Breaking the Atom with Samson

Jouko Väänänen*

Department of Mathematics and Statistics
University of Helsinki
and Institute for Logic, Language and Computation
University of Amsterdam

1 Dependence

The dependence atom $=(x, y)$ was introduced[1] in [11]. Here x and y are finite sets of attributes (or variables) and the intuitive meaning of $=(x, y)$ is that the attributes x completely (functionally) determine the attributes y. One may wonder, whether the dependence atom is truly an atom or whether it has further constituents. My very pleasant co-operation with Samson Abramsky led to the breaking of this atom, with hitherto unforeseen consequences. Here is the story.

A reasonable goal in logic is to capture the intuitive meaning of some concept by means of simple axioms. In the case of dependence atoms such simple axioms are the so-called Armstrong's Axioms[2]:

1. Reflexivity: $=(xy, x)$.
2. Augmentation: $=(x, y)$ implies $=(xz, yz)$.
3. Transitivity: If $=(x, y)$ and $=(y, z)$, then $=(x, z)$.

presented in one of the first[3] papers on database theory [3].

Armstrong's Axioms capture the meaning of dependence atoms completely in the sense that an atom $=(x, y)$ follows from a set Σ of other atoms by these rules if and only if every database in which the dependence atoms Σ hold also $=(x, y)$ holds.

A dependence atom holding in a database can be given the same meaning as a formula holding in a first order structure, but only if we make one very important leap. This is the leap from considering truth in one assignment to considering truth in a team, a set of assignments. This innovation is due to Hodges [9].

* Research partially supported by grant 251557 of the Academy of Finland. I am grateful to Pietro Galliani, Juliette Kennedy, Juha Kontinen and Fan Yang for reading the manuscript and making helpful comments.

[1] It was, however, known as "functional dependence" in database theory since the 70s.

[2] We write xy for the union $x \cup y$ of the sets x and y.

[3] According to R. Fagin in "Armstrong databases", 7th IBM Symposium on Mathematical Foundations of Computer Science, Kanagawa,Japan, May 1982.

B. Coecke, L. Ong, and P. Panangaden (Eds.): Abramsky Festschrift, LNCS 7860, pp. 327–335, 2013.

Let $\mathcal{M}$ be a background structure and X a set of assignments of variables into M. We call such sets *teams*. We define what it means for the team X to satisfy a dependence atom $=(x, y)$ in $\mathcal{M}$, denoted $\mathcal{M} \models_X =(x, y)$, as follows:

$$\forall s, s' \in X (s \upharpoonright x = s' \upharpoonright x \text{ implies } s \upharpoonright y = s' \upharpoonright y). \tag{1}$$

This gives exact meaning to $=(x, y)$ in perfect harmony with the idea that the values of x functionally determine the values of y. This is also the meaning of functional dependence as it started to appear in database theory after [3].

2 Constancy

A special case of $=(x, y)$ is the constancy atom $=(y)$ where $x = \emptyset$:

$$\forall s, s' \in X (s \upharpoonright y = s' \upharpoonright y). \tag{2}$$

The intuitive meaning of $=(y)$ is simply that y is constant. In a context like team semantics, where we have variation in the values of the attributes (or variables), it makes a lot of sense to take also into account the possibility of no variation. So in the context of team semantics, where formulas with free variables $x_1, \ldots, x_n$ are considered, the constancy atom

$$=(x_1 \ldots x_n) \tag{3}$$

limits the teams to singleton (or empty[4]) teams. In singleton (and empty) teams all dependence atoms $=(u, v)$ are true, so (3) has the effect of trivializing all dependence atoms.

A complete axiomatization of the logical consequence of a constancy atom from a set of other constancy atoms is almost too trivial to quote: it consists of just the rule

$$\text{Reflexivity: } =(xy) \text{ implies } =(x).$$

3 Dependence Logic

We can extend the definition of the meaning of dependence atoms to the entire first order logic built from identities $x = y$, relational atoms $R(x_1, \ldots, x_n)$ and the dependence atoms $=(x, y)$ as follows:

[4] Teams are sets of assignments and also the empty set is a team.

