Weaker Forms of Monotonicity for Declarative Networking: a more fine-grained answer to the CALM-conjecture.

Tom J Ameloot[1], Bas Ketsman[1], Frank Neven[1]
and Daniel Zinn[2]

[1] Hasselt University [2] LogicBlox, Inc

Overview

Introduction

- Declarative Networking: Datalog based languages for parallel and distributed computing
- Cloud-computing: Setting with asynchronous communication via messages which can be arbitrarily delayed but not lost
- CALM-conjecture: No coordination = Monotonicity

[Hellerstein, 2010]

(CALM = Consistency And Logical Monotonicity)

Monotonicity

Definition

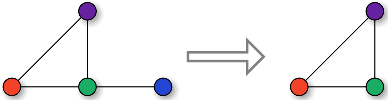
A query Q is monotone if $Q(\mathbf{I}) \subseteq Q(\mathbf{I} \cup \mathbf{J})$ for all database instances $\mathbf{I}$ and $\mathbf{J}$.

Notation

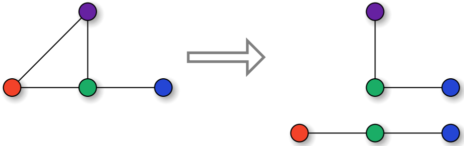
$\mathcal{M}$: *class of monotone queries*

Example

- Q_Δ: Select triangles in a graph $\in \mathcal{M}$

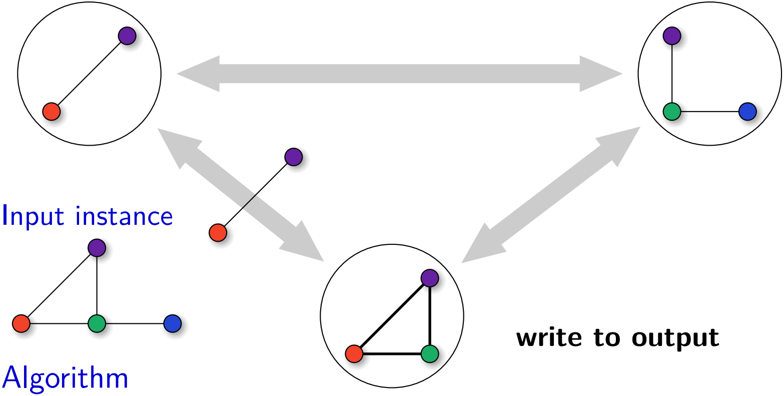


- $Q_<$: Select open triangles in a graph $\notin \mathcal{M}$



CALM by Example

Q_Δ: select all triangles $\in \mathcal{M}$



Input instance

write to output

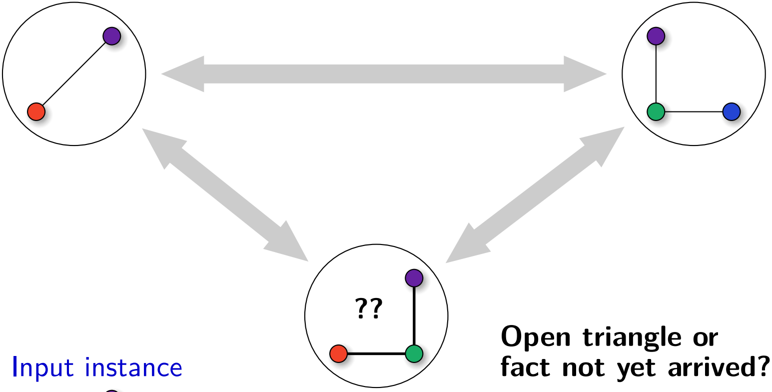
Algorithm

- broadcast all data
- periodically output local triangles

No coordination + Eventually consistent

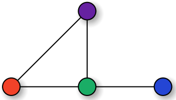
CALM by Example

$Q_<$: select all open triangles $\notin \mathcal{M}$



Open triangle or
fact not yet arrived??

