Minimality and Non-Determinism in Multi-Context Systems

Floris Roelofsen[1] and Luciano Serafini[2]

[1] Institute for Logic, Language, and Computation
Amsterdam, Netherlands
[2] Instituto per la Ricerca Scientifica e Tecnologica
Trento, Italy

Abstract. Multi-context systems can be used to represent contextual information and inter-contextual information flow. We show that the local model semantics of a multi-context system is completely determined by the information that is obtained when simulating the information flow specified by the system, in such a way that a *minimal* amount of information is deduced at each step of the simulation.
The multi-context system framework implicitly presupposes that information flow is *deterministic*. In many natural situations, this is not a valid assumption. We propose an extension of the framework to account for non-determinism and provide an algorithm to efficiently compute the meaning of non-deterministic systems.

1 Introduction

The representation of contextual information and inter-contextual information flow has been formalized in several ways. Most notable are the propositional logic of context developed by McCarthy, Buvač and Mason [5, 6], and the multi-context systems devised by Giunchiglia and Serafini [3, 4], which later have been associated with the local model semantics introduced by Giunchiglia and Ghidini [2]. The two approaches have been compared by Serafini and Bouquet [9] from a technical viewpoint, and by Benerecetti et.al. [1] from a conceptual perspective.

A multi-context system describes the information available in a number of contexts (i.e., to a number of people / agents / databases, etc.) and specifies the information flow between those contexts. The local model semantics defines a system to entail a certain piece of information in a certain context, if and only if that piece of information is acquired in that context, independently of how the information flow described by the system is accomplished.

The first contribution of this paper is based on the observation that the local model semantics of a multi-context system is completely determined by the information that is obtained when simulating the information flow specified by the system, in such a way that a *minimal* amount of information is deduced at each step of the simulation. We define an operator which suitably implements such a simulation, and thus determines the information entailed by the system. This operator constitutes a first constructive account of the local model semantics.

The second contribution of this paper is based on the observation that, in its original formulation, the multi-context system framework implicitly rests on the assumption that information flow is *deterministic*. In many situations, this is not a suitable assumption. In a multi-agent scenario, for example, upon establishing a certain piece of information, an agent may decide to pass this information on to either one of a group of other agents. His choice as to which agent he will inform could be made non-deterministically. Another typical situation in which information flow is inherently non-deterministic is when the information channels between different contexts are subject to temporary failure or unavailability. Consider the case of online repositories. If information is obtained in one repository, the protocol may be to pass this information on to any one of a number of associated "mirror repositories": if the communication channel with one of these is defective or temporarily unavailable, another one is tried, until at least one successful communication is established.

The local model semantics is easily adapted to account for non-deterministic systems. However, if a system describes a non-deterministic information flow, then the minimal information entailed by the system cannot be determined unequivocally. We provide a way to generate from a non-deterministic system a number of deterministic systems, the semantics of which can be determined contructively, and which, together, completely determine the semantics of the original non-deterministic system.

We proceed, in section 2, with a brief review of multi-context system syntax and local model semantics. Minimality and non-determinism are discussed in section 3 and 4, respectively. We conclude, in section 5, with a concise recapitulation of our main observations and results.

2 Preliminaries

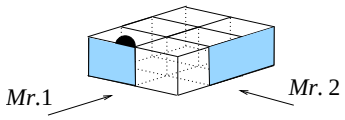

Fig. 1. A magic box.

A simple illustration of the main intuitions underlying the multi-context system framework is provided by the situation depicted in figure 1. Two agents, Mr.1 and Mr.2, are looking at a box from different angles. The box is called magic, because neither Mr.1 nor Mr.2 can make out its depth. As some sections of the box are out of sight, both agents have partial information about the box. To express this information, Mr.1 only uses proposition letters l (there is a ball on the left) and r (there is a ball on the right), while Mr.2 also uses a third proposition letter c (there is a ball in the center).

In general, we consider a set of contexts I, and a language L_i for each context $i \in I$. Henceforward, we assume I and $\{L_i\}_{i \in I}$ to be fixed, unless specified otherwise. Moreover, for the purpose of this paper we assume each L_i to be built over a finite set of proposition letters, using standard propositional connectives.