$$\begin{array}{lll}
\mathcal{M} \models_X x = y & \iff & \forall s \in X(s(x) = s(y)). \\
\mathcal{M} \models_X \neg x = y & \iff & \forall s \in X(s(x) \neq s(y)). \\
\mathcal{M} \models_X R(x_1, \ldots, x_n) & \iff & \forall s \in X((s(x_1), \ldots, s(x_n)) \in R^{\mathcal{M}}). \\
\mathcal{M} \models_X \neg R(x_1, \ldots, x_n) & \iff & \forall s \in X((s(x_1), \ldots, s(x_n)) \notin R^{\mathcal{M}}). \\
\mathcal{M} \models_X \phi \wedge \psi & \iff & \mathcal{M} \models_X \phi \text{ and } \mathcal{M} \models_X \psi. \\
\mathcal{M} \models_X \phi \vee \psi & \iff & \text{There are } X_1 \text{ and } X_2 \text{ such that} \\
& & X = X_1 \cup X_2,\ \mathcal{M} \models_{X_1} \phi, \text{ and } \mathcal{M} \models_{X_2} \psi. \\
\mathcal{M} \models_X \exists x \phi & \iff & \mathcal{M} \models_{X'} \phi \text{ for some } X' \text{ such that} \\
& & \forall s \in X\ \exists a \in M(s(a/x) \in X') \\
\mathcal{M} \models_X \forall x \phi & \iff & \mathcal{M} \models_{X'} \phi \text{ for some } X' \text{ such that} \\
& & \forall s \in X\ \forall a \in M(s(a/x) \in X')
\end{array}$$

We call the resulting semantically defined logic *Dependence Logic* [11].

Conceivably one could extend (1) to full dependence logic in different ways. An important guideline in making the choices for the above semantics is that for singleton teams $\{s\}$ this agrees with satisfaction in first order logic, that is, if we use the notation $\mathcal{M} \models_s \phi$ for the proposition that the assignment s satisfies the first order formula ϕ in $\mathcal{M}$, then for first order ϕ (i.e. for ϕ not containing dependence atoms):

$$\mathcal{M} \models_{\{s\}} \phi \iff \mathcal{M} \models_s \phi. \tag{4}$$

4 Downward Closure

Another guiding principle is *downward closure*: If $\mathcal{M} \models_X \phi$ and $Y \subseteq X$, then $\mathcal{M} \models_Y \phi$ for any dependence logic formula ϕ. Why do we want downward closure? The idea is that every dependence logic formula specifies a type of dependence. So, in particular, we do not aim at expressing non-dependence. Also, we do not consider dependencies which are manifested in part of the team only, even if the part was a very big part.

Our concept of dependence is thus *logical*, not *probabilistic*. For $=(x, y)$ to hold in X, every pair $\{s, s'\}$ chosen from X has to satisfy (1), not a single exception is allowed. This property is, of course, downward closed. We simply extend this to all formulas and thereby maintain the idea that every formula determines a weak form of this kind of dependence.

In practical applications probabilistic dependences are much more ubiquitous. In particular, in practical applications one can usually overlook a tiny portion of the team as irrelevant noise, possibly resulting from errors in data handling. In our mathematical theory of team semantics a single row can destroy the dependence manifested by millions of other rows.

Let us see how downward closure arises: Conjunction determines the simultaneity of two dependences. Downward closure is preserved. Disjunction says that the team splits into two subteams, both with their own dependence. Downward closure is preserved: a smaller team splits similarly into subteams obtained by intersecting the original subteams with the smaller team. The existential quantifier says that after some rows are updated, a dependence holds. A smaller team inherits the update canonically. Finally, the universal quantifier says that a given dependence holds even if a certain attribute has simultaneously all possible values. In a smaller team we simultaneously give all possible values to the given attribute in the remaining assignments. In each case downward closure is clearly preserved.

5 Axioms

Given that Armstrong's Axioms govern the dependence atom, what are the axioms governing the entire dependence logic? After all, we have just given the semantics. Ideally the semantics would reflect the completeness of the axioms. As it happens,[5] the above semantics does not reflect the completeness of *any* effectively given set of axioms and rules, because the set of Gödel numbers of valid sentences in dependence logic is a complete Π_2-set in the sense of the Levy hierarchy of set theory [12].

What *is* the meaning of the logical operations of dependence logic, if logical consequence cannot be axiomatized? A trivial answer is that the meaning comes from set theory according to the definition of the semantics. This, however, raises the further question, do we really have to understand set theory to understand the meaning of the logical operations $\wedge, \vee, \neg, \exists$ and $\forall$? Shouldn't "logical" mean something simpler than set theory?

