HUMBOLDT-UNIVERSITÄT ZU BERLIN

Learning Aggregate Queries Defined by First-Order Logic with Counting

Steffen van Bergerem and Nicole Schweikardt

Highlights 2024

How to be a Good Colleague

How to be a Good Colleague

Name	Popularity
Alice	5
Bob	1
Carol	2
Dan	3
Emma	1

(names changed for privacy reasons)

How to be a Good Colleague

Name	Popularity
Alice	5
Bob	1
Carol	2
Dan	3
Emma	1

(names changed for privacy reasons)

Name	Type of Cake
Alice	chocolate
Dan	lemon
Carol	strawberry
Alice	chocolate
Bob	carrot
Emma	apple
Dan	chocolate
Alice	strawberry
Carol	lemon

How to be a Good Colleague

Name	Popularity
Alice	5
Bob	1
Carol	2
Dan	3
Emma	1

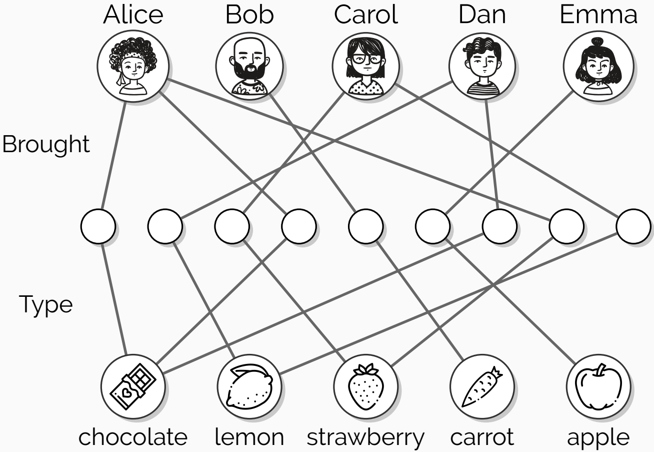
(names changed for privacy reasons)

Name	Type of Cake
Alice	chocolate
Dan	lemon
Carol	strawberry
Alice	chocolate
Bob	carrot
Emma	apple
Dan	chocolate
Alice	strawberry
Carol	lemon

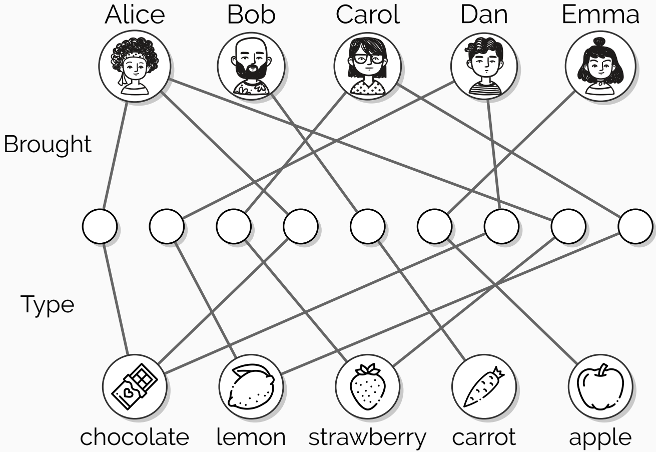
$$\text{Popularity} = 2 \cdot \#\text{chocolate cakes} + \#\text{other cakes}$$

