Pointer Semantics with Forward Propagation

Sujata Ghosh[*]
Center for Soft Computing Research
Indian Statistical Institute
Kolkata, West Bengal, India

Benedikt Löwe[†]
Institute for Logic,
Language and Computation
Universiteit van Amsterdam
1018 TV Amsterdam, The Netherlands

Sanchit Saraf[‡]
Department of Mathematics and Statistics
Indian Institute of Technology
Kanpur 208016, India

Abstract

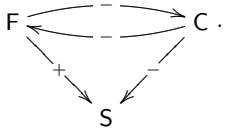
Pointer semantics describing a formal language with the possibility of self-reference have been invented by Haim Gaifman; they form a fundamental way of understanding the semantics of logic programming, but have also been used extensively in philosophical logic and other applications of logic. In pointer semantics, truth values flow backwards along from the defining statement to the propositional variable. As a consequence, pointer semantics cannot deal properly with dependence networks that have terminal nodes. Ghosh, Löwe and Scorelle have proposed an abstract system of combining pointer semantics with forward flow of truth values. Their system was difficult to handle, and apart from the fact that the system could handle some enlightening examples, very few theoretical insights were made. In this paper, we now produce a more concrete variant of this system, built on three-valued logic which allows us to gain more theoretical control over its properties.

Introduction

Pointer semantics were invented by Haim Gaifman (Gai88; Gai92) as a formal propositional language for finitely many propositions $\{\mathrm{p}_0, ..., \mathrm{p}_n\}$ defined in terms of each other. The language of pointer semantics is closely related to logic programming and is the logical reflection of the "Revision Theory of Truth" (Her82; GB93). Phenomena such as self-reference can be studied in pointer semantics as has been done in detail in Bolander's PhD thesis (Bol03) in a graph-theoretic setting where the dependence of a proposition p_i on a proposition p_j is represented by an edge from i to j in the *dependency graph*. The revision rules of pointer semantics let the truth value of p_j affect the truth value of p_i; we can say that truth values "flow backwards in the dependency graph".

In (Löw06), the second author proposed to apply the ideas of pointer semantics to a belief framework, interpreting the propositions not just as being defined in terms of each other, but at the same time being sources of information. As soon as you see propositions as sources of information, the pointer semantics model with its backward flow of truth values is not adequate anymore. The paradigmatic example for this is the following situation:

> "Suppose a reasoning agent is sitting in an office without windows. Next to him is his colleague, also located in the office without windows; the agent is simultaneously talking to his friend on the phone who is sitting in a street café.
>
> *Friend: Everything your colleague says is false; the sun is shining!*
>
> *Colleague: Everything your friend says is false; it is raining!*
>
> This situation can be described by the following graph:



> As in the *Nested Liars*, there are two consistent truth value assignments, but the context makes sure that one of them is intuitively preferred, as the agent's friend has first hand experience of the weather in the street café." (GLS07, p. 402–403)

Even if you assume that the agent trusts F, but initially believes in S (i.e., we would expect a belief revision), Gaifman pointer semantics will never change the value of S, as truth values only propagate backwards, and thus the values of terminal nodes in the dependency graph will never be revised.

In the context of belief, this is not a realistic feature: our trust in F, who tells us that S is true, should influence us to change our initial belief about S. Formally, we would need *forward propagation* of truth values along the dependency graph.

This idea has been implemented in a formal system in (GLS07), but the system proposed by the authors (using the interval $[-1, 1]$ as the truth values and a relatively complicated numerical function to combine the backward and forward influences) did not shed much light into divulging the intricate properties of belief change.

[*] *Additional affiliation:* Department of Mathematics, Visva-Bharati, Santiniketan, India.

[†] *Additional affiliations:* Department Mathematik, Universität Hamburg, Hamburg, Germany; Mathematisches Institut, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany.