Conceivably there is a fragment of dependence logic which is completely axiomatizable but still rich enough to express some interesting dependence properties. A step in this direction is [10], where a complete axiomatization of the logical consequence relation $\Sigma \models \phi$, where ϕ is first order, is given. Some of the rules of this axiomatization are quite involved but still all the rules have a clear intuitive content. Here is an example of the rules of [10]:

$$\frac{\exists \boldsymbol{y}(\bigwedge_{1\le j\le n} =(\boldsymbol{z}^j, y_j) \wedge C) \vee \exists \boldsymbol{y'}(\bigwedge_{n+1\le j\le n+m} =(\boldsymbol{z}^j, y_j) \wedge D)}{\exists \boldsymbol{y}\exists \boldsymbol{y'}(\bigwedge_{1\le j\le n+m} =(\boldsymbol{z}^j, y_j) \wedge (C \vee D))} \tag{5}$$

Work is underway to extend such results to non-first order—real dependence logic—consequences, and Juha Kontinen and his student Miika Hannula have unpublished results in this direction. In the light of this we may argue that there are meaningful and insightful steps between Armstrong's Axioms for atoms and the axiomatically intractable purely semantic theory of dependence.

6 Breaking the Atom

Are the complicated rules of [10], an example of which is (5), and the even more complicated ones needed for non-first order consequences, really the best way to understand the meaning of $=(x, y)$ and first order logic built on top of it? Maybe $=(x, y)$ can be analyzed in a different way, leading to simpler logical rules. Samson Abramsky suggested to look inside the atom $=(x, y)$ and see what

[5] This is essentially due to A. Ehrenfeucht, as Henkin reports in [8].

are its constituents. This led to the topic of the title of this paper, and to the paper [2].

To break the atom $=(x, y)$ we can rewrite its semantics as follows:

$$\forall Y \subseteq X(\text{if } x \text{ is constant on } Y, \text{ then } y \text{ is constant on } Y). \tag{6}$$

Using the constancy atoms this amounts to

$$\forall Y \subseteq X(\text{if } Y \text{ satisfies } =(x), \text{ then } Y \text{ satisfies } =(x)). \tag{7}$$

This resembles the semantics of intuitionistic implication in Kripke-semantics

$$w \Vdash \phi \to \psi \iff \forall u \geq w(\text{if } u \Vdash \phi, \text{ then } u \Vdash \psi),$$

so thinking of subsets of X as "extensions" of X we define a new logical operation:

$$\mathcal{M} \models \phi \to \psi \text{ iff } \forall Y \subseteq X(\text{if } \mathcal{M} \models_Y \phi, \text{ then } \mathcal{M} \models_Y \psi). \tag{8}$$

With this new implication we have a simple definition of the dependence atom:

$$=(x, y) \text{ iff } =(x) \to \ =(y) \tag{9}$$

with exactly the same semantics in team semantics as the original (1).

The idea that subteams are "extensions" of the team is not far-fetched. We can think of teams as uncertain information about an assignment (see [4] for more on this idea) and then a smaller team represents less uncertainty, i.e. more certainty. The ultimate extension in this sense is a singleton, representing total certainty about the assignment.

An obvious potential advantage of $=(x) \to \ =(y)$ over $=(x, y)$ is that on the one hand $=(y)$ is a much simpler atom than $=(x, y)$ and on the other hand $\to$ is not just an arbitrary new operation, it is the restriction to team semantics of the classical intuitionistic implication going back to Brouwer, Kolmogorov and Heyting, with an extensive literature about it.

Considering that $\phi \to \psi$ is the restriction of intuitionistic implication to the context of dependence logic, it can be hoped that it inherits some its rich meaning in constructive mathematics, and that this inheritance can be taken advantage of. Indeed, if Armstrong's Axioms are combined with (9) and dependence atoms are replaced by arbitrary formulas, Heyting's axioms for intuitionistic implication and conjunction arise:

1. Reflexivity: $(\phi \wedge \psi) \to \phi$.
2. Augmentation: $\phi \to \psi$ implies $(\phi \wedge \theta) \to (\psi \wedge \theta)$.
3. Transitivity: If $\phi \to \psi$ and $\psi \to \theta$, then $\phi \to \theta$.

This can be interpreted by saying that dependence logic has an intuitionistic element. It is not intuitionistic *per se*, but it shares some aspects with intuitionistic logic. Perhaps dependence logic could be developed completely constructively, but this has not been tried yet.