How to be a Good Colleague

How to be a Good Colleague

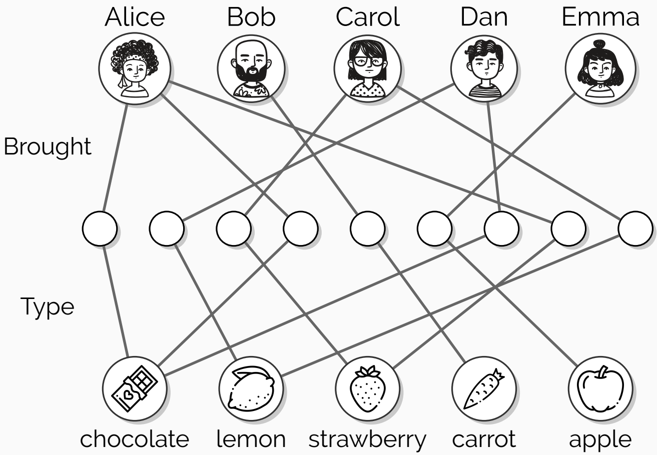


How to be a Good Colleague



(Alice, 5)
(Bob, 1)
(Carol, 2)
(Dan, 3)
(Emma, 1)

How to be a Good Colleague

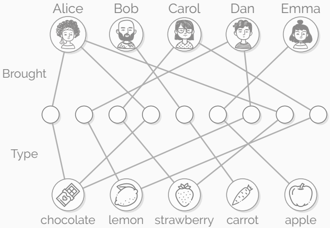


$(\text{Alice}, 5)$
$(\text{Bob}, 1)$
$(\text{Carol}, 2)$
$(\text{Dan}, 3)$
$(\text{Emma}, 1)$

$$p(x) \;=\; 2 \cdot \#(c).(\mathit{Brought}(x, c) \wedge \mathit{Type}(c, \text{🍫})) \;+\; \#(c).(\mathit{Brought}(x, c) \wedge \neg \mathit{Type}(c, \text{🍫}))$$

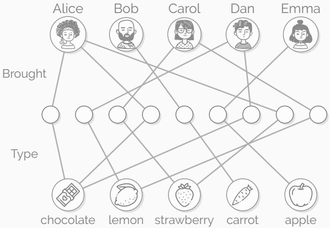
Learning from Examples

Precomputation: Given relational structure $\mathcal{A}$, build index structure



Learning from Examples

Precomputation: Given relational structure $\mathcal{A}$, build index structure



$(\text{Alice}, 5)$
$(\text{Bob}, 1)$
$(\text{Carol}, 2)$
$(\text{Dan}, 3)$
$(\text{Emma}, 1)$

Given list of labelled examples $(\bar{v}, \lambda) \in (U(\mathcal{A}))^k \times \mathbb{Z}$

Return term $t(\bar{x}) \in \text{FOC}_1$ (of certain maximum complexity)
such that $[\![t(\bar{v})]\!]^{\mathcal{A}} = \lambda$ for all given examples $(\bar{v}, \lambda)$

or reject if there is no such term

Results on structures of small degree

Grohe and Ritzert, LICS 2017

Boolean-valued concepts definable in first-order logic can be learned in sublinear time.

v. B. and Schweikardt, CSL 2021

Boolean-valued concepts definable in first-order logic with counting or first-order logic with weight aggregation can be learned in sublinear time after quasi-linear-time precomputation.

Results on structures of small degree

Grohe and Ritzert, LICS 2017

Boolean-valued concepts definable in first-order logic can be learned in sublinear time.

v. B. and Schweikardt, CSL 2021

Boolean-valued concepts definable in first-order logic with counting or first-order logic with weight aggregation can be learned in sublinear time after quasi-linear-time precomputation.

v. B. and Schweikardt, ICDT 2025

Integer-valued concepts definable in first-order logic with counting can be learned in sublinear time after quasi-linear-time precomputation.

Main tool: locality results similar to Gaifman normal forms

Results on structures of small degree

Grohe and Ritzert, LICS 2017

Boolean ... logic can be learned in s...

Bring more (chocolate) cakes!

v. B. and Schweikardt, CSL

Boolea... -order logic with counting or first-or... aggregation can be learned in sublinear time after qu... linear-time precomputation.

Come to the poster. There will be cookies. 🍪

v. B. and Schweikardt, ICDT 2025

Integer-valued concepts definable in first-order logic with counting can be learned in sublinear time after quasi-linear-time precomputation.

Main tool: locality results similar to Gaifman normal forms

Icons made by Freepik from Flaticon