[‡] The third author would like to thank the Institute for Logic, Language and Computation (ILLC) of the *Universiteit van Amsterdam* and the *Department Mathematik* of the *Universität Hamburg* for their hospitality during his visit from May to July 2008. All three authors would like to thank Bjarni Hilmarsson (Amsterdam) for programming support.

In this paper, we propose a more translucent system and discuss some of its properties. In the section "Definitions", we give the basic definitions, building on the formal system from (Löw06; GLS07), introducing an abstract algebra of pointer systems, and proving that this abstract algebra has logical properties if you restrict your system to backward propagation ("B-operators"). We then extend the concept of pointer semantics to include *forward propagation* in our section "Belief Semantics with Forward Propagation". In the section "Properties of our Belief Semantics" we test our system in an example originally used in (GLS07) and finally see that our system is ostensibly non-logical. However, this should not come as a shock, as the system is intended to model systems of belief:

> "The fact that the logic of belief, even rational belief, does not meet principles of truth-functional deductive logic, should no longer surprise us (Gol75, p. 6)."

Leaving the empirical study of comparing this system with our intuitions for future endeavors, we focus on the test case mentioned above to show non-trivial properties of the system, with some concluding remarks.

Definitions

Abstract Pointer Semantics

Fix a finite set of propositional variables $\{\mathrm{p}_0, ..., \mathrm{p}_n\}$. An **expression** is just a propositional formula using $\wedge$, $\vee$, and $\neg$ and some of the propositional variables or the empty sequence, denoted by $\lrcorner$.

We fix a finite algebra of truth values $\mathbb{T}$ with operations $\wedge$, $\vee$ and $\neg$ corresponding to the syntactic symbols. We assume a notion of order corresponding to information content that gives rise to a notion of **infimum** in the algebra of truth values, allowing to form $\inf(X)$ for some subset of $X \subseteq \mathbb{T}$. A truth value will represent the lowest information content (i.e., a least element in the given order); this truth value will be denoted by ½. We allow $\inf$ to be applied to the empty set and let $\inf \varnothing := ½$.

Our salient example is the algebra $\mathbb{T} := \{0, ½, 1\}$ with the following operations ("strong Kleene"):

$\wedge$	0	½	1
0	0	0	0
½	0	½	½
1	0	½	1

$\vee$	0	½	1
0	0	½	1
½	½	½	1
1	1	1	1

$\neg$	
0	1
½	½
1	0

The value ½ stands for ignorance, and thus the infimum is defined as $\inf(\{t\}) := t$, $\inf(\{½\} \cup X) := ½$, $\inf(\{0, 1\}) := ½$. This algebra of truth values will be used in this paper, even though the set-up in this section is more general.

If E is an expression and p_i is one of the propositional variables, then $\mathrm{p}_i \leftarrow E$ is a **clause**. We intuitively interpret $\mathrm{p}_i \leftarrow E$ as "p_i states E". If $E_0, ..., E_n$ are expressions, a set of clauses $\Sigma := \{\mathrm{p}_0 \leftarrow E_0, ..., \mathrm{p}_n \leftarrow E_n\}$ is called a **pointer system**. An **interpretation** is a function $I\colon \{\mathrm{p}_0, ..., \mathrm{p}_n\} \to \mathbb{T}$ assigning truth values to propositional letters. Note that if $\mathbb{T}$ is finite, the set of interpretations is a finite set (we shall use this later). A given interpretation I can be naturally extended to a function assigning truth values to all expressions (using the operations $\wedge$, $\vee$ and $\neg$ on $\mathbb{T}$). We denote this extended function with the same symbol I.

A clause can be transformed into an equation in $\mathbb{T}$: if $\mathrm{p}_i \leftarrow E$ is a clause, we can read it as an equation $\mathrm{p}_i = E$ in $\mathbb{T}$. If $\mathtt{Q}$ is such an equation, we say that an interpretation I is a **solution** of $\mathtt{Q}$ if plugging the values $\{I(\mathrm{p}_0), ..., I(\mathrm{p}_n)\}$ into the corresponding variables of the equation results in the same value left and right of the equals sign. An interpretation is a solution of a set of equations if it is a solution of each equation in the set.

