Local First-Order Logic with Two Data Values

Benedikt Bollig, Arnaud Sangnier, Olivier Stietel

FSTTCS 2021

December 15–17, 2021

Introduction

Context

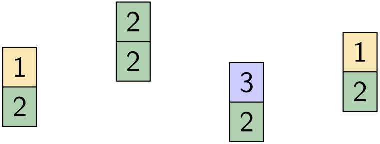
- Data-aware systems are omnipresent
  - Database
  - Sets of data for learning
  - Distributed/ Concurrent Systems
- Need for specification languages to describe systems with data

Motivation

- Logic to secify input-output behavior of distributed algorithms
- Structures with two data values
- The input values can be compared with the output values

Consensus problem

- Famous problem in distributed computing
- The goal is to design an algorithm such that:
  - All entities in a network have an input value
  - They should all agree on the same value
  - The chosen value should be one of the input values



First-order logic with two data values

Structures

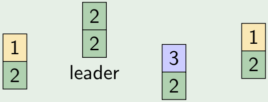
- Σ finite set of unary relation symbols

Definition

A structure is a tuple $\mathfrak{A} = (A, f_1, f_2, (P_\sigma)_{\sigma \in \Sigma})$ where:

- A is the nonempty finite universe
- $f_1, f_2 : A \to \mathbb{N}$ map each element to two *data values*
- $P_\sigma \subseteq A$ for all $\sigma \in \Sigma$

Example



Logic with two data values

- Σ finite set of unary relation symbols
- $\Gamma \subseteq \{1,2\} \times \{1,2\}$ set of binary relation symbols

Definition

The logic $\mathrm{FO}[\Sigma; \Gamma]$ is given as follows:

$$\varphi ::= \sigma(x) \mid x \;{}_i{\sim}_j\; y \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi$$

where $\sigma \in \Sigma$ and $(i,j) \in \Gamma$

Logic with two data values

- Σ finite set of unary relation symbols
- $\Gamma \subseteq \{1,2\} \times \{1,2\}$ set of binary relation symbols

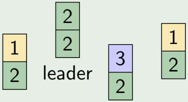
Definition

The logic FO[Σ; Γ] is given as follows:

$$\varphi ::= \sigma(x) \mid x \;{}_i{\sim}_j\; y \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi$$

where $\sigma \in \Sigma$ and $(i,j) \in \Gamma$

Example



$\models \exists^{=1} x.\mathrm{leader}(x) \wedge \forall y.\exists x.(\mathrm{leader}(x) \wedge x \;{}_1{\sim}_2\; y)$

Logic with two data values

- Σ finite set of unary relation symbols
- $\Gamma \subseteq \{1,2\} \times \{1,2\}$ set of binary relation symbols

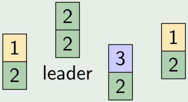
Definition

The logic $FO[\Sigma;\Gamma]$ is given as follows:

$$\varphi ::= \sigma(x) \mid x\ {}_i{\sim}_j\ y \mid x = y \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x.\varphi$$

where $\sigma \in \Sigma$ and $(i,j) \in \Gamma$

Example



$\models \exists^{=1} x.\text{leader}(x) \wedge \forall y.\exists x.(\text{leader}(x) \wedge x\ {}_1{\sim}_2\ y)$

$FO^2[\Sigma;\Gamma]$ is the two-variable fragment.

Known facts

Definition

The problem $\mathrm{DataSat}(\mathcal{F}, \Gamma)$ is defined as follows:

Input: Finite set Σ; sentence $\varphi \in \mathcal{F}[\Sigma; \Gamma]$.

Question: Is there a data structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi$?

Known facts

Definition

The problem DataSat($\mathcal{F}, \Gamma$) is defined as follows:

Input: Finite set Σ; sentence $\varphi \in \mathcal{F}[\Sigma; \Gamma]$.

Question: Is there a data structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi$?

Theorem (Janiczak, 1953)

DataSat(FO, $\{(1,1),(2,2)\}$) is undecidable, even when $\Sigma = \emptyset$.

Known facts

Definition

The problem DataSat($\mathcal{F}, \Gamma$) is defined as follows:

Input: Finite set Σ; sentence $\varphi \in \mathcal{F}[\Sigma; \Gamma]$.

Question: Is there a data structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi$?

Theorem (Janiczak, 1953)

DataSat(FO, $\{(1,1),(2,2)\}$) is undecidable, even when $\Sigma = \emptyset$.