Input instance



Requires global coordination

CALM-conjecture

CALM-conjecture

No-coordination = Monotonicity

[Hellerstein, 2010]

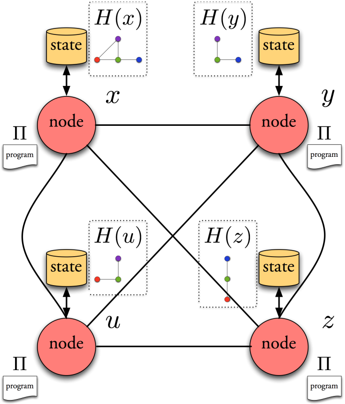
- [Ameloot, Neven, Van den Bussche, 2011]: TRUE
  - for a setting where nodes have **no** information about the distribution of facts
- [Zinn, Green, Ludäscher, 2012]: FALSE
  - for settings where nodes **have** information about the distribution of facts
- **TRUE when also refining montonicity**

Overview

Relational Transducer Networks

[Ameloot, Neven, Van den Bussche, 2011]

- Network $\mathcal{N} = \{x, y, u, z\}$
- Transducer Π
- messages can be arbitrarily delayed but never get lost



Semantics defined in terms of runs over a transition system

Relational Transducer Networks

[Ameloot, Neven, Van den Bussche, 2011]

Definition

A transducer Π computes a query Q if

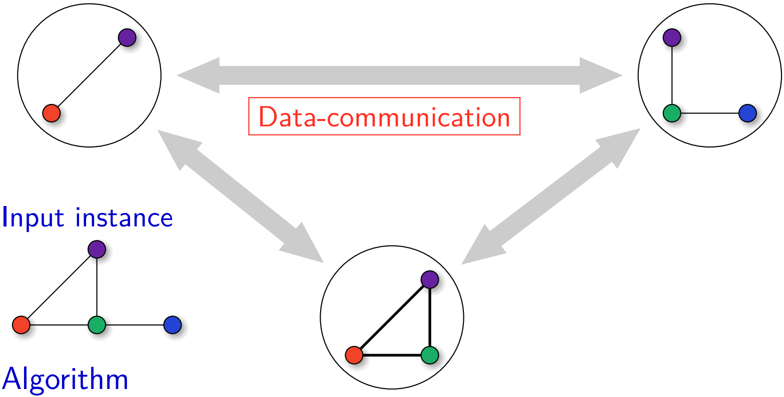
- for all networks $\mathcal{N}$, ← Network independent
- for all databases $\mathbf{I}$, Data distribution independent
- for all horizontal distributions H, and
- for every run of Π,

$$out(\Pi) = Q(\mathbf{I}).$$

Consistency requirement

Coordination-free Algorithms

Q_Δ: select all triangles



Algorithm

- broadcast all data
- output triangles whenever new data arrives

Coordination-free Algorithms

[Ameloot, Neven, Van den Bussche, 2011]

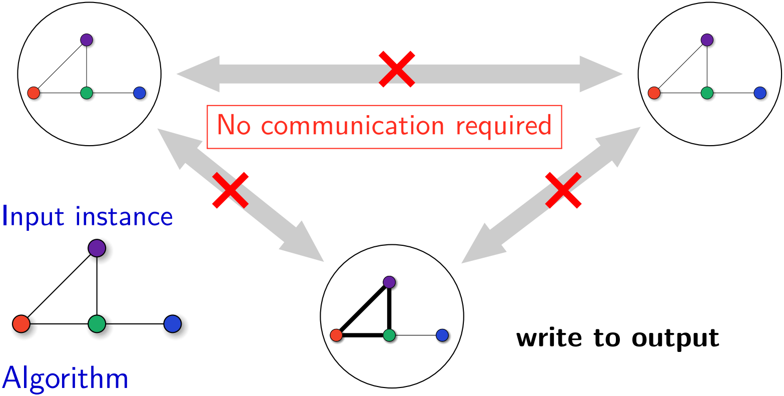
Definition

Π is coordination-free if for all inputs $\mathbf{I}$ there is a distribution on which Π computes $Q(\mathbf{I})$ without having to do communication.

