CHAPTER 28

DECLARATIVE SEMANTICS OF BELIEF QUERIES IN MLS DEDUCTIVE DATABASES

Hasan M. Jamil
Department of Computer Science
Mississippi State University, USA
jamil@cs.msstate.edu

Abstract A logic based language, called *MultiLog*, for multi level secure relational databases has recently been proposed. It has been shown that MultiLog is capable of capturing the notion of user *belief*, of filtering unwanted and *"useless"* information in its proof theory. Additionally, it can guard against a previously unknown security breach – the so called *surprise stories*. In this paper, we outline a possible approach to a declarative characterization of belief queries in MultiLog in a very informal manner. We show that for *"simple programs"* with belief queries, the semantics is rather straight forward. Semantics for the general Horn programs may be developed based on the understanding of the model theoretic characterization of belief queries developed in this paper.

Keywords: Multi level security, belief queries, declarative semantics, completeness.

Introduction

In a recent research, Jukic and Vrbsky [8] demonstrate that users in the relational MLS model potentially have a cluttered view and ambiguous belief of "visible data", and that the extraction process of knowledge and belief about data from such databases is manual and thus, error prone. In an earlier research [4], we showed that ad hoc knowledge extraction is quite an undertaking in such models, and understanding what others believe is not easily possible. We also showed that a special form of security breach, called *surprise stories*, is still possible in the MLS models and thus, have devised ways to guard against such breaches in *MultiLog*, a query language for deductive MLS databases. We continue to argue that it is imperative for users to theorize about the beliefs of other users at different visible levels. Current models, unfortunately, do not provide any support to this end. We have addressed some of the issues we perceive as bottlenecks for contemporary proposals in [4], We will not elaborate on those issues here for the sake of conciseness. We refer the readers to [3] for an introduction to MLS data model, and to [2, 8, 4] for a discussion of its shortcomings and possible enhancements. We also do not include a detailed discussion on

the syntax and the proof theory of MultiLog in this paper due to space limitations. Interested readers may refer to [4] for a preparatory reading. But for the sake of completeness, we present the syntax of all types of atoms of MultiLog below. The formulas and clauses of MultiLog are constructed in a way similar to classical logic programs.

MultiLog syntax includes a variety of atoms constructed from the alphabet of MultiLog language $\mathcal{L}$. In this alphabet (i) p is a predicate symbol, and *order* and *level* are two distinguished predicates, (ii) t_is, v and k are terms, (iii) a is an attribute name, (iv) l, h, s and c are symbols representing security levels, (v) and finally, m is a belief mode representing *firm* (fir), *cautious* (cau) or *optimistic* (opt) belief of a user. The so called m-atoms are of the form $s[p(k : a \xrightarrow{c} v)]$, and the b-atoms are constructed from m-atoms as $s[p(k : a \xrightarrow{c} v)] \ll m$. The p-atoms (or general predicates) are of the form $p(t_1, \ldots, t_k)$, while two distinguished predicates called the l-atoms and h-atoms are respectively of the form $level(s)$, and $order(l, h)$.

Our goals for this paper are two-fold: (i) to develop a direct Herbrand semantics for a subset of definite Horn clause fragment of MultiLog, called the *simple programs* by defining a model theory and a fixpoint theory, and (ii) to demonstrate on intuitive grounds that the equivalence of MultiLog's three characterizations – proof theory, model theory and fixpoint theory can be easily established for simple programs. Through this equivalence, we hope to convince readers that MultiLog's unique features and modeling capabilities, many of which are non-monotonic in nature, do not compromise the soundness and completeness of the language. This development is significant from a theoretical perspective, as it gives insight into the understanding of the logical behavior and mathematical foundations of the language. Complete details of the ideas discussed in this paper may be found in an extended version [6] elsewhere.

