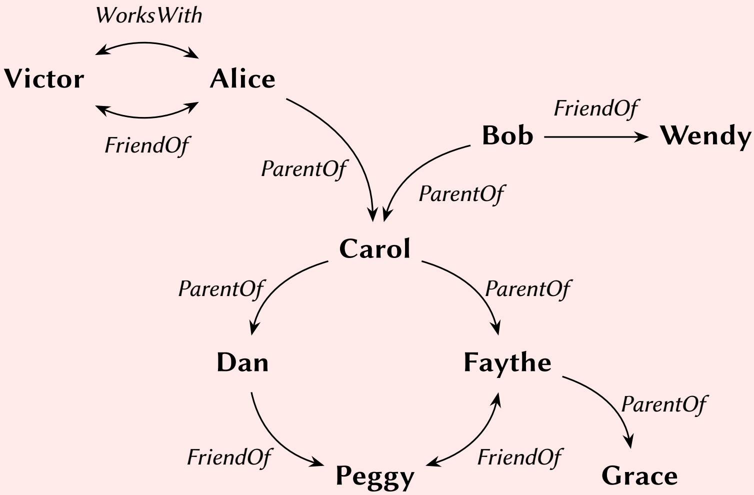
The Power of Tarski's Relation Algebra on Trees

Jelle Hellings[1]

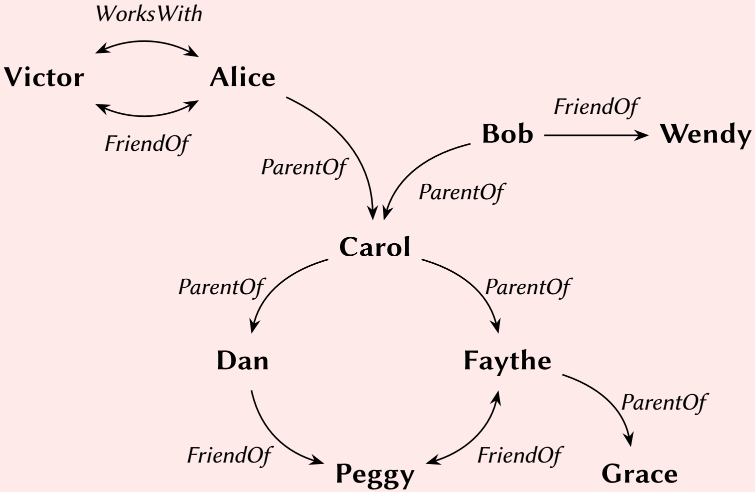
Yuqing Wu[2] Marc Gyssens[1] Dirk Van Gucht[3]

[1] Hasselt University
[2] Pomona College
[3] Indiana University

Relation algebra: graphs and queries

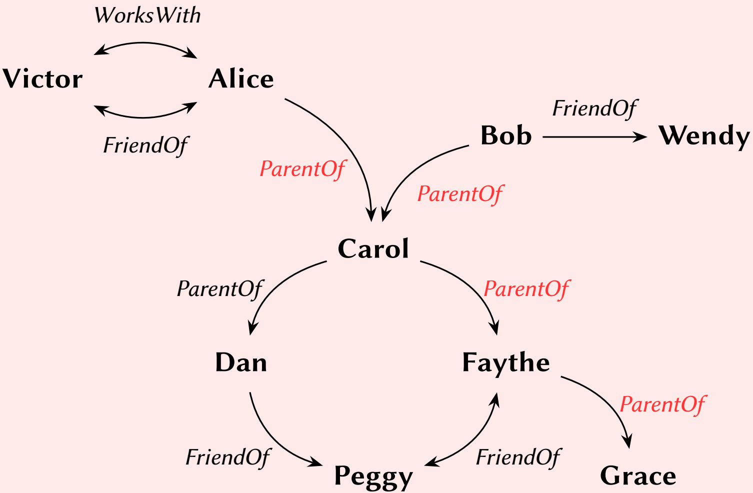


Relation algebra: graphs and queries



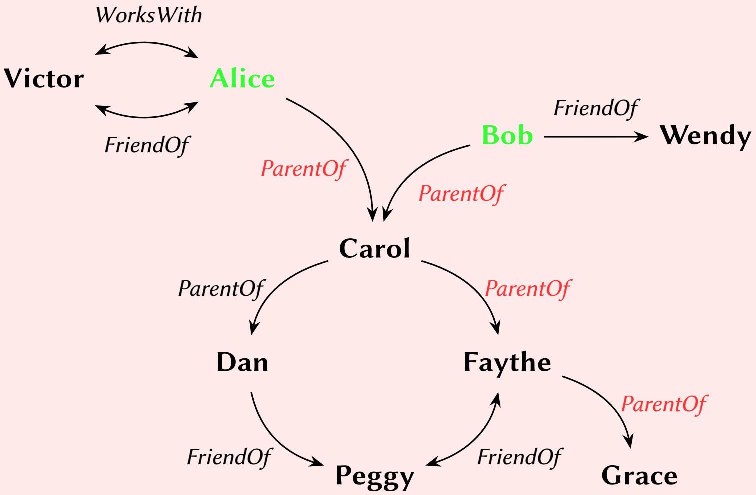
$$\mathit{FriendsOfGgp} = \pi_1[\mathit{ParentOf} \circ \mathit{ParentOf} \circ \mathit{ParentOf}] \circ \mathit{FriendOf}$$