Theorem (Kieronski and Tendera, 2009)

DataSat($\mathrm{FO}^2, \{(1,1),(2,2)\}$) is decidable.

Known facts

Definition

The problem DataSat($\mathcal{F}, \Gamma$) is defined as follows:

Input: Finite set Σ; sentence $\varphi \in \mathcal{F}[\Sigma; \Gamma]$.

Question: Is there a data structure $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi$?

Theorem (Janiczak, 1953)

DataSat(FO, $\{(1,1),(2,2)\}$) is undecidable, even when $\Sigma = \emptyset$.

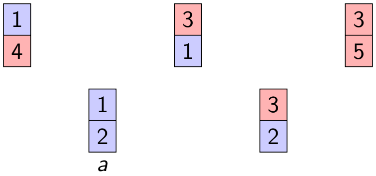
Theorem (Kieronski and Tendera, 2009)

DataSat(FO^2, $\{(1,1),(2,2)\}$) is decidable.

Other related work: Two-variable logic on data words [Bojanczyk, David, Muscholl, Schwentick, and Segoufin, 2011]

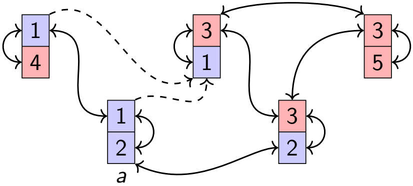
Local first-order logic

Local logics reason about neighborhoods



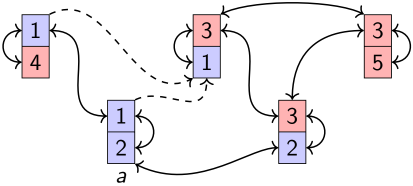
- $\mathfrak{A}$ — data structure

Local logics reason about neighborhoods



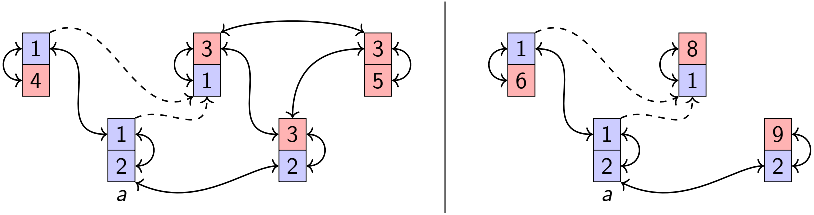
- $\mathfrak{A}$ — data structure
- $\mathcal{G}(\mathfrak{A})$ — data graph when $\Gamma = \{(1,1),(2,2),(1,2)\}$

Local logics reason about neighborhoods



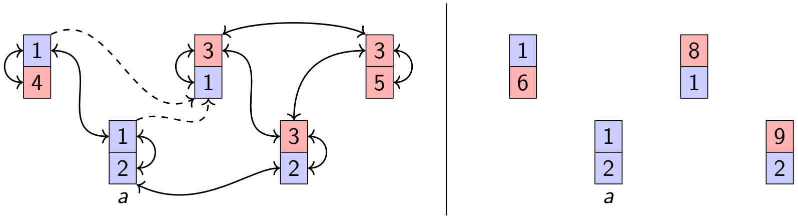
- $\mathfrak{A}$ — data structure
- $\mathcal{G}(\mathfrak{A})$ — data graph when $\Gamma = \{(1,1),(2,2),(1,2)\}$
- $B_1^{\mathfrak{A}}(a)$ — 1-ball (blue nodes)

Local logics reason about neighborhoods



- $\mathfrak{A}$ — data structure
- $\mathcal{G}(\mathfrak{A})$ — data graph when $\Gamma = \{(1,1),(2,2),(1,2)\}$
- $B_1^{\mathfrak{A}}(a)$ — 1-ball (blue nodes)
- $\mathfrak{A}|_a^1$ — 1-neighborhood of a

Local logics reason about neighborhoods



- $\mathfrak{A}$ — data structure
- $\mathcal{G}(\mathfrak{A})$ — data graph when $\Gamma = \{(1,1),(2,2),(1,2)\}$
- $B_1^{\mathfrak{A}}(a)$ — 1-ball (blue nodes)
- $\mathfrak{A}|_a^1$ — 1-neighborhood of a
- local formula $\langle\langle\psi\rangle\rangle_x^1$ with $\psi \in \mathrm{FO}[\Sigma;\Gamma]$ can reason about $\mathfrak{A}|_a^1$

