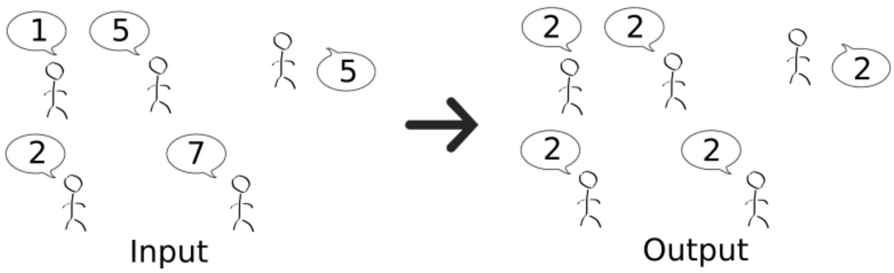
Local first order logic for distributed algorithms

Olivier Stietel[1]
IRIF, Université de Paris
LSV, ENS Paris Saclay
stietel@irif.fr

1 Introduction

Our goal is to provide a specification language for distributed algorithms designed for an unbounded number of participating processes which manipulate data. Hence consider an extension of first-order logic named IO-FO in which we can specify input and output of distributed algorithms. In the setting of distributed algorithms we have a unorganized cloud of computing units. So the models of IO-FO are algebraic structure consisting of a universe and functions assigning to each element a letter and two integers. A letter represents a state of the computing unit like "running", "finished" or "crashed". The two integers represent two data values of the element. The first one representing an input value whereas the second one is used for output value. We remark that for many distributed algorithms the precise value of the input/output values does not really matter, but what is important is the relation between these values. For example if the data values represent identifiers the only interesting thing is to know if two identifiers are equal or not. So the logic IO-FO can not directly specify the value of the integers but can only compare if two integers are equal. Our logic IO-FO[Σ] is powerful enough to specify consensus. The consensus problem can be stated as follows: we have $n > 0$ processes with an initial value and the algorithm should ensure that at the end they all suggest the same data value chosen among the initial ones. Here is an illustration for five processes:



In the literature, logics to describe structures equipped with data have already been considered, in particular in the field of database model. For instance in [1] the authors propose an extension of first order logic equipped with a total order and an equivalence relation to describe data words, an extension of words where at each position there is a letter and a data. For this latter logic, the satisfiability problem is undecidable but decidability can be regained by restricting to two the number of used variables. In [4], it is shown that the satisfiability problem for two-variables formulae of first logic with two equivalence relations (i.e. elements of the models have two data) and no linear order is decidable and in 3NEXPTIME (in [3] it is shown that this problem is 2NEXPTIME-hard). In this work, instead of limiting the number of variables in the formulae, we choose a different approach: we prevent the logic to speak about the neighborhood's neighborhood of an element. We call the obtained logic Loc-IO-FO[Σ]. When comparing the expressiveness, two-variable first-order logic can be embedded in our logic but it is not known yet whether the converse

[1] This is joint work with Ph.D.'s advisors Benedikt Bollig (LSV) and Arnaud Sangnier (IRIF).

holds. Until now our work has focused on the satisfiability problem. A next step would be to see how our logic can be used to verify in practice some distributed algorithms.

2 Local Input-Output first order logic

We fix a nonempty finite alphabet Σ and a countable infinite set of variables Var.

The set IO-FO[Σ] of first-order formulas is given as follows:

$$\varphi \ ::= \ a(x) \ | \ x = y \ | \ \varphi \vee \varphi \ | \ \neg\varphi \ | \ \exists x.\varphi \ | \ x \ {}_{\mathrm{I}}{\sim}_{\mathrm{I}} \ y \ | \ x \ {}_{\mathrm{I}}{\sim}_{\mathrm{O}} \ y \ | \ x \ {}_{\mathrm{O}}{\sim}_{\mathrm{I}} \ y \ | \ x \ {}_{\mathrm{O}}{\sim}_{\mathrm{O}} \ y$$

where $a \in \Sigma$ and $x, y, z \in \mathrm{Var}$.