Relation algebra: graphs and queries



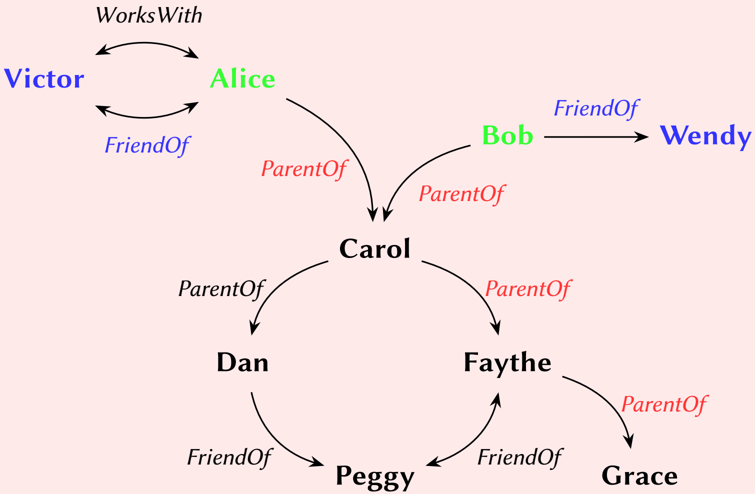
$$\mathit{FriendsOfGgp} = \pi_1[\mathit{ParentOf} \circ \mathit{ParentOf} \circ \mathit{ParentOf}] \circ \mathit{FriendOf}$$

Relation algebra: graphs and queries


WorksWith
Victor
Alice
FriendOf
ParentOf
Bob
FriendOf
Wendy
ParentOf
Carol
ParentOf
ParentOf
Dan
Faythe
ParentOf
FriendOf
FriendOf
Peggy
Grace

$$\mathit{FriendsOfGgp} = \pi_1[\mathit{ParentOf} \circ \mathit{ParentOf} \circ \mathit{ParentOf}] \circ \mathit{FriendOf}$$

Relation algebra: graphs and queries



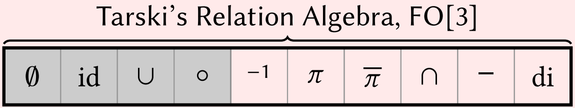
$$FriendsOfGgp = \pi_1[ParentOf \circ ParentOf \circ ParentOf] \circ FriendOf$$

Relation algebra: other operators

$$\begin{aligned}
\mathit{ChildOf} &= \mathit{ParentOf}^{-1} \\
\mathit{AcquaintanceOf} &= \mathit{FriendOf} \cup \mathit{WorksWith} \\
\mathit{WorkFriendOf} &= \mathit{FriendOf} \cap \mathit{WorksWith} \\
\mathit{NonWorkFriend} &= \mathit{FriendOf} - \mathit{WorksWith} \\
\mathit{Parent} &= \pi_1[\mathit{ParentOf}] \\
\mathit{NonParent} &= \overline{\pi}_1[\mathit{ParentOf}] \\
\mathit{GrandParentOf} &= \mathit{ParentOf} \circ \mathit{ParentOf}
\end{aligned}$$

Also: constants $\emptyset$, id, di.

Relation algebra and query languages



Tarski's Relation Algebra, FO[3]

$\emptyset$	id	$\cup$	$\circ$	$^{-1}$	π	$\overline{\pi}$	$\cap$	$\overline{\ }$	di

- On graphs: RPQs, 2RPQs, Nested RPQs, Graph XPath.
- XPath of Benedikt et al.
- Path$^+$ of Wu et al.
- Downward queries of Hellings et al.
- Conditional and Navigational XPath.

Questions about the relation algebra

Questions

- Are all these operators necessary?
- What does each operator add?

Questions about the relation algebra

Questions

- Are all these operators necessary?
- What does each operator add?

What is the expressive power of fragments of the relation algebra?

Expressive power of relation algebra

Background: when are queries equivalent?

Two types of queries and equivalences

Path-queries. The exact query result is important:

$$\mathit{FriendOf} \cap \mathit{WorksWith} \equiv_{\text{path}} \mathit{FriendOf} - (\mathit{FriendOf} - \mathit{WorksWith}).$$

Boolean-queries. The existence of a query result is important:

$$\mathit{FriendOf}^{-1} \circ \mathit{ParentOf} \equiv_{\text{bool}} \pi_1[\mathit{FriendOf}] \cap \pi_1[\mathit{ParentOf}].$$

Definition

Language $\mathcal{L}_1$ is z-subsumed by $\mathcal{L}_2$ if every query in $\mathcal{L}_1$ is z-equivalent to a query in $\mathcal{L}_2$ (denoted by $\mathcal{L}_1 \preceq_z \mathcal{L}_2$).

Background: query fragments

Let $\mathcal{F} \subseteq \{\mathrm{di}, {}^{-1}, \pi, \overline{\pi}, \cap, -\}$.

- We write $\mathcal{N}(\mathcal{F})$: only allows $\emptyset$, id, ℓ, $\circ$, $\cup$, and all operators in $\mathcal{F}$.
- We write $\mathcal{N}(\underline{\mathcal{F}})$ to represent all operators expressible in $\mathcal{N}(\mathcal{F})$ using the following basic rewrite rules:

$$\begin{aligned}
\pi_1[e] &= \overline{\pi}_j[\overline{\pi}_1[e]] = (e \circ e^{-1}) \cap \mathrm{id} = (e \circ (\mathrm{id} \cup \mathrm{di})) \cap \mathrm{id};\\
\pi_2[e] &= \overline{\pi}_j[\overline{\pi}_2[e]] = (e^{-1} \circ e) \cap \mathrm{id} = ((\mathrm{id} \cup \mathrm{di}) \circ e) \cap \mathrm{id};\\
\overline{\pi}_i[e] &= \mathrm{id} - \pi_i[e];\\
e_1 \cap e_2 &= e_1 - (e_1 - e_2).
\end{aligned}$$

Examples

- $\mathcal{N}(\underline{\overline{\pi}}) = \mathcal{N}(\pi, \overline{\pi})$.
- $\mathcal{N}(\underline{{}^{-1}, -}) = \mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$.

Results on graphs

Previous work by Fletcher et al.

Relative expressive power of relation algebra fragments *on graphs*:

- Path queries: each operator adds expressive power.
- Boolean queries: $^{-1}$ sometimes does not add expressive power.