1. DECLARATIVE SEMANTICS OF MULTILOG

The Herbrand semantics of MultiLog databases can be defined in terms of a composite set-theoretic structure which provides a model for each of the security levels in the language $\mathcal{L}$ of MultiLog, including the level $s_\perp$ – the system level which is not part of any database universe. In other words, each model in the composite structure interprets formulas pertaining to the corresponding levels in the security hierarchy. The notion of *"belief"* in such a structure is then captured using a function level semantics over the sets in the Herbrand structure, not as a set membership. The notion of Herbrand universe $\mathcal{U}$ and base $\mathcal{H}$ is defined in a manner similar to the classical case. Formally, an *Herbrand structure* **H** of $\mathcal{L}$ is a tuple $\langle H(s) : s \in \mathcal{S} \rangle$ such that $H(s) \subseteq \mathcal{H}$ for every $s \in \mathcal{S}$. When $s \neq s_\perp$, $H(s)$ contains only m-atoms, otherwise $H(s_\perp)$ contains only p-, l- and h-atoms. Intuitively, every $H(s)$, $s \neq s_\perp$ in **H** interprets the associated data items belonging to the level s as ground m-atoms that are true with respect to level s in **H**. To make a distinction between an interpretation corresponding to a security level and the interpretation structures for our language $\mathcal{L}$, we henceforth call them interpretations and T-interpretations respectively, since the latter is actually a tuple of simple interpretations or sets.

Definition 1.1 (Satisfaction of Formulas) Let **H** be aT-interpretation, $\bar{u}$ be a user clearance level, $H(s)$ be any arbitrary interpretation in **H** where s is a security level in **H**, and let n be the number of such security levels. Furthermore, let A and B denote ground atomic formulas, and F and G denote any arbitrary ground

formulas. Then, the satisfaction of ground formulas with respect to $H(s)$ in $\mathbf{H}$, denoted $H(s)\models_{\mathbf{H},\bar{u}} A$, or $H(s)\models_{\mathbf{H},\bar{u}} F$, is defined as follows:

(1) $H(s)\models_{\mathbf{H},\bar{u}} A \iff A \in H(s)$ where $depth(A) = s$ and $s \preceq \bar{u}$

(2) $H(s)\models_{\mathbf{H},\bar{u}} A \iff H(o)\models_{\mathbf{H},\bar{u}} A$ where $depth(A) = o$ and $s \neq o$

(3) $H(i)\models_{\mathbf{H},\bar{u}} A \leftarrow B_1,\ldots,B_m \iff H(i)\models_{\mathbf{H},\bar{u}} B_g,\ g = 1,\ldots,m \Longrightarrow H(i)\models_{\mathbf{H},\bar{u}} A$

(4) $H(i)\models_{\mathbf{H},\bar{u}} l[p(k : a \xrightarrow{c} v] \ll m \iff H'\models_{\mathbf{H},\bar{u}} l[p(k : a \xrightarrow{c} v]$ where $H' = \beta(\bigcup_{j=1}^{n} H(s_j), \bar{u}, m)$ such that $\forall j, s_j \preceq l$ and $l \preceq \bar{u}$

Finally, we say that $\mathbf{H} \models_{\bar{u}} A$ if and only if $H(l)\models_{\mathbf{H},\bar{u}} A$, where $l = depth(A)$.

In the definition above, β is a belief function defined as follows. Let S be an arbitrary set of m-atoms, $\bar{u}$ be a clearance level, $A = l[p(k : a \xrightarrow{c} v)]$ be a ground m-atom, and m be a belief mode in $\mu = \{fir, cau, opt\}$. Then, the belief function $\beta : \mathcal{P}(\mathbf{S}) \times \mathcal{S} \times \mu \rightarrow \mathcal{P}(\mathbf{S})$, where $\mathbf{S}$ is a set of all possible m-atoms and S is the set of all security symbols, such that:

