*

Let $\mathcal{T}_1 = (N_1, E_1)$ and $\mathcal{T}_2 = (N_2, E_2)$ be two trees.

**Definition (Definition 2)**

Nodes $n_1 \in N_1, n_2 \in N_2$ are *downward $(b, d)$-bounded equivalent* ($n_1 \approx_{\downarrow b,d} n_2$) if

- (they have the same node labels); and
- $d = 0$ or else the children of $n_1, n_2$ can be grouped into equivalence classes based on $\approx_{\downarrow b,d-1}$, and these classes for the children of $n_1, n_2$ have *the same size* (up-till-$b$).

**Definition (Definition 5)**

Nodes $n_1 \in N_1, n_2 \in N_2$ are *$(b, d)$-bounded equivalent* ($n_1 \approx_{b,d} n_2$) if

- $d = 0$ and $n_1 \approx_{\downarrow b,0} n_2$; or
- $n_1 \approx_{\downarrow b,d} n_2$ and both $n_1$ and $n_2$ are roots; or
- $n_1 \approx_{\downarrow b,d} n_2, n_1$ and $n_2$ have parents $p_1$ and $p_2$, and $p_1 \approx_{b,d-1} p_2$. 
Bounded Equivalence on *Nodes*
Bounded Equivalence on *Nodes*

\[(b, 0)\)-bounded equivalence classes

![Diagram of bounded equivalence classes]

6/8
Bounded Equivalence on *Nodes*

(2, 1)-bounded equivalence classes
Bounded Equivalence on *Nodes*

(3, 1)-bounded equivalence classes
Bounded Equivalence on *Nodes*

(3, 2)-bounded equivalence classes

(uncolored nodes are all in distinct equivalence classes)
Bounded Equivalence on Nodes

The 2-neighborhoods of (3, 2)-bounded equivalent nodes are not isomorph!
(but there does exist a ‘unique’ minimum-sized 2-neighborhood)
Theorem (Lemma 34(3) and consequence of Theorem 37)

1. There exists a finite number of distinct \((b, d)\)-bounded equivalence classes (with respect to a given set of node labels).

2. Given a \((b, d)\)-bounded equivalence class \(C\), there exists an \(\text{FO}^2+C\) query \(q\) such that

\[
    n \in \llbracket q \rrbracket_T \text{ if and only if } n \in C
\]

for every tree \(T\).
Bounded Equivalence on Trees

Let $\mathcal{T}_1 = (N_1, E_1)$ and $\mathcal{T}_2 = (N_2, E_2)$ be two trees.

Definition (Definition 29)
Trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are \((b, d, k)\)-bounded equivalent ($\mathcal{T}_1 \approx_{b,d,k} \mathcal{T}_2$) if

- for each node $n_1 \in N_1$, there is a node $n_2 \in N_2$ with $n_1 \approx_{b,d} n_2$ and vice versa; and
- for all nodes $m \in (N_1 \cup N_2)$ such that $M_1 \subseteq N_1$ and $M_2 \subseteq N_2$ are all nodes that are \((b, d)\)-bounded equivalent to $m$, the \((b, d')\)-equivalence classes of ancestors of nodes in $M_1$ and $M_2$ at distance $2d' + 1$, $0 \leq d' \leq d$, must have the same size (up-till-$k$).

Theorem (Lemma 34(4))
Given a tree $\mathcal{T}$, there exists a Boolean $\text{FO}^2 + \mathcal{C}$ query $q$ such that

$$[q]_{\mathcal{T}'} \neq \emptyset \text{ if and only if } \mathcal{T} \approx_{b,d,k} \mathcal{T}'.$$
Bounded Equivalence on Trees

Let $\mathcal{T}_1 = (N_1, E_1)$ and $\mathcal{T}_2 = (N_2, E_2)$ be two trees.

Definition (Definition 29)
Trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are $(b, d, k)$-bounded equivalent ($\mathcal{T}_1 \approx_{b, d, k} \mathcal{T}_2$) if

- for each node $n_1 \in N_1$, there is a node $n_2 \in N_2$ with $n_1 \approx_{b, d} n_2$ and vice versa; and
- for all nodes $m \in (N_1 \cup N_2)$ such that $M_1 \subseteq N_1$ and $M_2 \subseteq N_2$ are all nodes that are $(b, d)$-bounded equivalent to $m$, the $(b, d')$-equivalence classes of ancestors of nodes in $M_1$ and $M_2$ at distance $2d' + 1, 0 \leq d' \leq d$, must have the same size (up-till-$k$).

Theorem (Theorem 32)
Let $n_1 \in N_1$, $n_2 \in N_2$, $r \geq 0$, and $d = 7^r - 1$, $b = r + 2$, $k = 4d + 4$.
If $\mathcal{T}_1 \approx_{b, d, k} \mathcal{T}_2$ and $n_1 \approx_{b, d} n_2$, then $n_1$ and $n_2$ are indistinguishable by $r$-round EF-games.
Conclusion and Future Work

We have shown that any unary first-order query on node-labeled, unranked, and unordered trees can be rewritten into an equivalent query in FO²+C.

Future work

- Succinctness?
- Can we generalize our results to other classes of graphs? E.g., forests or restricted classes of DAGs.
- Can we refine our results, e.g., based on the number of variables used: can we relate FOⁿ to FO²+C with counting quantifiers that can only count to some-function-of-n?
- How does our result impact practical query answering on trees? E.g., can an algebraization of FO²+C aid in semi-join-based query optimizations?
References