Results on graphs

Previous work by Fletcher et al.

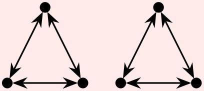
Relative expressive power of relation algebra fragments *on graphs*:

- Path queries: each operator adds expressive power.
- Boolean queries: $^{-1}$ sometimes does not add expressive power.

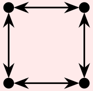
Example: strong Boolean separations

Consider $(\mathcal{E} \circ \mathcal{E}) \cap \mathcal{E}$:

$\mathcal{G}_{3,3}$:



$\mathcal{G}_4$:



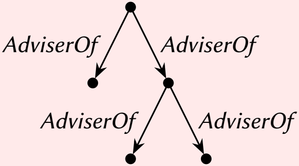
Without $\cap$ (or $-$): no query distinguishes $\mathcal{G}_{3,3}$ and $\mathcal{G}_4$.

Our work: studying trees

Goal

Improve understanding of the relation algebra by studying the expressive power in *restricted* settings.

Trees



- Trees are hierarchical/nested structures.
  E.g. taxonomies, organizational charts, documents, file systems.
- Tree-based formal data models and query languages
  E.g. XML data with XPath, JSON data, nested relational data.

Initial classification

		downward	non-downward		
non-local	non-monotone			$\mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \overline{\pi})$	$\mathcal{N}$
	monotone			$\mathcal{N}(\mathrm{di}, {}^{-1}, \pi)$	$\mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \cap)$
local	non-monotone	$\mathcal{N}(\pi, \overline{\pi}, \cap, -)$	$\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap)$		$\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$
	monotone	$\mathcal{N}(\cap, -)$ $\mathcal{N}(\pi, \cap)$	$\mathcal{N}({}^{-1}, \pi, \cap)$		
		1-subtree reducible		2-subtree reducible	3-subtree reducible

Initial classification

	downward	non-downward			
non-local			$\mathcal{N}(\text{di}, {}^{-1}, \pi, \overline{\pi})$	$\mathcal{N}$	non-monotone
			$\mathcal{N}(\text{di}, {}^{-1}, \pi)$	$\mathcal{N}(\text{di}, {}^{-1}, \pi, \cap)$	monotone
local	$\mathcal{N}(\pi, \overline{\pi}, \cap, -)$	$\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap)$		$\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$	non-monotone
	$\mathcal{N}(\cap, -)$ $\mathcal{N}(\pi, \cap)$	$\mathcal{N}({}^{-1}, \pi, \cap)$			monotone
	1-subtree reducible		2-subtree reducible	3-subtree reducible	

- $\mathcal{N}(\pi, \overline{\pi}, \cap, -)$ studied by Hellings et al.

Initial classification

		downward	non-downward		
		1-subtree reducible		2-subtree reducible	3-subtree reducible
non-local	non-monotone			$\mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \overline{\pi})$	$\mathcal{N}$
	monotone			$\mathcal{N}(\mathrm{di}, {}^{-1}, \pi)$	$\mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \cap)$
local	non-monotone	$\mathcal{N}(\pi, \overline{\pi}, \cap, -)$	$\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap)$		$\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$
	monotone	$\mathcal{N}(\cap, -)$ $\mathcal{N}(\pi, \cap)$	$\mathcal{N}({}^{-1}, \pi, \cap)$		

- $\mathcal{N}(\pi, \overline{\pi}, \cap, -)$ studied by Hellings et al.
- $\mathcal{N}({}^{-1}, \pi)$ studied by Benedikt et al.
- $\mathcal{N}({}^{-1}, \pi, \cap)$ studied by Wu et al.

Downward queries: Boolean and path queries

$$\begin{tikzcd}
\mathcal{N}(\pi,\overline{\pi},\cap,-) & \mathcal{N}(\overline{\pi}) \\
\mathcal{N}(\pi,\cap) & \mathcal{N}(\pi) \arrow[u] \\
\mathcal{N}(\cap,-) & \mathcal{N}() \arrow[u]
\end{tikzcd}$$

- $\mathcal{N}(\pi,\overline{\pi},\cap,-)$ is downward.
- $^{-1}$ and di are not downward.

Theorem (Hellings et al.)

Let $\mathcal{F} \subseteq \{\pi,\overline{\pi},\cap,-\}$.
We have $\mathcal{N}(\mathcal{F}) \preceq_{\text{path}} \mathcal{N}(\underline{\mathcal{F}} - \{\cap,-\})$.

Local queries

Locality

Definition

A query q is *local* if there exists $k \geq 0$ such that, for every tree $\mathcal{T}$, and for all nodes m and n, $(m, n) \in [\![q]\!]_{\mathcal{T}}$ if and only if $(m, n) \in [\![q]\!]_{\mathcal{T}'}$, with $\mathcal{T}'$ the smallest subtree of $\mathcal{T}$ containing all nodes at distance at most k from the nearest common ancestor of m and n.

- $\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$ is local.
- di is not local.

Local queries: main results

Path queries

$$\begin{tikzcd}
 & \begin{array}{r} \mathcal{N}(^{-1},\pi,\overline{\pi},\cap,-) \\ \mathcal{N}(^{-1},-) \end{array} \\
 & \begin{array}{r} \mathcal{N}(^{-1},\pi,\overline{\pi},\cap) \\ \mathcal{N}(^{-1},\overline{\pi}) \end{array} \arrow[u] \\
\begin{array}{r} \mathcal{N}(\pi,\overline{\pi},\cap,-) \\ \mathcal{N}(\overline{\pi}) \end{array} \arrow[ur] & \\
 & \begin{array}{r} \mathcal{N}(^{-1},\pi,\cap) \\ \mathcal{N}(^{-1},\pi) \end{array} \arrow[uu] \\
\begin{array}{r} \mathcal{N}(\pi,\cap) \\ \mathcal{N}(\pi) \end{array} \arrow[uu] \arrow[ur] & \\
 & \mathcal{N}(^{-1}) \arrow[uu] \\
\begin{array}{r} \mathcal{N}(\cap,-) \\ \mathcal{N}() \end{array} \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