$$\beta(S,\bar{u},m) = \left\{ \bar{u}[p(k : a \xrightarrow{c} v)] \;\middle|\; \begin{array}{l} \textbf{One of the following conditions hold:} \\ - \ m = fir \text{ and } \bar{u}[p(k : a \xrightarrow{c} v)] \in S \\ - \ m = cau \text{ and } \exists l'[p(k : a \xrightarrow{c} v)] \in S,\ l' \preceq \bar{u}, \text{ and} \\ \quad \neg\exists l''[p(k : a \xrightarrow{c'} v')] \in S,\ l'' \preceq \bar{u}, \text{ and } c \prec c'. \\ - \ m = opt \text{ and } l'[p(k : a \xrightarrow{c} v)] \in S \text{ and } l' \preceq \bar{u} \end{array} \right.$$

Furthermore, the depth of a p-, l- or h-atom is defined to be $s_\perp$ and as s for m- or b-atoms of the forms $s[p(k : a \xrightarrow{c} v)]$ and $s[p(k : a \xrightarrow{c} v)] \ll m$ respectively. The notion of models (T-models to be precise) can be developed using the machinery above. We explain the idea through a couple of examples. In the examples that follow, a database D is said to be in level l, denoted $\langle D, l\rangle$, if a user with a clearance level l accesses the database that sets a context for the queries and the responses returned by the database.

Example 1.1 Consider the following database $\langle D_1, c\rangle$ below. In this example, and also throughout this paper, we consider only four security levels for simplicity. Namely, $s_\perp$, u, c, and s with a total order $s_\perp < u < c < s$. That is, we have in our database the atoms *order(u,c),* and *order(c,s),* and that *order*$(s_\perp, u)$ is implicit.

$$\Lambda_{D_1} := \left| \begin{array}{ll} r_1: & level(u). \\ r_2: & level(c). \\ r_3: & level(s). \\ r_4: & order(u,c). \\ r_5: & order(c,s). \end{array} \right. \qquad \Sigma_{D_1} := \left| \begin{array}{ll} r_6: & c[p(k : a \xrightarrow{u} v)]. \\ r_7: & c[p(k : a \xrightarrow{c} t)] \leftarrow p(j). \\ r_8: & s[p(K : A \xrightarrow{O} V)] \leftarrow \\ & \quad c[p(K : A \xrightarrow{O} V)] \ll cau. \end{array} \right.$$

$$\Pi_{D_1} := \left| \begin{array}{ll} r_9: & q(j). \end{array} \right. \qquad \Omega_{D_1} := \left| \begin{array}{ll} r_{10}: & ?\ s[p(k : a \xrightarrow{u} v)] \end{array} \right.$$

For the database above, let the T-model be M_1, as shown below:

$$M_1 = \underbrace{\{\underbrace{\{level(u),\ level(c),\ level(s),\ order(u,c),\ order(c,s), p(j)\}}_{M_1(s_\perp)}, \underbrace{\emptyset}_{M_1(u)},}_{\underbrace{\{c[p(k : a \xrightarrow{u} v)], c[p(k : a \xrightarrow{c} t)]\}}_{M_1(c)}, \underbrace{\{s[p(k : a \xrightarrow{c} t)]\}}_{M_1(s)}\})$$

In the T-model M_1 above, the first set in the interpretation belongs to level $s_\perp$, i.e., $M_1(s_\perp)$. The second set belongs to u, the third to level c, and the last to s. Now,

several observations can be made here. Note that the database is at level c. Also note that $s[p(k : a \xrightarrow{c} t)] \in M_1$, $M_1(s)$ to be precise. Still $M_1 \not\models_c s[p(k : a \xrightarrow{c} t)]$. This is because $M_1(s) \not\models_{M_1,c} s[p(k : a \xrightarrow{c} t)]$ as it does not satisfy condition 1 of definition 1.1 of formula satisfaction, i.e., $s \not\preceq c$. But $M_1 \models_c c[p(k : a \xrightarrow{u} v)]$, and also $M_1 \models_c c[p(k : a \xrightarrow{c} t)]$. Yet, it is interesting to verify that $M_1 \models_c c[p(k : a \xrightarrow{c} t)] \ll cau$ but $M_1 \not\models_c c[p(k : a \xrightarrow{u} v)] \ll cau$. This observation follows from the definition of cautious belief in β for Herbrand sets.