Goal: separate data-communication from coordination-communication

Example: Ideal Distribution

Q_Δ: select all triangles



Algorithm

- (broadcast all data)
- periodically output local triangles

CALM-conjecture

[Ameloot, Neven, Van den Bussche, 2011]

A query has a coordination-free and eventually consistent execution strategy
iff
the query is monotone

Theorem
$\mathcal{F}_0 = \mathcal{M}$

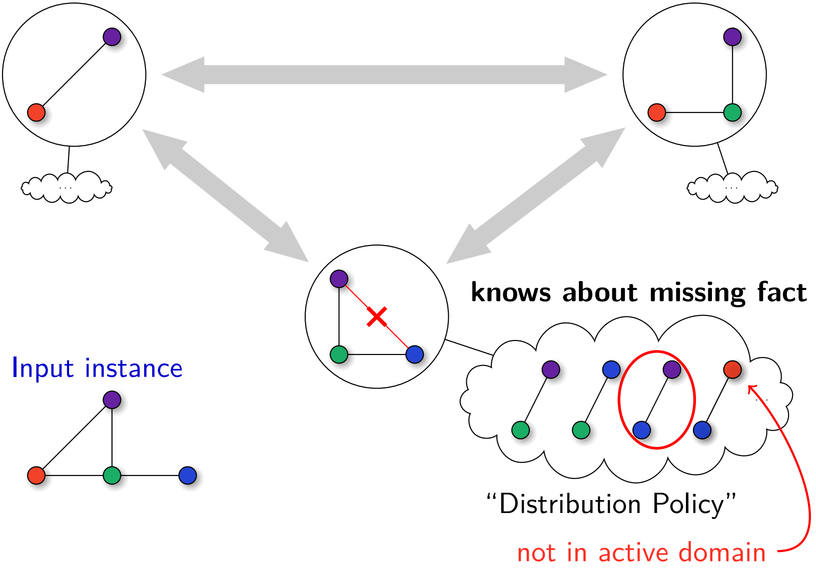
Definition
$\mathcal{F}_0$ = set of queries which are distributedly computed by coordination-free transducers

Overview

1. Introduction
2. CALM
3. CALM Revision 1
4. CALM Revision 2
5. Datalog
6. Conclusion

$$
\begin{array}{cccc}
 & \mathcal{F}_0 & = & \mathcal{M} \\
 & \rotatebox{90}{$\supsetneq$} & & \rotatebox{90}{$\supsetneq$} \\
\rightarrow & \mathcal{F}_1 & = & \mathcal{M}_{distinct} \\
 & \rotatebox{90}{$\supsetneq$} & & \rotatebox{90}{$\supsetneq$} \\
 & \mathcal{F}_2 & = & \mathcal{M}_{disjoint}
\end{array}
$$

Policy-aware Transducers



Policy-aware Transducers

[Zinn, Green, Ludäscher, 2012]

Definition

A distribution policy $\mathbf{P}$ for σ and $\mathcal{N}$ is a total function from $facts(\sigma)$ to the power set of $\mathcal{N}$.

Definition

A policy-aware transducer is a transducer with access to $\mathbf{P}$ restricted to its active domain

Definition

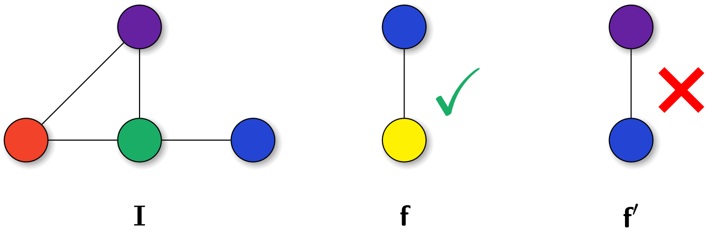
$\mathcal{F}_1$ = set of queries which are distributedly computed by policy-aware coordination-free transducers

Domain-distinct-monotonicity

Definition

A fact $\mathbf{f}$ is domain distinct from instance $\mathbf{I}$ when $adom(\mathbf{f}) \not\subseteq adom(\mathbf{I})$.