Boolean queries

$$\begin{tikzcd}
\begin{array}{r} \mathcal{N}(^{-1},\pi,\overline{\pi},\cap,-) \\ \mathcal{N}(^{-1},-) \end{array} \\
\begin{array}{r} \mathcal{N}(\pi,\overline{\pi},\cap,-) \\ \mathcal{N}(^{-1},\pi,\overline{\pi},\cap) \\ \mathcal{N}(\overline{\pi}) \end{array} \arrow[u] \\
\begin{array}{r} \mathcal{N}(\pi,\cap) \\ \mathcal{N}(^{-1},\pi,\cap) \\ \mathcal{N}(\pi) = \mathcal{N}(^{-1}) \end{array} \arrow[u] \\
\begin{array}{r} \mathcal{N}(\cap,-) \\ \mathcal{N}() \end{array} \arrow[u]
\end{tikzcd}$$

Condition tree queries

Definition

If an expression always evaluates to a subset of id then it is called a *node expression*.

Definition

A *condition tree query* $\mathcal{Q}$ is a tuple $\mathcal{Q} = (\mathcal{T}, C, \mathrm{s}, \mathrm{t}, \gamma)$ with

- $\mathcal{T} = (\mathcal{V}, \Sigma, \mathbf{E})$ a labeled tree,
- C is a set of *node expressions*,
- $\mathrm{s} \in \mathcal{V}$ is the *source node*,
- $\mathrm{t} \in \mathcal{V}$ is the *target node*,
- $\gamma \subseteq \mathcal{V} \times C$ is the *node-condition relation*.

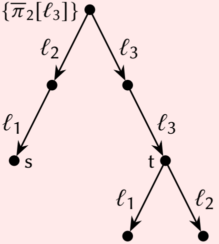
$\gamma(n)$ denotes the set $\{e_1, \ldots, e_k\}$ of conditions related to node $n \in \mathcal{V}$.
$\gamma(n)$ also denotes the expression $e_1 \circ \ldots \circ e_k$ ($\equiv_{\mathrm{path}}$ $e_1 \cap \ldots \cap e_k$).

A condition tree query example

Consider the union-free expression

$$\ell_1^{-1} \circ \ell_2^{-1} \circ \overline{\pi}_2[\ell_3] \circ \ell_3 \circ \ell_3 \circ \pi_1[\ell_1] \circ \pi_1[\ell_2].$$

This expression is path-equivalent to

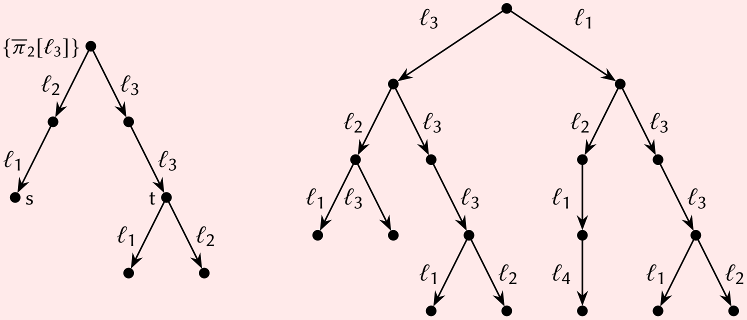


A condition tree query example

Consider the union-free expression

$$\ell_1^{-1} \circ \ell_2^{-1} \circ \overline{\pi}_2[\ell_3] \circ \ell_3 \circ \ell_3 \circ \pi_1[\ell_1] \circ \pi_1[\ell_2].$$

This expression is path-equivalent to

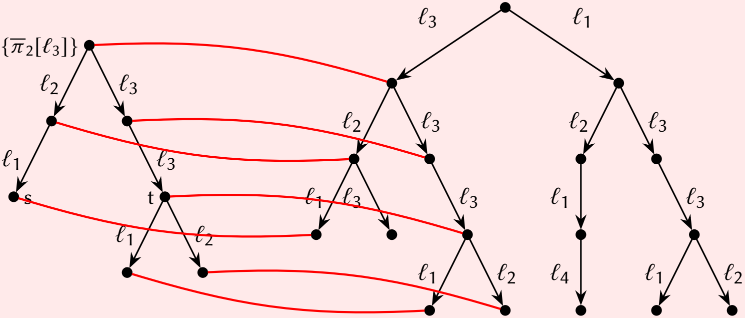


A condition tree query example

Consider the union-free expression

$$\ell_1^{-1} \circ \ell_2^{-1} \circ \overline{\pi}_2[\ell_3] \circ \ell_3 \circ \ell_3 \circ \pi_1[\ell_1] \circ \pi_1[\ell_2].$$

This expression is path-equivalent to

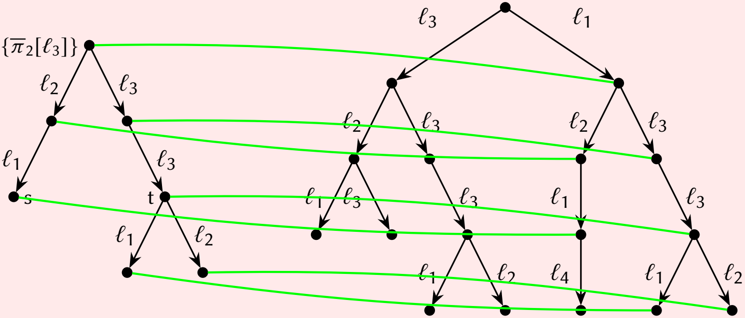


A condition tree query example

Consider the union-free expression

$$\ell_1^{-1} \circ \ell_2^{-1} \circ \overline{\pi}_2[\ell_3] \circ \ell_3 \circ \ell_3 \circ \pi_1[\ell_1] \circ \pi_1[\ell_2].$$

This expression is path-equivalent to



Condition tree queries and local queries

$\mathbf{Q}_{\text{tree}}(\mathcal{F})$ are the condition tree queries in which node conditions are restricted to union-free expressions in $\mathcal{N}(\mathcal{F})$.