To state that the information expressed by a formula $\varphi \in L_i$ is established in context i we use so-called *labeled formulas* of the form $i : \varphi$ (if no ambiguity arises, we simply refer to labeled formulas as formulas, and we even use capital letters F, G, and H to denote labeled formulas, if the context label is irrelevant). A *rule* r is an expression of the form:

$$F \leftarrow G_1 \wedge \ldots \wedge G_n \tag{1}$$

where F and all G's are labeled formulas; F is called the consequence of r and is denoted by $cons(r)$; all G's are called premises of r and together make up the set $prem(r)$. Rules without premises are called *facts*. Rules with at least one premiss are called *bridge rules*. A *multi-context system* (system hereafter) is a finite set of rules. A fact describes information that is established in a certain context, independent of which information is obtained in other contexts. A bridge rule specifies which information is established in one context, if other pieces of information are obtained in different contexts. So a system can be seen as a specification of contextual information available a priori plus an inter-contextual information flow.

Example 1. The situation in figure 1 can be modeled by the following system S:

$$\begin{array}{ll} 1 : \neg r & \leftarrow \\ 2 : l & \leftarrow \\ 1 : l \vee r & \leftarrow 2 : l \vee c \vee r \\ 2 : l \vee c \vee r & \leftarrow 1 : l \vee r \end{array}$$

Mr.1 knows that there is no ball on the right, Mr.2 knows that there is a ball on the left, and if any agent gets to knows that there is a ball in the box, then he will inform the other agent about it.

A classical interpretations m of language L_i is called a *local model* of context i. A set of local models is called a *local information state*. Intuitively, every local model in a local information state represents a "possible state of affairs". If a local information state contains exactly one local model, then it represents complete information. If it contains more than one local model, then it represents partial information: more than one state of affairs is considered possible. A *distributed information state* is a set of local information states, one for each context. In conformity with the literature, we will refer to distributed information states as *chains*.

Example 2. The situation in figure 1, in which Mr.1 knows that there is no ball on the right but does not know whether there is a ball on the left, is represented by a chain whose first component $\{\{l, \neg r\}, \{\neg l, \neg r\}\}$ contains two local models. As such, the chain reflects Mr.1's uncertainty about the left section of the box.

A chain c *satisfies* a labeled formula $i : \varphi$ (denoted $c \models i : \varphi$) if and only if all local models in its i^{th} component classically satisfy φ. A rule r is *applicable* with respect to a chain c if and only if c satisfies every premiss of r. Notice that facts are applicable with respect to any chain. A chain c *complies with* a rule r, if and only if, whenever r is applicable with respect to c, then c satisfies r's consequence. We call c a *solution chain* of a system S if and only if it complies with every rule in S. A formula F is *true* in S (denoted $S \models F$) if and only if every solution chain of S satisfies F.

For convenience, we introduce some auxiliary terminology and notation. Let $\mathbf{C}$ denote the set of all chains. Notice that, as each L_i is assumed to be built over a finite set of proposition letters, $\mathbf{C}$ is assumed to be finite as well. Let $c^{\perp}$ denote the chain containing every local model of every context ($c^{\perp}$ does not satisfy any non-tautological expression); let $c^{\top}$ denote the chain containing no local models at all ($c^{\top}$ satisfies all expressions). If C is a set of chains, then the component-wise union (intersection) of C is the chain, whose i^{th} component consists of all local models that are in the i^{th} component of some (every) chain in C. If c and c' are chains, then $c \setminus c'$ denotes the chain, whose i^{th} component consists of all local models that are in c_i but not in c'_i. Finally, let us sometimes say that a local model m is (not) in c, when we actually mean that m is (not) in some (any) component c_i of c.

3 Minimality

We order chains according to the amount of information they convey. Intuitively, the more local models a chain component contains, the more possibilities it permits, so the less informative it is. Formally, we say that c is *less informative* than c' ($c \preceq c'$), if for every i we have $c_i \supseteq c'_i$. If, moreover, for at least one i we have $c_i \supset c'_i$, then we say that c is *strictly less informative* than c' ($c \prec c'$).