Local logic with two data values

- Σ finite set of unary relation symbols
- $\Gamma \subseteq \{1,2\} \times \{1,2\}$ set of binary relation symbols

Definition

For a radius $r \in \mathbb{N}$, the logic r-Loc-FO[Σ; Γ] is given as follows:

$$\varphi \ ::= \ \langle\!\langle \psi \rangle\!\rangle_x^r \ | \ x = y \ | \ \exists x.\varphi \ | \ \varphi \vee \varphi \ | \ \neg\varphi$$

where $\sigma \in \Sigma$ and $(i,j) \in \Gamma$

Local logic with two data values

- Σ finite set of unary relation symbols
- $\Gamma \subseteq \{1,2\} \times \{1,2\}$ set of binary relation symbols

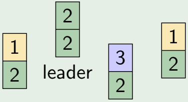
Definition

For a radius $r \in \mathbb{N}$, the logic r-Loc-FO[$\Sigma; \Gamma$] is given as follows:

$$\varphi \ ::= \ \langle\!\langle \psi \rangle\!\rangle_x^r \mid x = y \mid \exists x.\varphi \mid \varphi \vee \varphi \mid \neg\varphi$$

where $\sigma \in \Sigma$ and $(i,j) \in \Gamma$

Example



$$\models \quad \exists^{=1} x. \langle\!\langle \text{leader}(x) \rangle\!\rangle_x^1 \ \wedge$$
$$\forall y. \langle\!\langle \exists x.\text{leader}(x) \wedge y \ {}_2{\sim}_1 \ x \rangle\!\rangle_y^1$$

Local logic with two data values

- Σ finite set of unary relation symbols
- $\Gamma \subseteq \{1,2\} \times \{1,2\}$ set of binary relation symbols

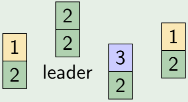
Definition

For a radius $r \in \mathbb{N}$, the logic r-Loc-FO$[\Sigma;\Gamma]$ is given as follows:

$$\varphi \ ::= \ \langle\!\langle \psi \rangle\!\rangle_x^r \ | \ x = y \ | \ \exists x.\varphi \ | \ \varphi \vee \varphi \ | \ \neg\varphi$$

where $\sigma \in \Sigma$ and $(i,j) \in \Gamma$

Example



$$\models \quad \exists^{=1} x. \langle\!\langle \mathrm{leader}(x) \rangle\!\rangle_x^1 \ \wedge$$
$$\forall y. \langle\!\langle \exists x. \mathrm{leader}(x) \wedge y \ {}_2{\sim}_1 \ x \rangle\!\rangle_y^1$$
$$\in 1\text{-Loc-FO}[\{\mathrm{leader}\}; (1,1), (2,2), (2,1)]$$

Decidability result

Main result

Theorem

DataSat(1-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) *is decidable.*

Main result

Theorem

$\mathsf{DataSat}(\text{1-Loc-FO}, \{(1,1),(2,2),(1,2)\})$ *is decidable.*

Proof.

Reduction to $\mathsf{DataSat}(\mathsf{FO}^2, \{(1,1),(2,2)\})$:

Main result

Theorem

DataSat(1-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) *is decidable.*

Proof.

Reduction to DataSat(FO^2, $\{(1,1),(2,2)\}$):

Step 1: Transform binary into unary relations

Main result

Theorem

$\mathrm{DataSat}(\text{1-Loc-FO}, \{(1,1),(2,2),(1,2)\})$ *is decidable.*

Proof.

Reduction to $\mathrm{DataSat}(\mathrm{FO}^2, \{(1,1),(2,2)\})$:

Step 1: Transform binary into unary relations

Step 2: Well-diagonalized structures

Main result

Theorem

DataSat(1-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) *is decidable.*

Proof.

Reduction to DataSat(FO^2, $\{(1,1),(2,2)\}$):

- Step 1: Transform binary into unary relations
- Step 2: Well-diagonalized structures
- Step 3: Getting rid of the diagonal relation

Main result

Theorem

DataSat(1-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) *is decidable.*

Proof.

Reduction to DataSat(FO^2, $\{(1,1),(2,2)\}$):

- Step 1: Transform binary into unary relations
- Step 2: Well-diagonalized structures
- Step 3: Getting rid of the diagonal relation
- Step 4: Counting in two-variable logic

Main result

Theorem

DataSat(1-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) *is decidable.*

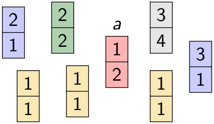
Proof.

