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 φ be an unary first-order query.
There exists an FO^2+C query ψ that is equivalent to φ on trees.

The Result

Theorem (Theorem 37)

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

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

$$\exists v\ (\exists^{=3} w\ \ \mathrm{edge}(v, w)), \qquad \forall v\ (\exists^{\leq 5} w\ \ \mathrm{edge}(v, w)).$$

- *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, ...

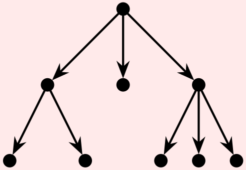
Related Work

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

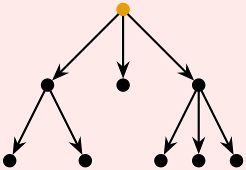
- Marx and de Rijke considered *ordered trees* with a descendant- and sibling-axis. They showed that *unary* FO^2 *queries* are equivalent to *Core XPath*.
- ten Cate and Marx showed that *binary* FO *queries* are equivalent to *Core XPath 2.0*.
- Marx showed that *binary first-order queries* are equivalent 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



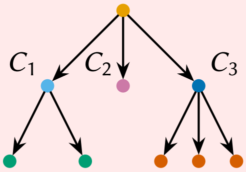
FO^2+C Queries on Trees



- Root with three children:

  $(\exists^{=1} v\ (\mathrm{root}(v) \wedge (\exists^{=3} w\ \ \mathrm{edge}(v, w)) \wedge C_1 \wedge C_2 \wedge C_3)$.

FO^2+C Queries on Trees



- Root with three children:

$$(\exists^{=1} v\ (\mathrm{root}(v) \wedge (\exists^{=3} w\ \ \mathrm{edge}(v, w)) \wedge C_1 \wedge C_2 \wedge C_3).$$

- One has two children (all leaves):

$$C_1 := \exists^{=1} w\ (\mathrm{edge}(v, w) \wedge (\exists^{=2} v\ \ \mathrm{edge}(w, v)) \wedge (\exists^{=2} v\ \ \mathrm{edge}(w, v) \wedge \mathrm{leaf}(v))).$$

- One is a leaf:

$$C_2 := \exists^{=1} w\ (\mathrm{edge}(v, w) \wedge\ \mathrm{leaf}(w)).$$

- One has three children (all leaves):

$$C_3 := \exists^{=1} w\ (\mathrm{edge}(v, w) \wedge (\exists^{=3} v\ \ \mathrm{edge}(w, v)) \wedge (\exists^{=3} v\ \ \mathrm{edge}(w, v) \wedge \mathrm{leaf}(v))).$$

FO^2+C Queries on Trees

Lemma

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

1. *There exists an unary* FO^2+C *query* $\mathsf{tq}_{\mathcal{T}}$ *such that*

$$[\![\mathsf{tq}_{\mathcal{T}}]\!]_{\mathcal{T}'} \neq \emptyset$$

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

2. *There exists an unary* FO^2+C *query* $\mathsf{tn}_{\mathcal{T}}$ *such that*

$$[\![\mathsf{tn}_{\mathcal{T}}]\!]_{\mathcal{T}} = [\![\varphi]\!]_{\mathcal{T}}.$$

FO^2+C Queries on Trees

Lemma

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

1. *There exists an unary* FO^2+C *query* $\mathsf{tq}_{\mathcal{T}}$ *such that*

$$[\![\mathsf{tq}_{\mathcal{T}}]\!]_{\mathcal{T}'} \neq \emptyset$$

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

2. *There exists an unary* FO^2+C *query* $\mathsf{tn}_{\mathcal{T}}$ *such that*

$$[\![\mathsf{tn}_{\mathcal{T}}]\!]_{\mathcal{T}} = [\![\varphi]\!]_{\mathcal{T}}.$$

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

$$Q_\varphi := \bigvee_{\mathcal{T}' \in \mathbb{T}} \Big((\exists v\ (\mathsf{tq}_{\mathcal{T}'})) \wedge \mathsf{tn}_{\mathcal{T}'} \Big).$$

FO^2+C Queries on Trees

Lemma

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

1. *There exists an unary* FO^2+C *query* $\mathsf{tq}_{\mathscr{T}}$ *such that*

$$[\![\mathsf{tq}_{\mathscr{T}}]\!]_{\mathscr{T}'} \neq \emptyset$$

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

2. *There exists an unary* FO^2+C *query* $\mathsf{tn}_{\mathscr{T}}$ *such that*

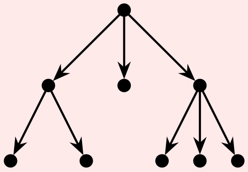
$$[\![\mathsf{tn}_{\mathscr{T}}]\!]_{\mathscr{T}} = [\![\varphi]\!]_{\mathscr{T}}.$$

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

$$Q_\varphi := \bigvee_{\mathscr{T}' \in \mathbb{T}} \Big((\exists v\ (\mathsf{tq}_{\mathscr{T}'})) \land \mathsf{tn}_{\mathscr{T}'} \Big).$$

Main challenge Argue that we can conceptually restrict $\mathbb{T}$ to a *finite set*.