Models will be structures equipped with a set of elements labeled with one letter and two data values. To put it formally a model is a tuple $\mathfrak{A} = (A, \ell, I, O)$ where A is a set and $\ell : A \to \Sigma$ and $I, O : A \to \mathbb{N}$ are three functions. We say that $\mathfrak{A}$ is finite if A is finite. A context is a function from the set of variables Var to the carrier of a model. Let $\mathfrak{A} = (A, \ell, I, O)$ be a model, φ a IO-FO[Σ] formula and Γ be a context. We define the relation $\mathfrak{A}, \Gamma \models \varphi$ inductively on φ by:

$\mathfrak{A}, \Gamma \models a(x)$ if $\ell(\Gamma(x)) = a$

$\mathfrak{A}, \Gamma \models x = y$ if $\Gamma(x) = \Gamma(y)$

$\mathfrak{A}, \Gamma \models z \ {}_{\mathrm{I}}{\sim}_{\mathrm{I}} \ x$ if $I(\Gamma(z)) = I(\Gamma(x))$

$\mathfrak{A}, \Gamma \models z \ {}_{\mathrm{I}}{\sim}_{\mathrm{O}} \ x$ if $I(\Gamma(z)) = O(\Gamma(x))$

$\mathfrak{A}, \Gamma \models z \ {}_{\mathrm{O}}{\sim}_{\mathrm{I}} \ x$ if $O(\Gamma(z)) = I(\Gamma(x))$

$\mathfrak{A}, \Gamma \models z \ {}_{\mathrm{O}}{\sim}_{\mathrm{O}} \ x$ if $O(\Gamma(z)) = O(\Gamma(x))$

$\mathfrak{A}, \Gamma \models \varphi \vee \varphi'$ if $\mathfrak{A}, \Gamma \models \varphi$ or $\mathfrak{A}, \Gamma \models \varphi'$

$\mathfrak{A}, \Gamma \models \neg\varphi$ if we do not have $\mathfrak{A}, \Gamma \models \varphi$

$\mathfrak{A}, \Gamma \models \exists x.\varphi$ if there exists an element e of A such that $\mathfrak{A}, \Gamma[x \leftarrow e] \models \varphi$

If φ is a closed formula, $\mathfrak{A}, \Gamma \models \varphi$ does not depend on Γ so we write $\mathfrak{A} \models \varphi$ whenever there exists a context Γ such that $\mathfrak{A}, \Gamma \models \varphi$. We say that a formula φ is (finitely) satisfiable if there is a (finite) model $\mathfrak{A}$ such that $\mathfrak{A} \models \varphi$. Given a set S of formulae of IO-FO[Σ] we define the finite satisfiability problem for S as given a closed formula φ of S decide if φ is finitely satisfiable. We can specify the consensus problem with the following formula φ_{con} (where $q_f \in \Sigma$ means the process finished):

$$\varphi_{con} = \exists x.(x \ {}_{\mathrm{I}}{\sim}_{\mathrm{O}} \ x \wedge \forall y.(x \ {}_{\mathrm{I}}{\sim}_{\mathrm{O}} \ y \wedge q_f(y))).$$

We are able to know the difficulty of the finite satisfiability problem by reducing from satisfiability of first-order logic on undirected finite graphs, which is undecidable [5].

▶ **Theorem 1.** *The finite satisfiability problem for* IO-FO[Σ] *is undecidable.*

This result motivates the study of fragments of IO-FO[Σ]. We propose the logic Loc-IO-FO[Σ]:

$$\varphi ::= \psi_z(z) \ | \ x = y \ | \ \exists x.\varphi \ | \ \varphi \vee \varphi \ | \ \neg\varphi$$

$$\psi_z ::= a(x) \ | \ x = y \ | \ \exists x.\psi_z \ | \ \psi_z \vee \psi_z \ | \ \neg\psi_z \ | \ z \ {}_{\mathrm{I}}{\sim}_{\mathrm{I}} \ y \ | \ z \ {}_{\mathrm{I}}{\sim}_{\mathrm{O}} \ y \ | \ z \ {}_{\mathrm{O}}{\sim}_{\mathrm{I}} \ y \ | \ z \ {}_{\mathrm{O}}{\sim}_{\mathrm{O}} \ y$$

where $a \in \Sigma$ and $x, y, z \in \mathrm{Var}$ and with the three restrictions that in $\psi_z(z)$, there is one free variable z, that every comparison in terms of $\sim$ has to involve z and that in $\exists x.\psi_z$ the variables x and z must be distinct. In a sense, this is a kind of *guarded* fragment. Every Loc-IO-FO[Σ] formula can be seen as a IO-FO[Σ] formula in a straightforward way: it suffices to replace in the syntax tree every ψ_z by a φ. So we can see Loc-IO-FO[Σ] as a fragment of IO-FO[Σ]. The formula φ_{con} is in Loc-IO-FO[Σ]. In order to be convinced we rewrite φ_{con} by putting square brackets where we have $\psi_x(x)$:

$$\varphi_{con} = \exists x.[x \;{}_{\mathrm{I}}{\sim}_{\mathrm{O}}\; x \wedge \forall y.(x \;{}_{\mathrm{I}}{\sim}_{\mathrm{O}}\; y \wedge q_f(y))].$$

An example of a formula which is not in Loc-IO-FO[Σ] could be

$$\forall x.\exists y.\exists z.x \;{}_{\mathrm{I}}{\sim}_{\mathrm{I}}\; y \wedge y \;{}_{\mathrm{O}}{\sim}_{\mathrm{O}}\; z \wedge a(z).$$

Intuitively in Loc-IO-FO[Σ] we can not specify neighborhood's neighborhood of an element.