Reduction to DataSat(FO^2, $\{(1,1),(2,2)\}$):

- Step 1: Transform binary into unary relations
- Step 2: Well-diagonalized structures
- Step 3: Getting rid of the diagonal relation
- Step 4: Counting in two-variable logic
- Step 5: Putting it All Together

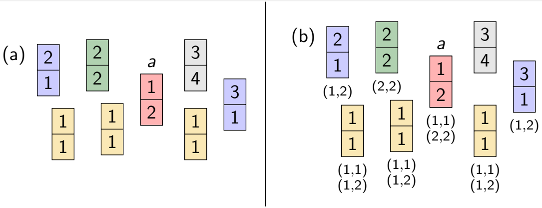
□

Step 1: Adding new unary predicates

(a) 

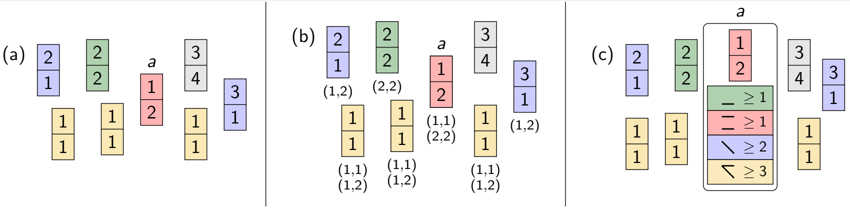
(a) data structure

Step 1: Adding new unary predicates



(a) data structure
(b) adding unary predicates for *a*

Step 1: Adding new unary predicates

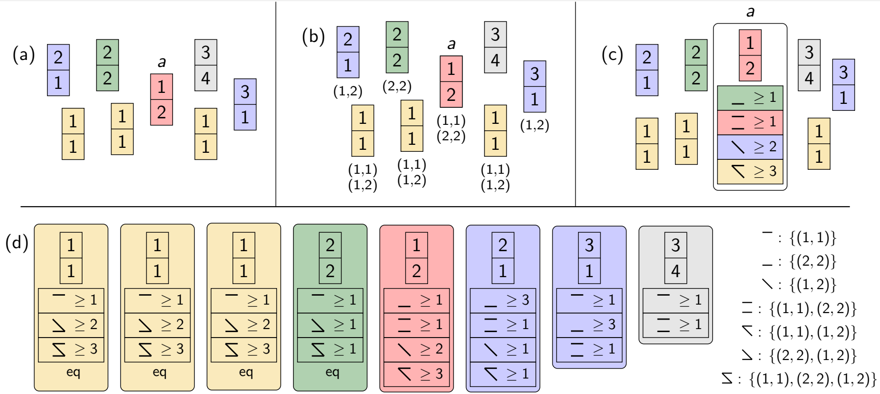


(a) data structure

(b) adding unary predicates for *a*

(c) adding counting constraints to *a*

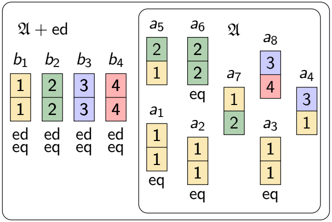
Step 1: Adding new unary predicates



- (a) data structure
- (b) adding unary predicates for *a*
- (c) adding counting constraints to *a*
- (d) well-typed data structure

Step 2: Adding diagonal elements

(a)

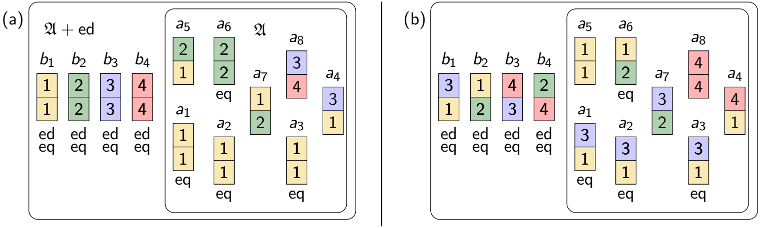


(a) Adding diagonal elements labeled with ed
- For each data, there is one element labeled with ed having the data in its input [resp. output] field

(b) Adding predicates eq for the elements with the same value

(c) Verify that the structure is correctly labeled without using diagonal relation
- Making data structure eq-respecting: (a)←(b)