But not every T-model is *"intended"* and the construction of an intended T-model is not so straightforward. Recall that the satisfaction of b-atoms depends on the belief function β which makes use of the Herbrand sets in **H**. Also recall that the set computed by β depends on the elements in the Herbrand sets corresponding to security levels dominated by user clearance $\bar{u}$ (alternatively, by the level of the database). While the satisfaction of b-atoms is not affected by elements not required for a structure to be a T-model for a database $\langle \Delta, \bar{u} \rangle$, it potentially affects the beliefs of users as unwanted models may result. The following example helps clarify this point.

Example 1.2 Consider a level s database $\langle D_2, s \rangle$. Assume that database D_2 is derived from database D_1 of example 1.1 by replacing rule r_6 with $u[p(k : a \xrightarrow{u} v)]$, rule r_8 by $s[p(k : a \xrightarrow{u} v)] \leftarrow c[p(k : a \xrightarrow{u} v)] \ll cau$, and finally by deleting rule r_9 and adding two rules $r_{11} : p(X) \leftarrow r(X)$ and $r_{12} : r(X) \leftarrow q(X)$. Now for database $\langle D_2, s \rangle$ as defined, the intended T-model M_2 may be identified as follows:

$$M_2 = \langle \underbrace{\{level(u),\ level(c),\ level(s),\ order(u,c),\ order(c,s)\}}_{M_2(s_\perp)},$$
$$\underbrace{\{u[p(k : a \xrightarrow{u} v)]\}}_{M_2(u)}, \underbrace{\emptyset}_{M_2(c)}, \underbrace{\{s[p(k : a \xrightarrow{u} v)]\}}_{M_2(s)} \rangle$$

However, it is easy to verify that M_2' or M_2'' below are not intended although they are T-models for D_2.

$$M_2' = \langle \underbrace{\{level(u),\ level(c),\ level(s),\ order(u,c),\ order(c,s)\}}_{M_2'(s_\perp)},$$
$$\underbrace{\{u[p(k : a \xrightarrow{u} v)]\}}_{M_2'(u)}, \underbrace{\{c[p(k : a \xrightarrow{c} t)]\}}_{M_2'(c)}, \underbrace{\emptyset}_{M_2'(s)} \rangle$$
$$M_2'' = \langle \underbrace{\{level(u),\ level(c),\ level(s),\ order(u,c),\ order(c,s), p(j)\}}_{M_2''(s_\perp)},$$
$$\underbrace{\{u[p(k : a \xrightarrow{u} v)]\}}_{M_2''(u)}, \underbrace{\{c[p(k : a \xrightarrow{c} t)]\}}_{M_2''(c)}, \underbrace{\emptyset}_{M_2''(s)} \rangle$$

M_2' and M_2'' are not intended because they make $s[p(k : a \xrightarrow{u} v)]$ false, i.e., $M_2' \not\models_s s[p(k : a \xrightarrow{u} v)]$ and $M_2'' \not\models_s s[p(k : a \xrightarrow{u} v)]$, for similar but different reasons (for $c[p(k : a \xrightarrow{c} t)]$ being in $M_2'(c)$ and as well as in $M_2''(c)$ that made satisfaction of $c[p(k : a \xrightarrow{u} v)] \ll cau$ not possible, instead forced $c[p(k : a \xrightarrow{c} t)]$ to be believed, but cautiously, at level c). If either one of these T-models were minimal, it would have modeled $s[p(k : a \xrightarrow{u} v)]$, as dictated by logical entailment and implication. A careful observation will reveal that if the component models are the smallest (no extra atoms present than needed to be a model), then the composite T-interpretation stands a chance to be an *intended T-model*.