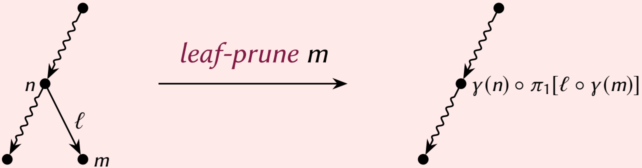
Claim

Let $\{^{-1}, \pi\} \subseteq \mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}\}$.
$\mathbf{Q}_{\text{tree}}(\mathcal{F})$ *and union-free* $\mathcal{N}(\mathcal{F})$ *path-subsume each other.*

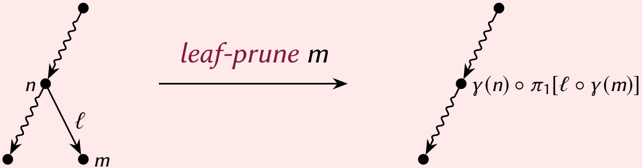
Proposition

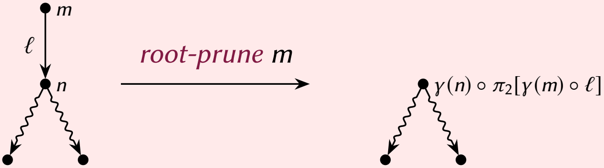
Let $\mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}\}$ *and* Q *be a condition tree query in* $\mathbf{Q}_{\text{tree}}(\mathcal{F})$.
There exists a union-free expression e *in* $\mathcal{N}(\mathcal{F})$ *such that* $e \equiv_{\text{path}} Q$.

Basic manipulations on condition tree queries



Basic manipulations on condition tree queries

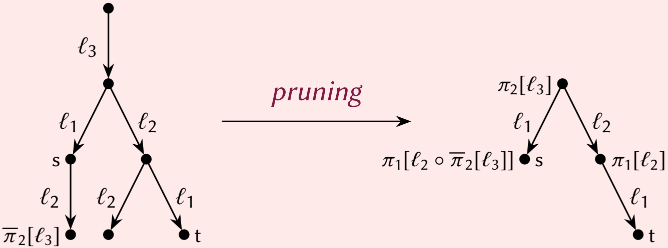




Up-down queries

Definition

A condition tree query $Q = (\mathcal{T}, C, \mathrm{s}, \mathrm{t}, \gamma)$ is an *up-down query* if all edges of $\mathcal{T}$ are on the unique path from s to t not taking into account the direction of the edges.



Lemma

Let $\{\pi\} \subseteq \underline{\mathcal{F}} \subseteq \{^{-1}, \overline{\pi}, \pi\}$, *and* Q *be a condition tree query in* $\mathbf{Q}_{\text{tree}}(\mathcal{F})$. *There exists an up-down query* Q' *in* $\mathbf{Q}_{\text{tree}}(\mathcal{F})$ *such that* $Q \equiv_{\text{path}} Q'$.

Up-down queries and local queries

Claim

Let $\{^{-1}, \pi\} \subseteq \mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}\}$.

$\mathbf{Q}_{\mathsf{tree}}(\mathcal{F})$ *and union-free* $\mathcal{N}(\mathcal{F})$ *path-subsume each other.*

Up-down queries and local queries

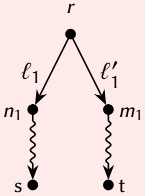
Claim

Let $\{^{-1}, \pi\} \subseteq \mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}\}$.
$\mathbf{Q}_{\mathrm{tree}}(\mathcal{F})$ *and union-free* $\mathcal{N}(\mathcal{F})$ *path-subsume each other.*

Proposition

Let $\mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}\}$ *and* e *be a union-free expression in* $\mathcal{N}(\mathcal{F})$.
There exists a condition tree query Q *in* $\mathbf{Q}_{\mathrm{tree}}(\mathcal{F})$ *such that* $e \equiv_{\mathrm{path}} Q$.

Proof.



$$\xrightarrow{\textit{direct rewrite}}$$

$$\gamma(\mathsf{s}) \circ \ldots \circ \gamma(n_1) \circ \ell_1^{-1} \circ \gamma(r) \circ \ell_1' \circ \gamma(m_1) \circ \ldots \circ \gamma(\mathsf{t})$$

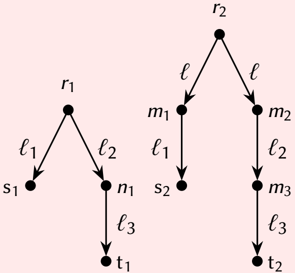
Path-queries: eliminate intersection

Theorem

Let $\{^{-1}, \pi\} \subseteq \underline{\mathcal{F}} \subseteq \{^{-1}, \pi, \overline{\pi}, \cap\}$.
We have $\mathcal{N}(\mathcal{F}) \preceq_{\text{path}} \mathcal{N}(\mathcal{F} - \{\cap\})$.

Proof.