Step 2: Adding diagonal elements



(a) Adding diagonal elements labeled with ed

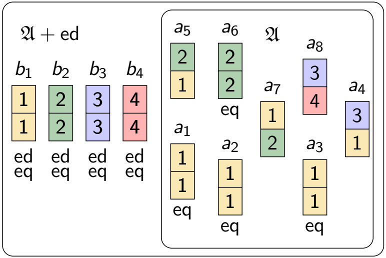
- For each data, there is one element labeled with ed having the data in its input [resp. output] field

(b) Adding predicates eq for the elements with the same value

(c) Verify that the structure is correctly labeled without using diagonal relation

- Making data structure eq-respecting: (a)⟵(b)

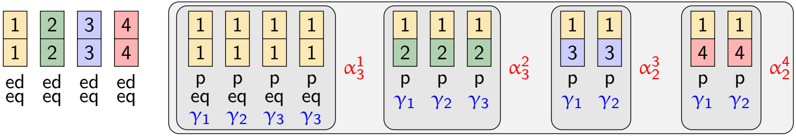
Step 3: Getting rid of the diagonal relation



⇝ checking counting constraints in extended data structure without using diagonal relation

- Use diagonal elements to remove diagonal relations
- To check whether there exist two elements with the same output value as the input value of $a1$
- Check whether for the diagonal element $b1$, there exist two elements with the same output value as $b1$

Step 4: Counting intersections



Counting intersections for $M = 3$ and elements with label p

- Intersection: elements with the same input and output values
- Trick:
  - Add unary predicates to count inside intersections
  - Add unary predicates to count intersections
  - Count up to a certain bound
- Use these predicates to verify that counting constraints labeling is correct

Step 5: Putting it all together

Given a formula $\varphi \in$ 1-Loc-FO$[\Sigma; \{(1,1),(2,2),(1,2)\}$:

(a) Build a formula $\varphi_{CC} \in \mathrm{FO}[\Sigma \cup CC; \emptyset]$ (where CC are counting constraints)

(b) Build a formula $\varphi_{ed} \in \mathrm{FO}^2[-; \{(1,1),(2,2)\}]$ to ensure the presence of diagonal elements;

(c) Build a formula $\varphi_{wt} \in \mathrm{FO}^2[\Sigma \cup CC; \{(1,1),(2,2)\}]$ to ensure that the counting constraints are correctly placed

(d) Check the satisfiability of $\varphi_{cc} \wedge \varphi_{ed} \wedge \varphi_{wt}$

(e) Redce to a satisfiability problem for $\mathrm{FO}^2[-; \{(1,1),(2,2)\}]$

Undecidability

Beyond $r = 1$: Undecidability

Theorem

The following problems are undecidable:

- $\mathsf{DataSat}(\text{3-Loc-FO}, \{(1,1),(2,2)\})$
- $\mathsf{DataSat}(\text{2-Loc-FO}, \{(1,1),(2,2),(1,2)\})$

Beyond r = 1: Undecidability

Theorem

The following problems are undecidable:

- $\mathrm{DataSat}(3\text{-Loc-FO}, \{(1,1),(2,2)\})$
- $\mathrm{DataSat}(2\text{-Loc-FO}, \{(1,1),(2,2),(1,2)\})$

Proof.

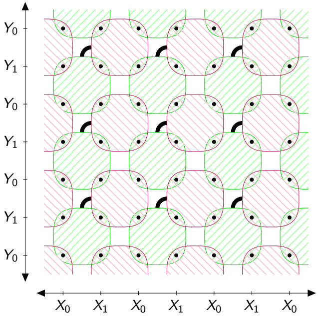
Reduction from Periodic Tiling Problem:

- Inputs: D a finite set of dominos, an horizontal relation $H \subseteq D \times D$ and a vertical relation $V \subseteq D \times D$;
- Output: Is there a periodic tiling of a plan with dominos respecting H and V ?

□

Main idea

- Encode the plan using data and equivalence relation
- Example for DataSat(2-Loc-FO, $\{(1,1),(2,2),(1,2)\}$)



Green zone : same input data, Red zone : same output data,
Black line : diagonal relation

Conclusion and Future Works

What we have seen:

- DataSat(1-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) is undecidable
- DataSat(3-Loc-FO, $\{(1,1),(2,2)\}$) and DataSat(2-Loc-FO, $\{(1,1),(2,2),(1,2)\}$) are undecidable

Perspectives:

- Complete the picture:
  - Consider both diagonal relations
  - Consider elements outside the ball
- Study other fragments with quantifiers restrictions
- Develop approximation methods

Thank you!