Lemma 1. *Let C be a set of chains. Let c^u (c^i) denote the component-wise union (intersection) of all chains in C. Then c^u (c^i) is less (more) informative than any chain in C.*

Proof. We proof the *union* part. Let c' be a chain in C. Then for every i, every local model m in c'_i is also in c^u_i. So $c^u_i \supseteq c'_i$, and thus $c^u \preceq c'$. □

Lemma 2. $(\boldsymbol{C}, \preceq)$ *forms a complete lattice.*

Proof. We should prove that every finite subset of $\mathbf{C}$ has both a greatest lower bound and a least upper bound in $\mathbf{C}$. Let C be a subset of $\mathbf{C}$ (note that $\mathbf{C}$ is finite, so C must be finite as well). Let c^u (c^i) denote the component-wise union (intersection) of all chains in C. Then, by lemma 1, c^u is a lower bound of C. Now consider a chain c', such that $c^u \prec c'$. For this to be the case, there must be a local model m, which is in c^u but not in c'. But then m must also be in some

chain c^m in C, which makes $c' \preceq c^m$ impossible. So c' cannot be a lower bound of C, which implies that c^u is the greatest lower bound of C. Analogously, it can be shown that c^i is least upper bound of C. □

Note that $c^\perp$ is strictly less informative than any other chain, whereas $c^\top$ is strictly more informative than any other chain. If $c \preceq c'$ we say that c' is an *extension* of c. So, intuitively, extending c corresponds to *adding* information to it. More technically, to extend c is to *remove* local models from it. We say that c is *minimal* among a set of chains C, if c is in C and no other chain c' in C is strictly less informative than c. In particular, we say that c is a *minimal solution chain* of a system S, if it is minimal among the set of all solution chains of S.

Lemma 3. *Let c and c' be two chains, such that $c \preceq c'$. Then any formula that is satisfied by c is also satisfied by c'.*

Proof. Suppose $c \models i : \phi$. Then, per definition, $m \models \phi$ for every $m \in c_i$. As c'_i is contained in c_i, we also have $m' \models \phi$ for every $m' \in c'_i$. So $c' \models i : \phi$. □

Lemma 4. *Let C be a set of chains and let c^u denote the component-wise union of all chains in C. Then a formula is satisfied by c^u if and only if it is satisfied by every chain in C.*

Proof.
(⇒) Follows directly from lemma 1 and lemma 3.
(⇐) Suppose all chains in C satisfy a formula $i : \phi$. Then all local models in the i^{th} component of every chain in C must satisfy ϕ. These are exactly the local models that make up the i^{th} component of c^u. So c^u must also satisfy $i : \phi$. □

Lemma 5. *Let S be a system. Then the set of all solution chains of S is closed under component-wise union. That is, if C is a set of solution chains of S, then the component-wise union c^u of all chains in C is again a solution chain of S.*

Proof. Let C be a set of solution chains of S. Let c^u be the component-wise union of C. Let r be an arbitrary rule in S. Then all c' in C comply with r. Suppose, towards a contradiction, that c^u does not comply with r, i.e., c^u satisfies all of r's premises, but does not satisfy r's consequence. By lemma 4 all c' in C satisfy all of r's premises, and therefore, by assumption, they all satisfy r's consequence as well. But then, again by lemma 4, c^u must also satisfy r's consequence, which contradicts the assumption that c^u does not comply with r. So c^u must comply with r, and as r was arbitrary, c^u must be a solution chain of S. □

Theorem 1. *Every system S has a unique minimal solution chain c_S.*

Proof. Every system has at least one solution chain, namely $c^\top$. Now, let S be a system and let C_S be the set of all its solution chains. Then, by lemma 5, the component-wise union c_S of C_S is itself in C_S. Moreover, by lemma 1, c_S is less informative than any other chain in C_S. So c_S is minimal among C_S and, moreover, any chain c' in C_S which is minimal among C_S, must be equal to c_S. In other words, c_S is the unique minimal solution chain of S. □

Theorem 2. *The meaning of a system S is completely determined by its unique minimal solution chain c_S. For any formula F we have:*

$$S \models F \quad \Leftrightarrow \quad c_S \models F$$

Proof. Let S be a system and let F be a formula. Then F is true in S if and only if F is satisfied by all solution chains of S. By lemma 4, this is the case if and only if F is satisfied by the component-wise union of all solution chains of S. From the proof of theorem (1) this union constitutes the minimal solution chain c_S of S. □

Theorem (1) and (2) are extremely useful, because they establish that, to answer queries about a system S, it is no longer necessary to compute all solution chains of S; we only need to consider the system's minimal solution chain c_S.