$\mathcal{Q}_1 \quad \cap \quad \mathcal{Q}_2$

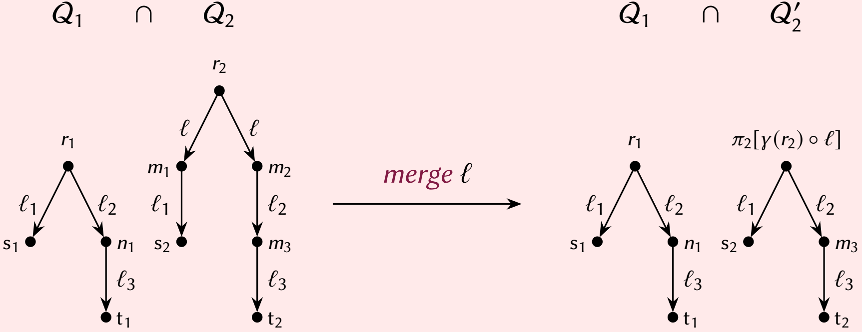


Path-queries: eliminate intersection

Theorem

Let $\{^{-1}, \pi\} \subseteq \underline{\mathcal{F}} \subseteq \{^{-1}, \pi, \overline{\pi}, \cap\}$.
We have $\mathcal{N}(\mathcal{F}) \preceq_{\mathsf{path}} \mathcal{N}(\mathcal{F} - \{\cap\})$.

Proof.



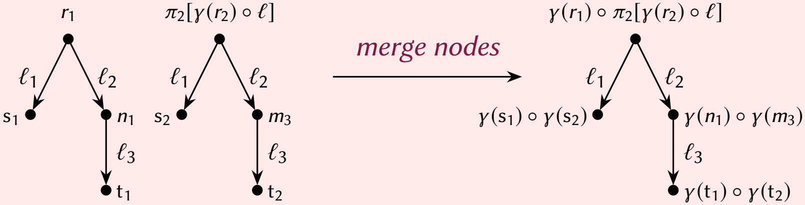
Path-queries: eliminate intersection

Theorem

Let $\{^{-1}, \pi\} \subseteq \underline{\mathcal{F}} \subseteq \{^{-1}, \pi, \overline{\pi}, \cap\}$.
We have $\mathcal{N}(\mathcal{F}) \preceq_{\text{path}} \mathcal{N}(\mathcal{F} - \{\cap\})$.

Proof.

$\mathcal{Q}_1 \quad \cap \quad \mathcal{Q}_2'$ $\qquad\qquad$ $\mathcal{Q}_1 \cap \mathcal{Q}_2$



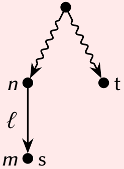
Boolean queries: eliminate projection or converse

Proposition

$\mathcal{N}(\pi) \preceq_{\text{bool}} \mathcal{N}(^{-1})$ *and* $\mathcal{N}(^{-1}) \preceq_{\text{bool}} \mathcal{N}(\pi)$*.*

Proof.

Wu et al.: union-free $\mathcal{N}(\pi)$ is path-equivalence to $\mathbf{Q}_{\text{tree}}()$.



Boolean queries: eliminate projection or converse

Proposition

$\mathcal{N}(\pi) \preceq_{\text{bool}} \mathcal{N}({}^{-1})$ *and* $\mathcal{N}({}^{-1}) \preceq_{\text{bool}} \mathcal{N}(\pi)$.

Proof.

Wu et al.: union-free $\mathcal{N}(\pi)$ is path-equivalence to $\mathbf{Q}_{\text{tree}}()$.

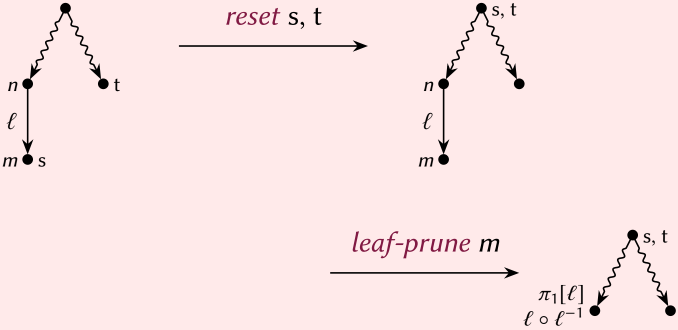
Boolean queries: eliminate projection or converse

Proposition

$\mathcal{N}(\pi) \preceq_{\text{bool}} \mathcal{N}({}^{-1})$ *and* $\mathcal{N}({}^{-1}) \preceq_{\text{bool}} \mathcal{N}(\pi)$.

Proof.

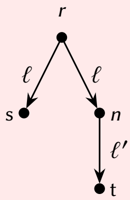
Wu et al.: union-free $\mathcal{N}(\pi)$ is path-equivalence to $\mathbf{Q}_{\text{tree}}()$.



Bonus: condition tree queries on chains

Definition

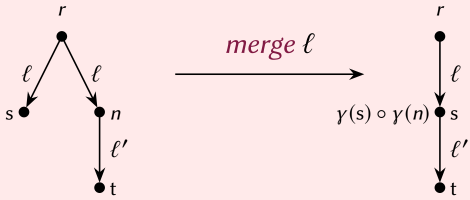
A condition tree query is a *chain query* if it is up-down and $\mathcal{T}$ consists of a single path.



Bonus: condition tree queries on chains

Definition

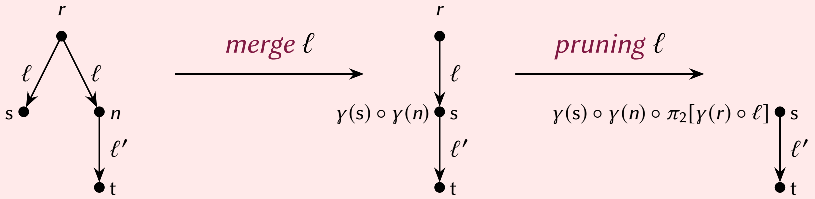
A condition tree query is a *chain query* if it is up-down and $\mathcal{T}$ consists of a single path.



Bonus: condition tree queries on chains

Definition

A condition tree query is a *chain query* if it is up-down and $\mathcal{T}$ consists of a single path.