Example

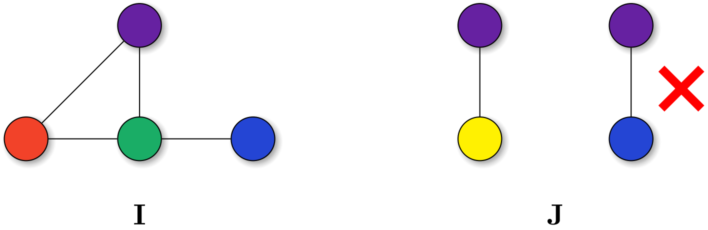


Domain-distinct-monotonicity

Definition

An instance $\mathbf{J}$ is domain distinct from instance $\mathbf{I}$ when every fact $\mathbf{f} \in \mathbf{J}$ is domain distinct from $\mathbf{I}$.

Example



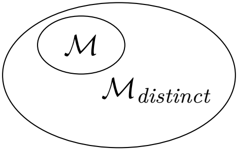
Domain-distinct-monotonicity

Definition

A query Q is domain-distinct-monotone if $Q(\mathbf{I}) \subseteq Q(\mathbf{I} \cup \mathbf{J})$ for all $\mathbf{I}$ and $\mathbf{J}$ for which $\mathbf{J}$ is domain distinct from $\mathbf{I}$.

Notation

$\mathcal{M}_{distinct}$: *class of domain-distinct-monotone queries*



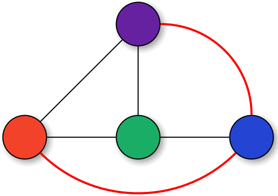
Remark

$\mathcal{M}_{distinct}$: *class of queries preserved under extensions*

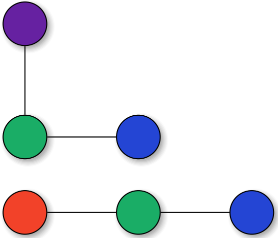
Domain-distinct-monotonicity

Example

Select open triangles in graph $\in \mathcal{M}_{distinct}$.



$\mathbf{I}$



$Q(\mathbf{I})$

Not domain-distinct from $\mathbf{I}$

Revised CALM-conjecture

A query has a coordination-free and eventually consistent execution strategy under distribution policies
iff
the query is domain-distinct-monotone

Theorem
$\mathcal{F}_1 = \mathcal{M}_{distinct}$

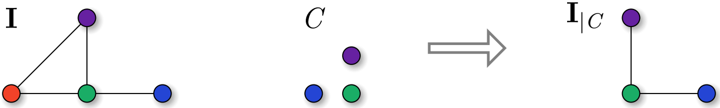
Definition
$\mathcal{F}_1$ = set of queries which are distributedly computed by policy-aware coordination-free transducers

Proof of $\mathcal{M}_{distinct} \subseteq \mathcal{F}_1$

- Monotonicity: $Q(\mathbf{J}) \subseteq Q(\mathbf{I})$ for every $\mathbf{J} \subseteq \mathbf{I}$
- Domain-distinct-monotonicity:

Let $\mathbf{I}$ be an instance, $C \subseteq adom(\mathbf{I})$.

Induced instance: $\mathbf{I}_{|C} = \{\mathbf{f} \in \mathbf{I} \mid adom(\mathbf{f}) \subseteq C\}$



By domain-distinct-monotonicity: $Q(\mathbf{I}_{|C}) \subseteq Q(\mathbf{I})$

Proof of $\mathcal{M}_{distinct} \subseteq \mathcal{F}_1$

- $\mathcal{F}_1$ setting:

  Let $\mathbf{I}$ be an instance, $C \subseteq adom(\mathbf{I})$.