Another remarkable property of the intuitionistic implication in dependence logic is that it is the adjoint of conjunction, just as it should be:

$$\phi \wedge \psi \models \theta \text{ iff } \phi \models \psi \to \theta. \tag{10}$$

Probably the introduction of intuitionistic implication into dependence logic will eventually lead to better proof theory, not least because of the natural Galois connection (10). But alas, intuitionistic implication is not definable[6] in dependence logic! In fact Fan Yang [13] has shown that adding intuitionistic implication to dependence logic leads to full second order logic. So the introduction of the much needed implication to dependence logic leads to an explosion of the expressive power. Remarkably, we can still keep downward closure, so we have not introduced a negation in the classical sense, even though full second order logic is closed under negation. This is one of the peculiarities of team semantics, and its oddness disperses with closer investigation, for which we refer to [13].

7 Independence

Given that we have made some headway in understanding dependence by introducing the dependence atom and investigating its logic, the question naturally arises, what about independence? With this in mind, in [7] the independence atom $x \perp y$ was introduced[7].

Intuitively speaking, $x \perp y$ says that x and y are so independent of each other that knowing one gives no information about the other. This form of independence turns out to be ubiquitous among attributes in science and society, wherever independence is talked about. As it turned out in discussions with Samson, the independence concept of quantum mechanics in [1] is also of the type $x \perp y$. This observation is the subject of further study in co-operation with Samson.

To give independence exact meaning, let $\mathcal{M}$ be a background structure and X a set of assignments of variables into M. We define what it means for the team X to satisfy an independence atom $x \perp y$ in $\mathcal{M}$, denoted $\mathcal{M} \models_X x \perp y$, as follows:

$$\forall s, s' \in X \exists s''(s'' \upharpoonright x = s \upharpoonright x \text{ and } s'' \upharpoonright y = s' \upharpoonright y). \tag{11}$$

In other words, if a value a occurs in some assignment s as a value of x and a value b occurs as a value of y in some other assignment s', then there is a third assignment s'' which has simultaneously a as the value of x and b as the value of

[6] Pietro Galliani has a related but different, and very interesting, analysis of the dependence atom in terms of what he calls public announcement operators and the constancy atoms [5]. The public announcement operators have the advantage over $\to$ that they are definable in dependence logic itself.

[7] As with dependence atom, it turned out (this observation was made by Fredrik Engström) that our independence atom was already studied under a different name (embedded multivalued dependence) in database theory.

y. So from x being a we cannot infer what y is (unless it is constant), and from y being b we cannot infer what x is (unless it is constant).

Speaking of being constant, in fact, the constancy atom $=(x)$ implies $x \perp y$ because we can then choose $s'' = s'$ in (11). This is the curious state of affairs uncovered in [7], which shows that independence is not necessarily the opposite of dependence. Since $=(x)$ implies $=(x, y)$, we can have simultaneously $=(x, y)$ and $x \perp y$. Being constant is one form of independence.

The analogue of Armstrong's Axioms is in the case of independence atom the Geiger-Paz-Pearl [6] axioms:

1. Empty set rule: $x \perp \emptyset$.
2. Symmetry Rule: If $x \perp y$, then $y \perp x$.
3. Weakening Rule: If $x \perp yz$, then $x \perp y$.
4. Exchange Rule: If $x \perp y$ and $xy \perp z$, then $x \perp yz$.

These axioms satisfy in team semantics the same kind of Completeness Theorem[8] as Armstrong's Axioms. So we may regard them really as incorporating the essence of independence on the atomic level.

Independence atoms can be added to dependence logic[9] and we get a proper extension, called independence logic, which no longer satisfies the Downward Closure property. This logic is able to express existential second order properties in a particularly strong sense [5]. If we again add intuitionistic implication, we get full second order logic [13].

8 Speculation: Breaking the Independence Atom

Let us then try to break the independence atom into pieces. The reasons for attempting this are the same as in the case of dependence atom: the logic is non-axiomatizable and trying to axiomatize even just first order consequences[10] leads to rather complicated axioms.