A function mapping interpretations to interpretations is called a **revision function**; a family of these functions indexed by pointer systems is called a **revision operator**. If δ is a revision operator, we write δ_Σ for the revision function assigned to the pointer system Σ (and sometimes just write δ if Σ is clear from the context). We use the usual notation for iteration of revision functions, i.e., $\delta^0(I) := I$, $\delta^{n+1}(I) := \delta(\delta^n(I))$.

Given a pointer system $\{\mathrm{p}_0 \leftarrow E_0, ..., \mathrm{p}_n \leftarrow E_n\}$, we define its **dependency graph** by letting $\{0, ..., n\}$ be the vertices and allowing an edge from i to j if p_j occurs in E_i. Given a proposition p_i, arrows point to i from the propositions occurring in E_i, and thus we call a revision operator δ an **B-operator** (for "backward") if the value of $\delta(I)(\mathrm{p}_i)$ only depends on the values of $I(\mathrm{p}_j)$ for p_j occurring in E_i.

Fix Σ and δ. We call an interpretation I (Σ, δ)**-stable** if there is some k such that for all $n \geq k$, $\delta^n(I) = I$. We call I (Σ, δ)**-recurring** if for every k there is a $n \geq k$ such that $\delta^n(I) = I$.[1] If Σ is fixed by the context, we drop it from the notation and call interpretations δ**-stable** and δ**-recurring**. If H is an interpretation, we consider the sequence $H^\infty := \{\delta^i(H)\,;\, i \in \mathbb{N}\}$ of interpretations occurring in the infinite iteration of δ on H. Clearly, if there is a stable interpretation in H^∞, then this is the only recurring interpretation in H^∞. We write $\mathrm{Rec}_{\Sigma,\delta}(H)$ for the set of recurring interpretations in H^∞. Note that since the set of interpretations in finite, this set must be non-empty. If $I \in \mathrm{Rec}_{\Sigma,\delta}(H)$, then there are $i, j > 0$ such that $I = \delta^i(H) = \delta^{i+j}(H)$. Then for every $k < j$ and every n, we have $\delta^{i+k} = \delta^{i+n\cdot j+k}(H)$, so the sequence H^∞ exhibits a periodicity of length j (or a divisor of j). After the first occurrance of an $I \in \mathrm{Rec}_{\Sigma,\delta}(H)$, all further elements of H^∞ are recurring as well, and in particular, there is a recurring J such that $\delta(J) = I$. We shall call this an I**-predecessor** and will use this fact in our proofs.

Finally, we define our semantics and let

$$[\![\Sigma, \mathrm{p}_i]\!]_{\delta,H} := \inf\{I(\mathrm{p}_i)\,;\, I \in \mathrm{Rec}_{\Sigma,\delta}(H)\}, \text{ and}$$

$$[\![\Sigma, \mathrm{p}_i]\!]_{\delta} := \inf\{I(\mathrm{p}_i)\,;\, \exists H(I \in \mathrm{Rec}_{\Sigma,\delta}(H))\}.$$

An algebra of pointer systems

In the language of abstract pointer systems, the possibility of complicated referential structures means that the individual proposition cannot be evaluated without its context.

[1] We are ignoring here the additional complications that might arise if the initial hypothesis oscillates infinitely many times but stabilizes after a limit ordinal. For more on this, cf. (GB93; Her82; Löw06).

As a consequence, the natural notion of logical operations is not that between propositions, but that between systems. In this section, we define conjunction, disjunction and negation of statements in pointer systems. The definitions are straightforward, but very little systematic work has been done.