Lemma

Let $\{\pi\} \subseteq \underline{\mathcal{F}} \subseteq \{^{-1}, \overline{\pi}, \pi\}$*, and* Q *be a condition tree query in* $\mathbf{Q}_{\text{tree}}(\mathcal{F})$*. There exists a chain query* Q' *in* $\mathbf{Q}_{\text{tree}}(\mathcal{F})$ *such that* $Q \equiv_{\text{path}} Q'$*.*

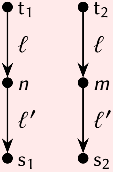
Bonus: eliminate difference on chains

Theorem

Let $\{^{-1}, \overline{\pi}\} \subseteq \mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}, \cap, -\}$.
On chains, we have $\mathcal{N}(\mathcal{F}) \equiv_{\mathrm{path}} \mathcal{N}(^{-1}, \overline{\pi})$.

Proof.

$$Q_1 - Q_2$$



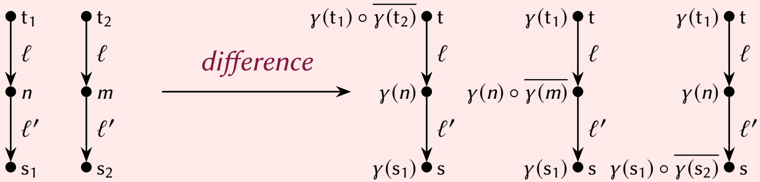
Bonus: eliminate difference on chains

Theorem

Let $\{^{-1}, \overline{\pi}\} \subseteq \mathcal{F} \subseteq \{^{-1}, \pi, \overline{\pi}, \cap, -\}$.
On chains, we have $\mathcal{N}(\mathcal{F}) \equiv_{\text{path}} \mathcal{N}(^{-1}, \overline{\pi})$.

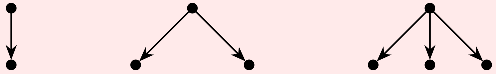
Proof.

$Q_1 \; - \; Q_2$



Separation results

Recognizing branches



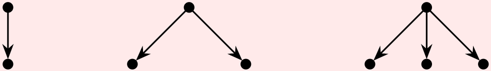
Definition

A *k*-subtree-reduction step on tree $\mathcal{T}$ consists of removing a subtree rooted at a node *n* with parent *m*, given that parent *m* has at least *k* other children isomorphic to *n*.

Theorem

1. $\mathcal{N}(\mathrm{di}, \cap)$ *and* $\mathcal{N}({}^{-1}, -)$ *are closed under* 3*-subtree-reductions.*
2. $\mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \overline{\pi})$ *is closed under* 2*-subtree-reductions.*
3. $\mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap)$ *is closed under* 1*-subtree-reductions.*

The subtree-reductions classification is strict



Proposition

Already on unlabeled trees, we have

1. $\mathcal{N}(\mathrm{di}) \not\leq_{\mathrm{bool}} \mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap)$,
2. $\mathcal{N}({}^{-1}, -) \not\leq_{\mathrm{bool}} \mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap)$,
3. $\mathcal{N}(\mathrm{di}, \cap) \not\leq_{\mathrm{bool}} \mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \overline{\pi})$.

Proof.

1. $e_1 = \mathcal{E} \circ \mathrm{di} \circ \mathrm{di} \circ \mathcal{E}$.
2. $e_2 = (\mathcal{E}^{-1} \circ \mathcal{E}) - \mathrm{id}$.
3. $e_3 = (((\mathrm{di} \circ \mathcal{E}) \cap \mathrm{di}) \circ ((\mathrm{di} \circ \mathcal{E}) \cap \mathrm{di})) \cap \mathrm{di}$.

Diversity and jumping in trees

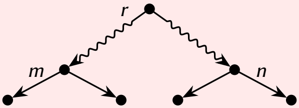
Proposition

Already on unlabeled trees, $\mathcal{N}(\mathrm{di}, \cap) \not\leq_{\mathrm{bool}} \mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$.

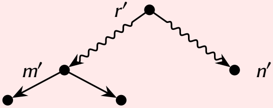
Proof.

$$\begin{aligned} e &= P_{2,\neg r} \circ \mathrm{di} \circ P_{2,\neg r} \\ P_{2,\neg r} &= \pi_2[\mathcal{E}] \circ P_2 \\ P_2 &= \pi_1[S_2] \\ S_2 &= (\mathcal{E} \circ \mathrm{di}) \cap \mathcal{E}. \end{aligned}$$

$\mathcal{T}_1$:



$\mathcal{T}_2$:



Diversity and jumping in chains

Proposition

Already on chains, we have

1. $\mathcal{N}(\mathrm{di}, {}^{-1}) \not\leq_{\text{bool}} \mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$;
2. $\mathcal{N}(\mathrm{di}, \pi) \not\leq_{\text{bool}} \mathcal{N}({}^{-1}, \pi, \overline{\pi}, \cap, -)$.

Proof.

1. $e_1 = (\ell \circ \ell^{-1}) \circ \mathrm{di} \circ (\ell \circ \ell^{-1})$.
2. $e_2 = \pi_1[\ell] \circ \mathrm{di} \circ \pi_1[\ell]$.

C_1:

C_2:

Brute-force: path queries

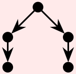
Proposition

Already on unlabeled trees, we have

1. $\mathcal{N}(^{-1}) \not\leq_{\text{path}} \mathcal{N}(\text{di}, \pi, \overline{\pi})$,
2. $\mathcal{N}(\pi) \not\leq_{\text{path}} \mathcal{N}(^{-1})$.

Proof.

