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

Tom J Ameloot\textsuperscript{1}, Bas Ketsman\textsuperscript{1}, Frank Neven\textsuperscript{1} and Daniel Zinn\textsuperscript{2}

\textsuperscript{1} Hasselt University \textsuperscript{2} LogicBlox, Inc
Overview

1. Introduction

2. CALM

3. CALM Revision 1

4. CALM Revision 2

5. Datalog

6. Conclusion
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

[Calif, 2010]

(CALT = Consistency And Logical Monotonicity)
Monotonicity

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

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

Example
- $Q_{\Delta}$: Select triangles in a graph $\in \mathcal{M}$

- $Q_{<}$: Select open triangles in a graph $\not\in \mathcal{M}$
CALM by Example

$Q_\Delta$: select all triangles $\in M$

Algorithm
- broadcast all data
- periodically output local triangles

No coordination + Eventually consistent
CALM by Example

$Q_\lambda$: select all open triangles $\not\in M$

Input instance

Open triangle or fact not yet arrived??

Requires global coordination
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

1. Introduction

2. CALM

3. CALM Revision 1

4. CALM Revision 2

5. Datalog

6. Conclusion
Relational Transducer Networks

[Ameloot, Neven, Van den Bussche, 2011]

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

Semantics defined in terms of runs over a transition system
Definition

A transducer $\Pi$ computes a query $Q$ if
- for all networks $N$,
- for all databases $I$,
- for all horizontal distributions $H$, and
- for every run of $\Pi$,

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

Consistency requirement
Coordination-free Algorithms

$Q_\Delta$: select all triangles

Input instance

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 $\Pi$ computes $Q(\mathbf{I})$ without having to do communication.

**Goal:** separate data-communication from coordination-communication
Example: Ideal Distribution

$Q_\Delta$: select all triangles

No communication required

Input instance

Algorithm

- (broadcast all data)
- periodically output local triangles
A query has a coordination-free and eventually consistent execution strategy iff the query is monotone.

**Theorem**
\[ \mathcal{F}_0 = \mathcal{M} \]

**Definition**
\[ \mathcal{F}_0 = \text{set of queries which are distributedly computed by coordination-free transducers} \]
Overview

1. Introduction

2. CALM

3. CALM Revision 1

4. CALM Revision 2

5. Datalog

6. Conclusion

\[ \mathcal{F}_0 = \mathcal{M} \]

\[ \downarrow \quad \downarrow \]

\[ \mathcal{F}_1 = \mathcal{M}_{\text{distinct}} \]

\[ \downarrow \quad \downarrow \]

\[ \mathcal{F}_2 = \mathcal{M}_{\text{disjoint}} \]
Policy-aware Transducers

knows about missing fact

"Distribution Policy"
not in active domain
Policy-aware Transducers

[Zinn, Green, Ludäscher, 2012]

Definition

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

Definition

A policy-aware transducer is a transducer with access to $\mathcal{P}$ 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 \( f \) is **domain distinct** from instance \( I \) when
\[
\text{adom}(f) \not\subseteq \text{adom}(I).
\]

Example

\[ I \quad \quad \quad \quad \quad f \quad \quad \quad \quad f' \]
Domain-distinct-monotonicity

**Definition**

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

**Example**

![Diagram showing the domain distinctness between instances $I$ and $J$.]
Domain-distinct-monotonicity

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

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

Remark
$\mathcal{M}_{\text{distinct}}$: class of queries preserved under extensions
Domain-distinct-monotonicity

Example
Select open triangles in graph $\in M_{\text{distinct}}$.

Not domain-distinct from $I$
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
\[ \mathcal{F}_1 = \mathcal{M}_{\text{distinct}} \]

Definition
\[ \mathcal{F}_1 = \text{set of queries which are distributedly computed by policy-aware coordination-free transducers} \]
Proof of $\mathcal{M}_{\text{distinct}} \subseteq \mathcal{F}_1$

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

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

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

By domain-distinct-monotonicity: $Q(I |_C) \subseteq Q(I)$
Proof of $\mathcal{M}_{distinct} \subseteq \mathcal{F}_1$

- $\mathcal{F}_1$ setting:

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

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

complete set = instance based on complete $C$
= induced instance of $\mathbf{I}$ based on $C$

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

\[ \mathcal{F}_0 = \mathcal{M} \]
\[ \mathcal{F}_1 = \mathcal{M}_{\text{distinct}} \]
\[ \mathcal{F}_2 = \mathcal{M}_{\text{disjoint}} \]
Domain-guided Policies

Input instance

“Distribution Policy”
Domain-guided Policies

[Zinn, Green, Ludäscher, 2012]

Definition

\[ F_2 = \text{queries which are distributedly computed under domain-guided distribution policies by policy-aware coordination-free transducers.} \]
Domain-disjoint-monotonicity

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

Example

$I$

$J$

$J'$
Domain-disjoint-monotonicity

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

Notation
\( \mathcal{M}_{\text{disjoint}} \): 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

\[
\mathcal{F}_2 = \mathcal{M}_{\text{disjoint}}
\]

Definition

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

<table>
<thead>
<tr>
<th>$\mathcal{F}_0$</th>
<th>$=\mathcal{M}$</th>
<th>$\subseteq\mathcal{wILOG}(\neq)$</th>
<th>$\cap$</th>
<th>$\subseteq\mathcal{Datalog}(\neq)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\cap$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathcal{F}_1$</td>
<td>$=\mathcal{M}_{distinct}$</td>
<td>$\subseteq\mathcal{SP-wILOG}$</td>
<td>$\cap$</td>
<td>$\subseteq\mathcal{SP-Datalog}$</td>
</tr>
<tr>
<td></td>
<td>$\cap$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mathcal{F}_2$</td>
<td>$=\mathcal{M}_{disjoint}$</td>
<td>$\subseteq\mathcal{semicon-wILOG}$</td>
<td>$\cap$</td>
<td>$\subseteq\mathcal{semicon-Datalog}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordination</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>freeness</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monotonicity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datalog +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>value invention</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Datalog</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Datalog Variants

\[
\text{Datalog}(\neq) \subsetneq \text{wILOG}(\neq) = \mathcal{M}
\]

- \[
\text{Datalog}(\neq) \subsetneq \mathcal{M} \cap \text{PTIME}
\]
  [Afrati, Cosmadakis, Yannakakis, 1994]

- \[
\text{wILOG}(\neq) = \mathcal{M}
\]
  [Cabibbo, 1998]

\[
\text{SP-Datalog} \subsetneq \text{SP-wILOG} = \mathcal{M}_{\text{distinct}}
\]

- \[
\text{SP-Datalog} \subsetneq \mathcal{M}_{\text{distinct}} \cap \text{PTIME}
\]
  [Afrati, Cosmadakis, Yannakakis, 1994]

- \[
\text{SP-wILOG} = \mathcal{M}_{\text{distinct}}
\]
  [Cabibbo, 1998]

Datalog variant of \[\mathcal{M}_{\text{disjoint}}\]?
semicon-Datalog

semicon-Datalog \subseteq \neg \text{semicon-wILOG} = \mathcal{M}_{disjoint}

Connected Rules

\begin{align*}
O(x, y, z) & \leftarrow E(x, y), E(y, z), E(z, x) \text{ is connected} \\
O(x, y, z) & \leftarrow E(x, y), E(z, z) \text{ is not connected}
\end{align*}

Definition

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

Example

Complement of transitive closure:

\begin{align*}
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{align*}
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?