3.1 Computing the Minimal Solution Chain

Recall that a system S can be thought of as a specification of inter-contextual information flow. It turns out that the minimal solution chain of S can be characterized as the $\preceq$-least fixpoint of an operator $\mathbf{T}_S$, which, intuitively, simulates the information flow specified by S.

Let $S^*(c)$ denote the set of rules in S, which are applicable w.r.t. c. Then:

$$\mathbf{T}_S(c) = c \setminus \{m \mid \exists r \in S^*(c) : m \not\models \mathit{cons}(r)\} \tag{2}$$

For every rule r in S that is applicable w.r.t. c, $\mathbf{T}_S$ removes from c all local models that do not satisfy $\mathit{cons}(r)$. Intuitively, this corresponds to augmenting c with the information expressed by $\mathit{cons}(r)$. In this sense, $\mathbf{T}_S$ simulates the information flow described by S. Clearly, $\mathbf{T}_S(c)$ is obtained from c only by *removing* local models from it. As a result, $\mathbf{T}_S(c)$ is always more informative than c.

Lemma 6. *For every chain c and every system S: $c \preceq \boldsymbol{T}_S(c)$.* □

We now prove that, starting with the least informative chain $c^\bot$, $\mathbf{T}_S$ will reach its $\preceq$-least fixpoint after finitely many iterations, and that this $\preceq$-least fixpoint coincides with the minimal solution chain of S. The first result is typically established using Tarski's fixpoint theorem [10]. In order to apply this theorem, we first need to show that $\mathbf{T}_S$ is monotone and continuous with respect to $\preceq$.

Lemma 7. *$\boldsymbol{T}_S$ is monotone with respect to $\preceq$.*

Proof. Let c and c' be any two chains such that $c \preceq c'$. We need to prove that $\mathbf{T}_S(c) \preceq \mathbf{T}_S(c')$. Suppose, towards a contradiction that this is not the case. Then there is a local model m that belongs to $\mathbf{T}_S(c')$ but not to $\mathbf{T}_S(c)$. Clearly, m must already be present in c', and therefore also in c. From the fact that m has been removed from c by $\mathbf{T}_S$ it follows that there must be a rule r in S such that c satisfies $prem(r)$, whereas m does not satisfy $cons(r)$. But then, by lemma 3, c' must also satisfy $prem(r)$, so $\mathbf{T}_S$ should have removed m from c' as well. We conclude that $\mathbf{T}_S(c) \preceq \mathbf{T}_S(c')$. So $\mathbf{T}_S$ is monotone with respect to $\preceq$. □

Lemma 8. *$\boldsymbol{T}_S$ is continuous with respect to $\preceq$.*

Proof. Let $c^0 \preceq c^1 \preceq c^2 \preceq \ldots$ be an infinite sequence of chains, each of which contains more information than all preceding ones. We need to prove that $\mathbf{T}_S(\bigsqcup_{n=0}^{\infty} c^n) = \bigsqcup_{n=0}^{\infty} \mathbf{T}_S(c^n)$. As $\mathbf{C}$ is finite, $\{c^0, c^1, c^2, \ldots\}$ must have a maximum c^m in $\mathbf{C}$. So $\mathbf{T}_S(\bigsqcup_{n=0}^{\infty} c^n) = \mathbf{T}_S(c^m) = \bigsqcup_{n=0}^{\infty} \mathbf{T}_S(c^n)$. □

Theorem 3. *$\boldsymbol{T}_S$ has a $\preceq$-least fixpoint, which is obtained after a finite number of consecutive applications of $\boldsymbol{T}_S$ to $c^{\perp}$.*

Proof. Follows from lemmas 2, 7, and 8 by Tarski's fixpoint theorem [10]. □

Lemma 9. *Let c be a chain and let S be a system. Then c is a fixpoint of $\boldsymbol{T}_S$ if and only if c is a solution chain of S.*

Proof. A chain c is a fixpoint of $\mathbf{T}_S$ if and only if for every rule r in S, c satisfies $cons(r)$ whenever c satisfies $prem(r)$. This is the case if and only if c is a solution chain of S. □