3 Satisfiability of local ioFO

We believe that the satisfiability of Loc-IO-FO[Σ] is decidable even though we do not have a proof yet. So far we have proven the decidability of the finite satisfiability for the existential fragment and obtained in this case the exact complexity bound. Furthermore we have as well an NEXPTIME lower bound for the general case.

By FO[Σ] we denote the subset of IO-FO[Σ] formulae which do not use the symbol $\sim$ (note that those formulae are Loc-IO-FO[Σ] too). We are able to decide the complexity and we show:

▶ **Theorem 2.** *Finite satisfiability for* FO[Σ] *is PSPACE-complete.*

We proved this by showing that FO[Σ] satisfies a small model property by means of Ehrenfeucht-Fraïssé games. This result may seem contradictory with Theorem 10 from [2] which states that the satisfiability of two-variable first-order logic with equality and unary predicates is NEXPTIME-complete. That's because in [2] on an element of a model any number of predicates can hold whereas in our logic exactly one predicate holds.

Next we define the existential fragment ∃Loc-IO-FO[Σ] as follows:

$$\begin{aligned}
\varphi \; &::= \; \psi_z(z) \mid x = y \mid \neg(x = y) \mid \exists x.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \\
\psi_z \; &::= \; a(x) \mid x = y \mid \exists x.\psi_z \mid \psi_z \vee \psi_z \mid \neg\psi_z \\
&\quad \mid z \;{}_{\mathrm{I}}{\sim}_{\mathrm{I}}\; y \mid z \;{}_{\mathrm{I}}{\sim}_{\mathrm{O}}\; y \mid z \;{}_{\mathrm{O}}{\sim}_{\mathrm{I}}\; y \mid z \;{}_{\mathrm{O}}{\sim}_{\mathrm{O}}\; y
\end{aligned}$$

with the same restriction on $\psi_z(z)$ as for Loc-IO-FO[Σ]. Notice that the formula φ_{cons} specifying consensus is in ∃Loc-IO-FO[Σ].

▶ **Theorem 3.** *Finite satisfiability for* ∃Loc-IO-FO[Σ] *is PSPACE-complete.*

We proved this theorem by a reduction to FO[Σ].

By Loc-IO-FO^2[Σ], we denote the fragment of Loc-IO-FO[Σ] that uses at most two variable names (which, however, can be reused at discretion). By a reduction from two-variable first-order logic over words which is NEXPTIME-complete by [2] we obtain a lower bound for the complexity of our logic:

▶ **Theorem 4.** *Finite satisfiability for* Loc-IO-FO^2[Σ] *is NEXPTIME-hard.*

In the future we aim at proving the following conjecture:

▶ **Conjecture 5.** *The finite satisfiability problem for* Loc-IO-FO[Σ] *is decidable.*

If it turns out that the conjecture is false then we will try to prove the decidability of formulae of bounded quantifier alternation at the top level. Another objective of our work is to use such logics to design some verification procedures of distributed algorithms.

References

1 Mikolaj Bojanczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data words. *ACM Trans. Comput. Log.*, 12(4):27:1–27:26, 2011. URL: `https://doi.org/10.1145/1970398.1970403`, `doi:10.1145/1970398.1970403`.

2 Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*, pages 228–235. IEEE Computer Society, 1997. URL: `https://doi.org/10.1109/LICS.1997.614950`, `doi:10.1109/LICS.1997.614950`.

3 Emanuel Kieronski and Martin Otto. Small substructures and decidability issues for first-order logic with two variables. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 448–457. IEEE Computer Society, 2005. URL: `https://doi.org/10.1109/LICS.2005.49`, `doi:10.1109/LICS.2005.49`.

4 Emanuel Kieronski and Lidia Tendera. On finite satisfiability of two-variable first-order logic with equivalence relations. In *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009, 11-14 August 2009, Los Angeles, CA, USA*, pages 123–132. IEEE Computer Society, 2009. URL: `https://doi.org/10.1109/LICS.2009.39`, `doi:10.1109/LICS.2009.39`.

5 Boris A. Trakhtenbrot. Impossibility of an algorithm for the decision problem in finite classes. *American Mathematical Society Translations*, 23:1–6, 1950. `doi:10.1090/trans2/023/01`.