  C is complete at node x when x knows for every fact **f** with $adom(\mathbf{f}) \subseteq C$ whether $\mathbf{f} \in \mathbf{I}$ or $\mathbf{f} \notin \mathbf{I}$.

  $$\begin{aligned}\text{complete set} &= \text{instance based on complete } C \\ &= \text{induced instance of } \mathbf{I} \text{ based on } C\end{aligned}$$

Algorithm

- broadcast all present and deduced absent facts
- Evaluate query on complete sets

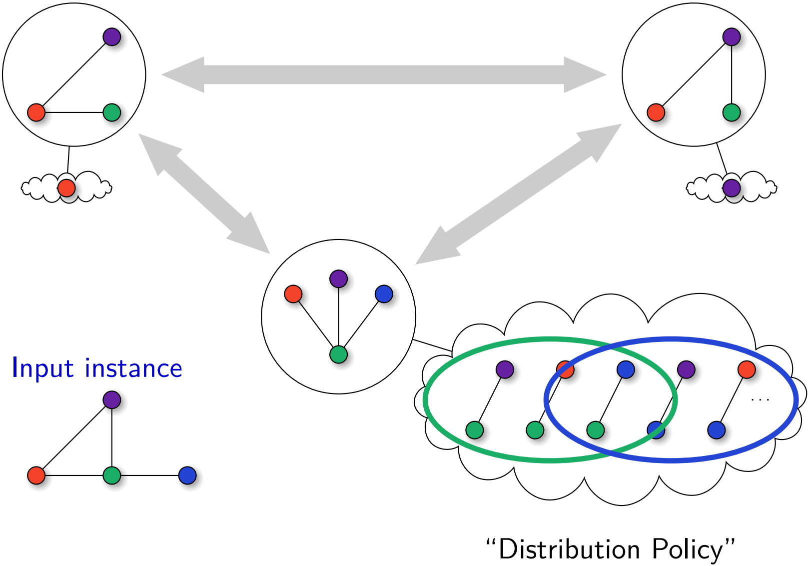
Overview

1. Introduction
2. CALM
3. CALM Revision 1
4. CALM Revision 2
5. Datalog
6. Conclusion

$$
\begin{array}{ccc}
\mathcal{F}_0 & = & \mathcal{M} \\
\rotatebox{90}{$\subsetneq$} & & \rotatebox{90}{$\subsetneq$} \\
\mathcal{F}_1 & = & \mathcal{M}_{distinct} \\
\rotatebox{90}{$\subsetneq$} & & \rotatebox{90}{$\subsetneq$} \\
\end{array}
$$

$$
\rightarrow \quad \boxed{\mathcal{F}_2 \quad = \quad \mathcal{M}_{disjoint}}
$$

Domain-guided Policies



Domain-guided Policies

[Zinn, Green, Ludäscher, 2012]

Definition

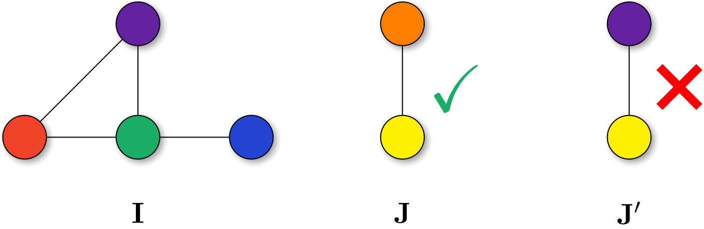
$\mathcal{F}_2 =$ queries which are distributedly computed under domain-guided distribution policies by policy-aware coordination-free transducers.

Domain-disjoint-monotonicity

Definition

An instance $\mathbf{J}$ is domain disjoint from instance $\mathbf{I}$ when $adom(\mathbf{I}) \cap adom(\mathbf{J}) = \emptyset$.