Theorem 4. *Let S be a system. Then the minimal solution chain c_S of S coincides with the $\preceq$-least fixpoint of $\boldsymbol{T}_S$.*

Proof. Follows directly from lemma 9. □

From theorems 3 and 4 we conclude that the minimal solution chain c_S of a system S is obtained by a finite number of applications of $\mathbf{T}_S$ to the least informative chain $c^{\perp}$. But we can even prove a slightly stronger result:

Theorem 5. *Let S be a system and let $|S|$ denote the number of* bridge rules *in S. Then the minimal solution chain c_S of S is obtained by at most $|S| + 1$ consecutive applications of $\boldsymbol{T}_S$ to $c^{\perp}$.*

Proof. Let c be a chain and let S be a system. Notice that $\mathbf{T}_S(c)$ is a fixpoint of $\mathbf{T}_S$ if and only if $S^*(\mathbf{T}_S(c))$ coincides with $S^*(c)$. Lemmas 3 and 6 imply that, in any case, $S^*(\mathbf{T}_S(c)) \supseteq S^*(c)$. In other words, during each iteration of $\mathbf{T}_S$ some (possibly zero) rules are added to S^*. In the case that S^* remains unaltered, $\mathbf{T}_S$ must have reached a fixpoint. Now we observe that during the first application of $\mathbf{T}_S$ (to $c^\perp$) all facts in S are added to S^*. Clearly, after that, $\mathbf{T}_S$ can be applied at most $|S|$ times before a fixpoint is reached. □

In fact, a slightly more involved, but essentially equivalent procedure was introduced for rather different reasons in [7]. This procedure was shown to have worst-case time complexity $O(|S|^2 \times 2^M)$, where M is the maximum number of propositional variables in either one of the contexts involved in S. The greater part of a typical computation is taken up by propositional reasoning within individual contexts, which itself requires exponential time in the worst case.

Example 3. Consider the system S given in example 1. Applying $\mathbf{T}_S$ to $c^\perp$ establishes the facts given by the first two rules of the system. But then Mr.2 knows that there is a ball in the box, so the next application of $\mathbf{T}_S$ simulates the information flow specified by the third rule of the system: Mr.2 informs Mr.1 of the presence of the ball. The resulting chain is left unaltered by any further application of $\mathbf{T}_S$, and therefore constitutes the minimal solution chain of S. The fact that this chain satisfies the formula $1 : l$ reflects, as desired, that Mr.1 has come to know that there is a ball in the left section of the box.

4 Non-Determinism

The original formulation of multi-context systems implicitly rests on the assumption that information flow is *deterministic*. However, there are many natural situations in which information flow is inherently non-deterministic.

Example 4. Adriano is on holiday after having submitted his final school exams. He has promised to call his father or his mother in case his teacher lets him know that he has passed his exams. This situation can be modeled by a system S consisting of the following rule:

$$m : p \textbf{ or } f : p \leftarrow a : p$$

Notice that Adriano may be conceived of as an agent in a multi-agent system, who non-deterministically decides which other agents to inform when acquiring novel information. Alternatively, Adriano's parents may be conceived of as mirror repositories of information about Adriano's well-being (assuming that they tell each other everything they come to know about Adriano). Typical telephonic connections may be broken or temporarily unavailable. Analogous to the situation sketched in the introduction, Adriano will try to reach his parents, until at least one of them is informed.

In general, we would like to consider systems in which rules r are of the form:

$$F_1 \textbf{ or } \ldots \textbf{ or } F_m \leftarrow G_1 \wedge \ldots \wedge G_n \tag{3}$$

where all F's and G's are labeled formulas; all F's are called consequences of r and together form the set $cons(r)$; and as before, all G's are called premises of r and together constitute the set $prem(r)$. A rule doesn't necessarily have any premises ($n \geq 0$), but always has at least one consequence ($m \geq 1$). We call a rule deterministic if it has only one consequence, and non-deterministic otherwise. We call finite sets of possibly non-deterministic rules *non-deterministic multi-context systems* (non-deterministic systems for short). Systems which consist of deterministic rules only, are from now on refered to as deterministic systems.