1. $e_1 = \mathcal{E}^{-1}$
2. $e_2 = \pi_1[\mathcal{E}] \circ \pi_2[\mathcal{E}]$



Brute-force: Boolean queries

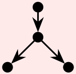
Proposition

Already on unlabeled trees, we have

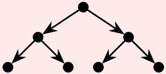
1. $\mathcal{N}(\mathrm{di}, {}^{-1}) \not\leq_{\text{bool}} \mathcal{N}(\mathrm{di})$ *and* $\mathcal{N}(\mathrm{di}, \pi) \not\leq_{\text{bool}} \mathcal{N}(\mathrm{di})$,
2. $\mathcal{N}(\mathrm{di}, {}^{-1}, \cap) \not\leq_{\text{bool}} \mathcal{N}(\mathrm{di}, \pi, \overline{\pi}, \cap, -)$,
3. $\mathcal{N}(\mathrm{di}, \pi) \not\leq_{\text{bool}} \mathcal{N}(\mathrm{di}, {}^{-1})$.

Proof.

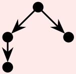
$\mathcal{T}_1$:



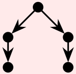
$\mathcal{T}_2$:



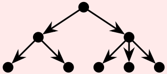
$\mathcal{T}_3$:



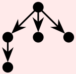
$\mathcal{T}_1'$:



$\mathcal{T}_2'$:



$\mathcal{T}_3'$:



Conclusion and discussion

Overview of results

			Boolean queries						Path queries					
			di	$^{-1}$	π	$\overline{\pi}$	$\cap$	$-$	di	$^{-1}$	π	$\overline{\pi}$	$\cap$	$-$
Local queries	**Downward queries**	$\mathcal{N}()$	✗	✗•	✗•	✗•	✓•	✓•	✗•	✗•	✗•	✗•	✓•	✓•
		$\mathcal{N}(\cap)$	✗	✗•	✗•	✗•		✓•	✗•	✗•	✗•	✗•		✓•
		$\mathcal{N}(\cap, -)$	✗•	✗•	✗•	✗•			✗•	✗•	✗•	✗•		
		$\mathcal{N}(\pi)$	✗	✓•		✗•	✓•	✗•	✗•	✗•		✗•	✓•	✗•
		$\mathcal{N}(\pi, \cap)$	✗	✓		✗•		✗•	✗•	✗•		✗•		✗•
		$\mathcal{N}(\pi, \overline{\pi})$	✗	✓•			✓•	✓•	✗•	✗•			✓•	✓•
		$\mathcal{N}(\pi, \overline{\pi}, \cap)$	✗	✓				✓•	✗•	✗•				✓•
		$\mathcal{N}(\pi, \overline{\pi}, \cap, -)$	✗	✗					✗•	✗•				
		$\mathcal{N}(^{-1})$	✗		✓	✗•	✓	✗•	✗•		✗•	✗•	✗	✗•
		$\mathcal{N}(^{-1}, \pi)$	✗			✗•	✓•	✗•	✗•			✗•	✓•	✗•
		$\mathcal{N}(^{-1}, \pi, \cap)$	✗			✗•		✗•	✗•			✗•		✗•
		$\mathcal{N}(^{-1}, \pi, \overline{\pi})$	✗				✓	✗	✗•				✓	✗
		$\mathcal{N}(^{-1}, \pi, \overline{\pi}, \cap)$	✗					✗	✗•					✗
		$\mathcal{N}(^{-1}, \pi, \overline{\pi}, \cap, -)$	✗						✗•					
		$\mathcal{N}(\mathrm{di})$		✗	✗	✗•	✗	✗•		✗	✗	✗•	✗	✗•
		$\mathcal{N}(\mathrm{di}, \pi)$		✓•		✗•	✗	✗•		✗		✗•	✗	✗•
		$\mathcal{N}(\mathrm{di}, \pi, \cap)$		✗		✗•		✗•		✗		✗•		✗•
		$\mathcal{N}(\mathrm{di}, \pi, \overline{\pi})$		✓•			✗	✗		✗			✗	✗
		$\mathcal{N}(\mathrm{di}, \pi, \overline{\pi}, \cap)$		✗				?		✗				?
		$\mathcal{N}(\mathrm{di}, \pi, \overline{\pi}, \cap, -)$		✗						✗				
		$\mathcal{N}(\mathrm{di}, ^{-1})$			✗	✗•	✗	✗•			✗	✗•	✗	✗•
		$\mathcal{N}(\mathrm{di}, ^{-1}, \pi)$				✗•	✗	✗•				✗•	✗	✗•
		$\mathcal{N}(\mathrm{di}, ^{-1}, \pi, \cap)$				✗•		✗•				✗•		✗•
		$\mathcal{N}(\mathrm{di}, ^{-1}, \pi, \overline{\pi})$					✗	✗					✗	✗
		$\mathcal{N}(\mathrm{di}, ^{-1}, \pi, \overline{\pi}, \cap)$						?						?
		$\mathcal{N}(\mathrm{di}, ^{-1}, \pi, \overline{\pi}, \cap, -)$												

Conclusion

We fully established relationships between the local fragments.

Open problem

Let $\{\mathrm{di}, \overline{\pi}, \cap\} \subseteq \mathcal{F} \subseteq \{\mathrm{di}, {}^{-1}, \pi, \overline{\pi}, \cap\}$ and let $z \in \{\mathrm{bool}, \mathrm{path}\}$.
Do we have a collapse $\mathcal{N}(\mathcal{F} \cup \{-\}) \leq_z \mathcal{N}(\mathcal{F})$ or not?

Conclusion

We fully established relationships between the local fragments.

Open problem

Let $\{\mathrm{di}, \overline{\pi}, \cap\} \subseteq \mathcal{F} \subseteq \{\mathrm{di}, {}^{-1}, \pi, \overline{\pi}, \cap\}$ and let $z \in \{\mathrm{bool}, \mathrm{path}\}$.
Do we have a collapse $\mathcal{N}(\mathcal{F} \cup \{-\}) \preceq_z \mathcal{N}(\mathcal{F})$ or not?

Lemma

On unlabeled chains, we have $\mathcal{N}(\mathrm{di}, {}^{-1}, \pi, \overline{\pi}, \cap, -) \preceq_{\mathrm{bool}} \mathcal{N}(\overline{\pi})$.