1.1. FIXPOINT THEORY

The issue now is – can the intended model of a database be constructed algorithmically? In this section, we present a constructive way of defining the least T-model for a MultiLog database $\langle \Delta, \bar{u} \rangle$. The key idea is to construct the least T-model $\mathbf{M}_\Delta$ of a database $\langle \Delta, \bar{u} \rangle$ by means of a bottom-up least fixpoint computation based on an immediate consequence operator $\mathbf{T}^{\bar{u}}_\Delta$. Since our T-interpretations are tuples of interpretations, we define $\mathbf{T}^{\bar{u}}_\Delta$ in terms of the immediate consequence transformation of each of the levels in $\langle \Delta, \bar{u} \rangle$.

Definition 1.2 (Fixpoint Operator) Let Δ be a "closed" database and let $\hat{\Delta} = \langle \hat{\Lambda}, \hat{\Sigma}, \hat{\Pi}, \hat{\mathcal{Q}} \rangle$ be its Herbrand instantiation defined as usual. Let I be an Herbrand interpretation for Δ. We define $\mathbf{T}^{\bar{u}}_\Delta$ to be the immediate consequence operator such that $\mathbf{T}^{\bar{u}}_\Delta = \langle T^{\bar{u}}_\Delta(I(s)) : s \in \mathcal{S} \rangle$. The operator $T^{\bar{u}}_\Delta$ for each component $I(s) \in I$ is defined similar to the classical case as $T^{\bar{u}}_\Delta : \mathcal{P}(\mathcal{H}) \mapsto \mathcal{P}(\mathcal{H})$, such that

$$T^{\bar{u}}_\Delta(I(s)) = \{A \mid A \leftarrow G \in \hat{\Delta}, depth(A) = s \text{ and } I(s) \models_{I,\bar{u}} G\}$$

Unfortunately, the fixpoint of $\mathbf{T}^{\bar{u}}_\Delta$ does not yield the the intended model of $\langle \Delta, \bar{u} \rangle$ in general. This is because the belief function β is non-monotonic in nature (recall the case of overriding of less strict data item in a cautious mode), and thus the behavior of β depends on the stage of computation. While it does not affect the monotonicity of the $\mathbf{T}^{\bar{u}}_\Delta$ operator, it does spoil the intended model computation process. The following example exposes the unscrupulous nature of $\mathbf{T}^{\bar{u}}_\Delta$.

Example 1.3 Consider the database $\langle D_3, s \rangle$ derived from database D_2 in example 1.2 by adding the rule $r_{13} : q(j)$ in Π_{D_3}. For the database $\langle D_3, s \rangle$, the intended T-model M_3 may be identified as follows:

$$M_3 = \langle \underbrace{\{level(u),\ level(c),\ level(s),\ order(u,c),\ order(c,s),\ q(j),\ r(j),\ p(j)\}}_{M_3(s_\perp)},$$
$$\underbrace{\{u[p(k : a \xrightarrow{u} v)]\}}_{M_3(u)}, \underbrace{\{c[p(k : a \xrightarrow{c} t)]\}}_{M_3(c)}, \underbrace{\emptyset}_{M_3(s)} \rangle$$

However, if we consider the sets computed at every stage of $T^{\bar{u}}_\Delta$, we have the following sequence,

$$\begin{aligned} \delta_1 &= \{level(u),\ level(c),\ order(u,c),\ q(j),\ u[p(k : a \xrightarrow{u} v)]\} \\ \delta_2 &= \{r(j),\ s[p(k : a \xrightarrow{u} v)]\} \\ \delta_3 &= \{p(j)\} \\ \delta_4 &= \{c[p(k : a \xrightarrow{c} t)]\} \end{aligned}$$

giving us the T-model

$$M'_3 = \langle \underbrace{\{level(u),\ level(c),\ level(s),\ order(u,c),\ order(c,s),\ q(j),\ r(j),\ p(j)\}}_{M'_3(s_\perp)},$$
$$\underbrace{\{u[p(k : a \xrightarrow{u} v)]\}}_{M'_3(u)}, \underbrace{\{c[p(k : a \xrightarrow{c} t)]\}}_{M'_3(c)}, \underbrace{\{s[p(k : a \xrightarrow{u} v)]\}}_{M'_3(s)} \rangle$$

which is not intended as the component model $M'_3(s)$ is not minimal, i.e., $M'_3(s) \neq \emptyset$. As such the query returns the answer `true`.