Example



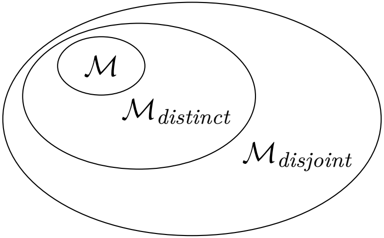
Domain-disjoint-monotonicity

Definition

A query Q is domain-disjoint-monotone if $Q(\mathbf{I}) \subseteq Q(\mathbf{I} \cup \mathbf{J})$ for all $\mathbf{I}$ and $\mathbf{J}$ for which $\mathbf{J}$ is domain disjoint from $\mathbf{I}$.

Notation

$\mathcal{M}_{disjoint}$: class of domain-disjoint-monotone queries



Revised CALM-conjecture

A query has a coordination-free and eventually consistent execution strategy under domain-guided distribution policies

iff

the query is domain-disjoint-monotone

Theorem

$$\mathcal{F}_2 = \mathcal{M}_{disjoint}$$

Definition

$\mathcal{F}_2 =$ queries which are distributedly computed under domain-guided distribution policies by policy-aware coordination-free transducers.

Intermediate Summary

$\mathcal{F}_0$	$=$	$\mathcal{M}$	$=$	wILOG($\neq$)	$\supsetneq$	Datalog($\neq$)
$\subsetneq$		$\subsetneq$		$\subsetneq$		$\subsetneq$
$\mathcal{F}_1$	$=$	$\mathcal{M}_{distinct}$	$=$	SP-wILOG	$\supsetneq$	SP-Datalog
$\subsetneq$		$\subsetneq$		$\subsetneq$		$\subsetneq$
$\mathcal{F}_2$	$=$	$\mathcal{M}_{disjoint}$	$=$	semicon-wILOG$^{\neg}$	$\supsetneq$	semicon-Datalog$^{\neg}$
Coordination freeness		Monotonicity		Datalog + value invention		Datalog

Datalog Variants

Datalog($\neq$) $\subsetneq$ wILOG($\neq$) $= \mathcal{M}$

- Datalog($\neq$) $\subsetneq \mathcal{M} \cap \mathrm{PTIME}$

  [Afrati, Cosmadakis, Yannakakis, 1994]

- wILOG($\neq$) $= \mathcal{M}$

  [Cabibbo,1998]

SP-Datalog $\subsetneq$ SP-wILOG $= \mathcal{M}_{distinct}$

- SP-Datalog $\subsetneq \mathcal{M}_{distinct} \cap \mathrm{PTIME}$

  [Afrati, Cosmadakis, Yannakakis, 1994]

- SP-wILOG $= \mathcal{M}_{distinct}$

  [Cabibbo,1998]

Datalog variant of $\mathcal{M}_{disjoint}$?

semicon-Datalog$^\neg$

semicon-Datalog$^\neg$ $\subsetneq$ semicon-wILOG$^\neg$ $= \mathcal{M}_{disjoint}$

Connected Rules

$O(x, y, z) \leftarrow E(x, y), E(y, z), E(z, x)$ is connected

$O(x, y, z) \leftarrow E(x, y), E(z, z)$ is not connected

Definition

A stratified-Datalog program is semi-connected if all rules are connected except (possibly) those of the last stratum.

Example

Complement of transitive closure:

$$
\begin{aligned}
TC(x, y) &\leftarrow E(x, y) \\
TC(x, y) &\leftarrow E(x, z), TC(z, y) \\
O(x, y) &\leftarrow \neg TC(x, y), x \neq y
\end{aligned}
$$

Conclusion and Future Work

Conclusion

- Coordination-free evaluation = (refined) monotonicity
- (semi-)connected Datalog

Can we put the CALM-conjecture to rest?

Future Work

- Other settings / other distribution policies?
- Coordination-free + efficient evaluation?