Hanf Locality



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

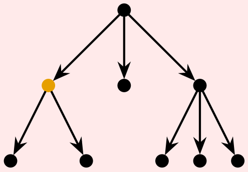
Definition

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 $n \in \mathcal{N}$.

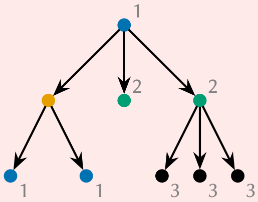
Definition

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.

Hanf Locality



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

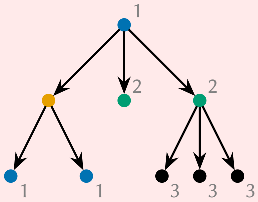
Definition

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.

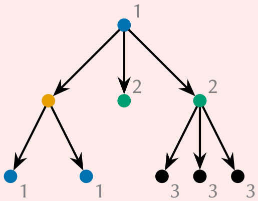
Definition

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

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 $\mathscr{T} = (\mathcal{N}, \mathcal{E})$ 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.)

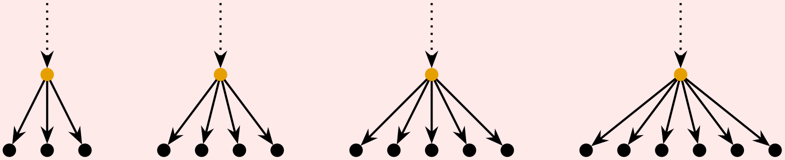
Let r be a positive integer. If every node has at-most f children, then there exists d, m that only depend on r, f such that if two trees are (d, m)-equivalent, then they are indistinguishable by r-round EF-games.

Hanf Locality

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

Limitations of Hanf Locality

We consider *unranked* trees!



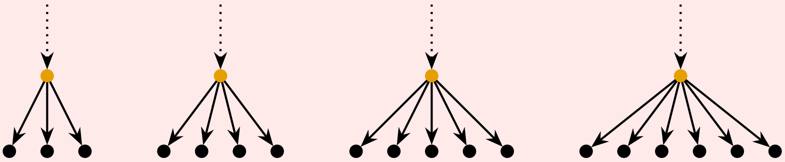
All four nodes have distinct d-neighborhoods, $d \geq 1$.

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 $\mathcal{T}_1 = (\mathcal{N}_1, \mathcal{E}_1)$ and $\mathcal{T}_2 = (\mathcal{N}_2, \mathcal{E}_2)$ be two trees.

Definition (Definition 2)

Nodes $n_1 \in \mathcal{N}_1, n_2 \in \mathcal{N}_2$ are *downward (b, d)-bounded equivalent* ($n_1 \approx_{\downarrow b,d} n_2$) if

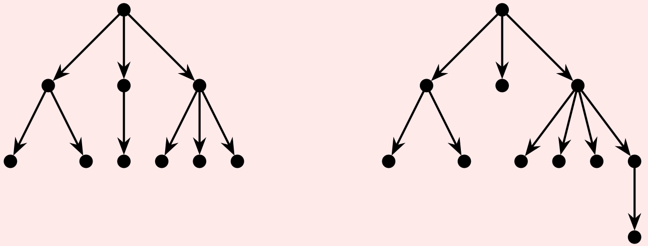
- (they have the same node labels); and
- $d = 0$ or else the children of n_1, n_2 can be grouped into equivalence classes based on $\approx_{\downarrow b,d-1}$, and these classes for the children of n_1, n_2 have *the same size* (up-till-b).

Definition (Definition 5)

Nodes $n_1 \in \mathcal{N}_1, n_2 \in \mathcal{N}_2$ are *(b, d)-bounded equivalent* ($n_1 \approx_{b,d} n_2$) if

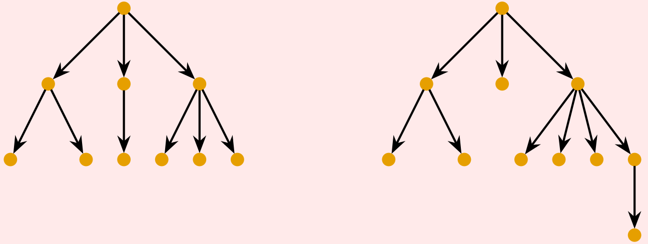
- $d = 0$ and $n_1 \approx_{\downarrow b,0} n_2$; or
- $n_1 \approx_{\downarrow b,d} n_2$ and both n_1 and n_2 are roots; or
- $n_1 \approx_{\downarrow b,d} n_2$, n_1 and n_2 have parents p_1 and p_2, and $p_1 \approx_{b,d-1} p_2$.