A chain c complies with a non-deterministic rule r if and only if, whenever r is applicable w.r.t. c, *at least one of* its consequences is satisfied by c. A chain is a solution chain of S if and only if it complies with all rules in S. A formula F is true in S, $S \models F$, if and only if F is satisfied by all solution chains of S.

Observation 1 *Let S be a non-deterministic system, let c' and c'' be two solution chains of S, and let c be the component-wise union of c' and c''. Then it is* not *generally the case that c is again a solution chain of S. Therefore, S does not generally have a unique minimal solution chain.*

Example 5. Suppose Adriano's teacher lets him know that he passed his exams. The resulting system S is given by the following rules:

$$a : p \leftarrow$$
$$m : p \textbf{ or } f : p \leftarrow a : p$$

This system has two minimal solution chains:

$$c^m = \{\{p\}_m, \quad \{p, \neg p\}_f, \quad \{p\}_a\}$$
$$c^f = \{\{p, \neg p\}_m, \quad \{p\}_f, \quad \{p\}_a\}$$

whose component-wise union $\{\{p, \neg p\}_m, \quad \{p, \neg p\}_f, \quad \{p\}_a\}$ is not a solution chain of S.

Theorem 6. *The meaning of a non-deterministic system S is completely determined by the set C_S of all its minimal solution chains. For any formula F we have:*

$$S \models F \quad \Leftrightarrow \quad \forall c \in C_S : c \models F$$

Proof. Let S be a non-deterministic system and let F be a formula. Then F is true in S if and only if F is satisfied by every solution chain of S. Clearly, if F is satisfied by every solution chain of S, then it must in particular be satisfied by every minimal solution chain of S. Moreover, every solution chain of S is an extension of some minimal solution chain of S, which implies, by lemma 3, that F is satisfied by all minimal solution chains of S only if F is satisfied by all solution chains of S. □

Theorem 6 establishes that the meaning of a non-deterministic system S is completely determined by the set C_S of all its minimal solution chains. We will now provide a way to compute C_S, re-using the method outlined in section 3.

4.1 Computing Minimal Solution Chains

Our approach, inspired by an idea originally developed for disjunctive databases [8], is to generate from a non-deterministic system S a number of deterministic systems S_1, S_2, ..., S_n, in such a way that the minimal solution chains of S are among the minimal solution chains of S_1, S_2, ..., S_n (note that each S_i has a unique minimal solution chain which can be computed as outlined in section 3). Hereto, we introduce the notion of a *generated system*. Let S be a non-deterministic system and let r be a rule in S. Then we say that a deterministic rule r' is *generated by* r if and only if $cons(r') \in cons(r)$ and $prem(r') = prem(r)$. We say that a system S' is *generated by* S if and only if it is obtained from S by replacing each rule r in S by some rule r' generated by r. Notice that, indeed, a generated system is always deterministic, and that any non-deterministic system S generates at most $\prod_{r \in S} |cons(r)|$ different deterministic systems.

Example 6. The non-deterministic system from example 5 generates two deterministic systems: $\{a : p \leftarrow, \quad m : p \leftarrow a : p\}$ and $\{a : p \leftarrow, \quad f : p \leftarrow a : p\}$. The only system generated by a deterministic system is that system itself.

Lemma 10. *A chain c is a solution chain of a non-deterministic system S if and only if it is a solution chain of some system S' generated by S.*

Proof.
($\Rightarrow$) Suppose c is a solution chain of S. Then c complies with every rule in S. For every rule r in S, if c complies with r, then there must be a rule r' generated by r such that c complies with r' as well. Let S' be the system $\{r' \mid r \in S\}$. Then c is a solution chain of S'.
($\Leftarrow$) Suppose c is a solution chain of a system S' generated by S. Then c complies with every rule in S'. Every rule r in S has generated some rule r' in S', and clearly, if c complies with r' then it must also comply with r. So c is a solution chain of S. □

We call a chain c a *potential solution chain* of S if and only if c is a minimal solution chain of some system S' generated by S.