Since we are bound to lose downward closure, intuitionistic implication alone is not enough. The following more complicated *compatible conjunction* suggests itself: We add a new logical connective $\phi \odot \psi$ to dependence logic with the following semantics:

$$\mathcal{M} \models_X \phi \odot \psi \iff$$

$$\forall_{\neq\emptyset} Y, Z \subseteq X((\mathcal{M} \models_Y \phi \text{ and } \mathcal{M} \models_Z \psi) \to$$
$$\exists Y', Z' \subseteq X(Y \subseteq Y', Z \subseteq Z', \mathcal{M} \models_{Y'} \phi, \mathcal{M} \models_{Z'} \psi, \text{ and } Y' \cap Z' \neq \emptyset)).$$

[8] Proved in [6] in the case of random variables.

[9] By an unpublished result of Pietro Galliani the dependence atom is definable from the independence atom, so if we add the independence atoms to first order logic, we get the dependence atoms free.

[10] Miika Hannula has a complete axiomatization (unpublished).

In words, every non-empty subteam Y satisfying ϕ and every non-empty subteam Z satisfying ψ, can be extended inside X, respectively, to Y' and Z' such that they still satisfy ϕ and, respectively, ψ, but, moreover, they meet. In a finite model this means that non-empty maximal teams satisfying ϕ and ψ meet. In finite models $\mathcal{M} \models_X \phi \odot \phi$ says the non-empty maximal subteams of X satisfying ϕ all meet. In forcing terms this means that below X the formula ϕ defines a set of compatible teams. In forcing terms $\phi \odot \psi$ is satisfied by teams below which ϕ and ψ are compatible. For sentences ϕ and ψ the sentence $\phi \odot \psi$ is always true. For first order $\phi(x)$ and $\psi(x)$:

$$\begin{aligned} \mathcal{M} \models \forall x(\phi(x) \odot \psi(x)) \iff & \mathcal{M} \not\models \exists x \phi(x) \text{ or} \\ & \mathcal{M} \not\models \exists x \psi(x) \text{ or else} \\ & \mathcal{M} \models \exists x(\phi(x) \wedge \psi(x)). \end{aligned}$$

Having added the new operation we can now break the independence atom into smaller constituents:

$$x \perp y \iff {=}(x) \odot {=}(y). \tag{12}$$

To what avail? In what sense is $=(x) \odot =(y)$ simpler than $x \perp y$? At the moment it is not clear whether the equivalence (12) is an insightful analysis of $x \perp y$. Certainly the atoms $=(x)$ represent a simplification from $x \perp y$, but it is more difficult to estimate the connective $\odot$. It is not one of the logical operations known in logic, and no general theory of $\odot$ exists.

References

1. Abramsky, S.: Relational hidden variables and non-locality. Studia Logica, arXiv:1007.2754 (2013), doi: 10.1007/s11225-013-9477-4
2. Abramsky, S., Väänänen, J.: From IF to BI: a tale of dependence and separation. Synthese 167(2, Knowledge, Rationality & Action), 207–230 (2009)
3. Armstrong, W.W.: Dependency structures of data base relationships. In: IFIP Congress, pp. 580–583 (1974)
4. Galliani, P.: The dynamics of imperfect information. Doctoral Thesis, University of Amsterdam (2012)
5. Galliani, P.: Inclusion and exclusion dependencies in team semantics—on some logics of imperfect information. Ann. Pure Appl. Logic 163(1), 68–84 (2012)
6. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inform. and Comput. 91(1), 128–141 (1991)
7. Grädel, E., Väänänen, J.: Dependence and independence. Studia Logica, arXiv:1208.5268 (2013), doi: 10.1007/s11225-013-9479-2
8. Henkin, L.: Some remarks on infinitely long formulas. In: Infinitistic Methods (Proc. Sympos. Foundations of Math., Warsaw, 1959), pp. 167–183. Pergamon, Oxford (1961)
9. Hodges, W.: Some strange quantifiers. In: Mycielski, J., Rozenberg, G., Salomaa, A. (eds.) Structures in Logic and Computer Science. LNCS, vol. 1261, pp. 51–65. Springer, Heidelberg (1997)

10. Kontinen, J., Väänänen, J.: Axiomatizing first order consequences in dependence logic. Annals of Pure and Applied Logic (to appear)
11. Väänänen, J.: Dependence logic. London Mathematical Society Student Texts, vol. 70. Cambridge University Press, Cambridge (2007)
12. Väänänen, J.A.: Second-order logic and foundations of mathematics. Bulletin of Symbolic Logic 7(4), 504–520 (2001)
13. Yang, F.: Expressing second-order sentences in intuitionistic dependence logic. Studia Logica, arXiv:1302.2279 (2013), doi: 10.1007/s11225-013-9476-5