Bounded Equivalence on *Nodes*



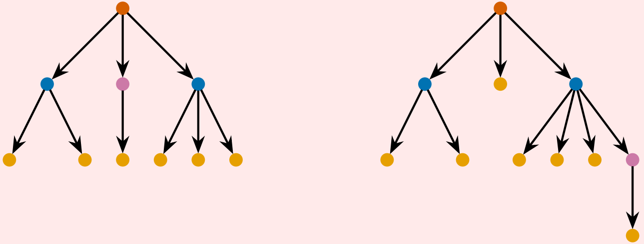
Bounded Equivalence on *Nodes*

$(b, 0)$-bounded equivalence classes



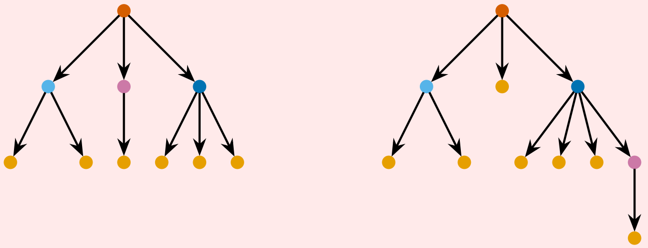
Bounded Equivalence on *Nodes*

(2, 1)-bounded equivalence classes



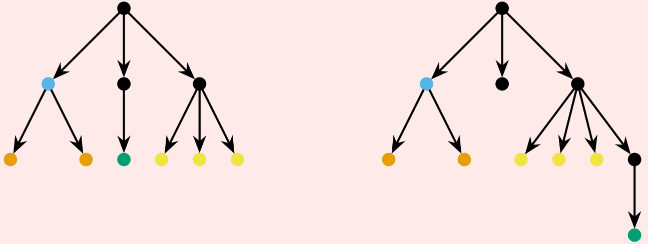
Bounded Equivalence on *Nodes*

(3, 1)-bounded equivalence classes



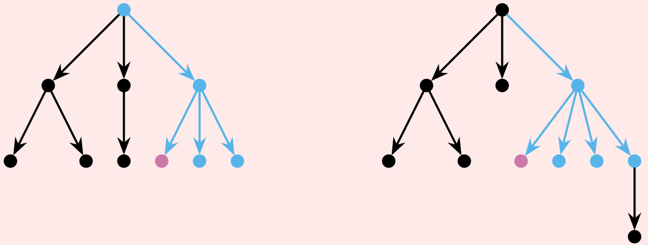
Bounded Equivalence on *Nodes*

$(3, 2)$-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)

Bounded Equivalence on *Nodes*

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* $\mathrm{FO}^2+\mathrm{C}$ *query q such that*

$$n \in [\![q]\!]_{\mathcal{T}} \text{ if and only if } n \in C$$

for every tree $\mathcal{T}$.

Bounded Equivalence on *Trees*

Let $\mathcal{T}_1 = (\mathcal{N}_1, \mathcal{E}_1)$ and $\mathcal{T}_2 = (\mathcal{N}_2, \mathcal{E}_2)$ 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 $n_1 \in \mathcal{N}_1$, there is a node $n_2 \in \mathcal{N}_2$ with $n_1 \approx_{b,d} n_2$ and vice versa; and
- for all nodes $m \in (\mathcal{N}_1 \cup \mathcal{N}_2)$ such that $M_1 \subseteq \mathcal{N}_1$ and $M_2 \subseteq \mathcal{N}_2$ 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 FO^2+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 $\mathcal{T}_1 = (\mathcal{N}_1, \mathcal{E}_1)$ and $\mathcal{T}_2 = (\mathcal{N}_2, \mathcal{E}_2)$ 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 $n_1 \in \mathcal{N}_1$, there is a node $n_2 \in \mathcal{N}_2$ with $n_1 \approx_{b,d} n_2$ and vice versa; and
- for all nodes $m \in (\mathcal{N}_1 \cup \mathcal{N}_2)$ such that $M_1 \subseteq \mathcal{N}_1$ and $M_2 \subseteq \mathcal{N}_2$ 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 $n_1 \in \mathcal{N}_1$, $n_2 \in \mathcal{N}_2$, $r \geq 0$, 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 FO^2+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 FO^n to FO^2+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 FO^2+C aid in semi-join-based query optimizations?

References

[1] Balder ten Cate and Maarten Marx. “Navigational XPath: Calculus and Algebra”. In: *SIGMOD Record* 36.2 (2007), pp. 19–26. DOI: 10.1145/1328854.1328858.

[2] Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. “On Monadic NP vs Monadic co-NP”. In: *Information and Computation* 120.1 (1995), pp. 78–92. DOI: 10.1006/inco.1995.1100.

[3] Jelle Hellings, Catherine L Pilachowski, Dirk Van Gucht, Marc Gyssens, and Yuqing Wu. “From Relation Algebra to Semi-join Algebra: An Approach to Graph Query Optimization”. In: *The Computer Journal* 64.5 (2020), pp. 789–811. DOI: 10.1093/comjnl/bxaa031.

[4] Maarten Marx. “Conditional XPath”. In: *ACM Transactions on Database Systems* 30.4 (2005), pp. 929–959. DOI: 10.1145/1114244.1114247.

[5] Maarten Marx and Maarten de Rijke. “Semantic Characterizations of Navigational XPath”. In: *SIGMOD Record* 34.2 (2005), pp. 41–46. DOI: 10.1145/1083784.1083792.