Lemma 11. *Every minimal solution chain of a system S is also a potential solution chain of S.*

Proof. Suppose c is a minimal solution chain of S. Then, by lemma 10, c is a solution chain of some system S' generated by S. Let c' be the minimal solution chain of S'. Then c must be an extension of c'. By lemma 10 c' must be a solution chain of S. But then, as c is a minimal solution chain of S, c' must be equal to c. So c is a minimal solution chain of S', and therefore a potential solution chain of S. □

Observation 2 *It is* not *generally the case that a potential solution chain of S is also a minimal solution chain of S.*

Example 7. Suppose Adriano's teacher also called Adriano's mother to tell her the good news. The resulting system S is given by the following rules:

$$a : p \leftarrow$$
$$m : p \leftarrow$$
$$m : p \textbf{ or } f : p \leftarrow a : p$$

This system has two potential solution chains:

$$c^m = \{\{p\}_m, \quad \{p, \neg p\}_f, \quad \{p\}_a\}$$
$$c^{mf} = \{\{p\}_m, \quad \{p\}_f, \quad \{p\}_a\}$$

But as c^{mf} extends c^m only the latter is a minimal solution chain of S.

We call c an *essential solution chain* of S if and only if c is minimal among all potential solution chains of S.

Theorem 7. *A chain is a minimal solution chain of S if and only if it is an essential solution chain of S.*

Proof.
($\Rightarrow$) Suppose c is a minimal solution chain of S. Then, by lemma 11, c is a potential solution chain of S. If c is minimal among all potential solution chains of S, then, per definition, it is essential. Now, towards a contradiction, suppose that c is *not* minimal among all potential solution chains of S. Then there must be another potential solution chain c' of S, such that $c' \prec c$. But, by lemma 10, c' must also be a solution chain of S, which contradicts the assumption that c is a minimal solution chain of S.
($\Leftarrow$) Suppose c is an essential solution chain of S. Furthermore, towards a contradiction, suppose that c is *not* a minimal solution chain of S. Then there must be a minimal solution chain c' of S, such that $c' \prec c$. By lemma 11, c' is a potential solution chain of S. But this contradicts the assumption that c is minimal among all potential solution chains of S. □

Theorem 7 establishes that, to compute the meaning of a non-deterministic system S it suffices to compute the meaning of all deterministic systems generated by S. This can be done re-using the method developed in section 3. Given that S generates at most $\prod_{r \in S} |cons(r)|$ different systems, and that computing the meaning of each of these systems takes at most time $O(|S|^2 \times 2^M)$, we conclude that, in the worst case, computing the meaning of S takes time $O(\prod_{r \in S} |cons(r)| \times |S|^2 \times 2^M)$.

5 Conclusions

In this paper, we investigated the multi-context system formalism as a framework for representing contextual information and inter-contextual information flow.

We observed that the semantics of a multi-context system is completely determined by the information that is obtained when simulating the information flow specified by the system, in such a way that a *minimal* amount of information is deduced at each step of the simulation. Based on this observation, we defined an operator which determines the information entailed by the system by implementing a suitable simulation of the prescribed information flow. This operator provides a first constructive account of the local model semantics.

Next we observed that the multi-context system framework implicitly rests on the assumption that information flow is *deterministic*. We sketched a number of situations, in which this is not a valid assumption. We extended the framework in order to account for non-deterministic information flow, and provided a way to express the semantics of a non-deterministic system in terms of the semantics of a number of associated, deterministic systems. This allowed us to give a constructive account of the semantics of non-deterministic systems as well.

References

1. M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual reasoning distilled. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(3):279–305, 2000.
2. C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.
3. F. Giunchiglia. Contextual reasoning. *Epistemologia*, XVI:345–364, 1993.
4. F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics, or: how we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994.
5. J. McCarthy. Notes on formalizing context. In *International Joint Conference on Artificial Intelligence (IJCAI 93)*, pages 555–560, 1993.
6. J. McCarthy and S. Buvač. Formalizing context (expanded notes). In *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. 1998.
7. F. Roelofsen, L. Serafini, and A. Cimatti. Many hands make light work: Localized satisfiability for multi-context systems. In *European Conference on Artificial Intelligence (ECAI 04)*, pages 58–62, 2004.
8. C. Sakama and K. Inoue. An alternative approach to the semantics of disjunctive programs and deductive databases. *Journal of Automated Reasoning*, 13:145–172, 1994.
9. L. Serafini and P. Bouquet. Comparing formal theories of context in AI. *Artificial Intelligence*, 155:41–67, 2004.
10. A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.