It turns out that if b-atoms are allowed only in the queries, and not in the clauses, then the intended models can be constructed fairly easily. Such restricted databases

(or programs) are called *simple* databases (or programs). For simple databases it is easy to prove the following results.

Proposition 1.1 (Existence and Uniqueness of Intended Models) For any consistent database $\langle \Delta, \bar{u} \rangle$ [4, 6] and a least consistent T-model I of $\langle \Delta, \bar{u} \rangle$ [6], I is the unique intended model of $\langle \Delta, \bar{u} \rangle$ if $\langle \Delta, \bar{u} \rangle$ is simple.

Theorem 1.1 (Least T-models) Let $\langle \Delta, \bar{u} \rangle$ be a database and $\mathbf{M}_\Delta$ be its least T-model. Then, $\mathbf{M}_\Delta = lfp(\mathbf{T}^{\bar{u}}_\Delta) = \mathbf{T}^{\bar{u}}_\Delta \uparrow^\omega$.

The equivalence between the model theoretic semantics and the proof theory can now be established as follows.

Theorem 1.2 (Equivalence) Let $\langle \Delta, \bar{u} \rangle$ be a database, $\mathbf{M}_\Delta$ be its least T-model, and G be a ground goal. Then, we have

$$\langle \Delta, \bar{u} \rangle \vdash_e G \iff \mathbf{M}_\Delta \models_{\bar{u}} G$$

2. CONCLUSION

For simplicity of presentation, we have assumed that our databases are simple, and have essentially made them free from b-atoms while we still allowed b-atoms in the queries. This restriction can be removed by considering a more elaborate model theoretic treatment similar to stratification [1], or by taking an approach similar to the overriding concept developed in [7]. Our recent work on parametric inheritance [5] also provides additional formal basis for the belief function we have introduced in [4], and used in this paper. Intuitively, the rules containing b-atoms should be placed at the highest possible stratum so that the belief computation kicks off when the m-atoms are completely computed. But if the computation of m-atoms depends upon b-atoms, the scenario will become complicated and we will possibly have to settle for multiple minimal models. These are some of the issues we seek to investigate in our future research. The details may be found in an extended version of this paper in [6].

References

[1] M. Bugliesi and H. M. Jamil. A stable model semantics for behavioral inheritance in deductive object oriented languages. In *Proc ICDT*, pages 222–237, 1995.

[2] F. Cuppens. Querying a multilevel database: A logical analysis. In *VLDB Proc.*, pages 484–494, 1996.

[3] S. Jajodia and R. Sandhu. Toward a multilevel secure relational data model. In *ACM SIGMOD*, pages 50–59, 1991.

[4] Hasan M. Jamil. Belief reasoning in MLS deductive databases. In *ACM SIGMOD*, pages 109–120, 1999.

[5] H. M. Jamil. A logic based language for parametric inheritance. In *Proc KR '2000*.

[6] Hasan M. Jamil and Gillian Dobbie. Logical characterization of multi-level secure databases. Technical report, Department of Computer Science, Mississippi State University, USA, October 2000.

[7] Hasan M. Jamil and L. V. S. Lakshmanan. A declarative semantics for behavioral inheritance and conflict resolution. In *Proc ILPS*, pages 130–144, December 1995.

[8] N. A. Jukic and S. V. Vrbsky. Asserting beliefs in MLS relational models. In *SIGMOD Record*, pages 30–35, Ithaca, NY, 1997.