If $\Sigma = \{\mathrm{p}_0 \leftarrow E_0, ..., \mathrm{p}_n \leftarrow E_n\}$ is a pointer system and $0 \leq i \leq n$, we define a pointer system that corresponds to the negation of p_i with one additional propositional variable $\mathrm{p}_\neg$,

$$\neg(\Sigma, \mathrm{p}_i) := \Sigma \cup \{\mathrm{p}_\neg \leftarrow \neg \mathrm{p}_i\}.$$

If we have two pointer systems

$$\Sigma_0 = \{\mathrm{p}_0 \leftarrow E_{0,0}, ..., \mathrm{p}_{n_0} \leftarrow E_{0,n_0}\}, \text{ and}$$

$$\Sigma_1 = \{\mathrm{p}_0 \leftarrow E_{1,0}, ..., \mathrm{p}_{n_1} \leftarrow E_{1,n_1}\},$$

we make their sets of propositions disjoint by considering a set $\{\mathrm{p}_0, ..., \mathrm{p}_{n_0}, \mathrm{p}^*_0, ..., \mathrm{p}^*_{n_1}, \mathrm{p}_*\}$ of $n_0 + n_1 + 2$ propositional variables. We then set

$$\Sigma^*_1 := \{\mathrm{p}^*_0 \leftarrow E_{1,0}, ..., \mathrm{p}^*_{n_1} \leftarrow E_{1,n_1}\}.$$

With this, we can now define two new pointer systems (with an additional propositional variable p_*):

$$(\Sigma_0, \mathrm{p}_i) \wedge (\Sigma_1, \mathrm{p}_j) := \Sigma_0 \cup \Sigma^*_1 \cup \{\mathrm{p}_* \leftarrow \mathrm{p}_i \wedge \mathrm{p}^*_j\},$$

$$(\Sigma_0, \mathrm{p}_i) \vee (\Sigma_1, \mathrm{p}_j) := \Sigma_0 \cup \Sigma^*_1 \cup \{\mathrm{p}_* \leftarrow \mathrm{p}_i \vee \mathrm{p}^*_j\}.$$

Logical properties of Gaifman pointer semantics

Fix a system $\Sigma = \{\mathrm{p}_0 \leftarrow E_0, ..., \mathrm{p}_n \leftarrow E_n\}$. A proposition p_i is called a **terminal node** if $E_i = \llcorner$; it is called a **source node** if p_i does not occur in any of the expressions $E_0, ..., E_n$. This corresponds directly to the properties of i in the dependency graph: p_i is a terminal node if and only if i has no outgoing edges in the dependency graph, and it is a source node if and only if i has no incoming edges in the dependency graph.

The **Gaifman-Tarski operator** δ_{B} is defined as follows:

$$\delta_{\mathrm{B}}(I)(\mathrm{p}_i) := \begin{cases} I(E_i) & \text{if } \mathrm{p}_i \text{ is not terminal,} \\ I(\mathrm{p}_i) & \text{if } \mathrm{p}_i \text{ is terminal.} \end{cases}$$

Note that this operator can be described as follows:

> "From the clause $\mathrm{p}_i \leftarrow E_i$ form the equation $\mathtt{Q}_i$ by replacing the occurrences of p_i on the right-hand side of the equality sign with the values $I(\mathrm{p}_i)$. If p_i is a terminal node, let $\delta(I)(\mathrm{p}_i) := I(\mathrm{p}_i)$. Otherwise, let I^* be the unique solution to the system of equations $\{\mathtt{Q}_0, ..., \mathtt{Q}_n\}$ and let $\delta(I)(\mathrm{p}_i) := I^*(\mathrm{p}_i)$." (*)

This more complicated description will provide the motivation for the forward propagation operator δ_{F} in the section "Belief semantics with forward propagation".

The operator δ_{B} gives rise to a *logical* system, as the semantics defined by δ_{B} are compatible with the operations in the algebra of pointer systems.

Proposition 1 *Let $\Sigma = \{\mathrm{p}_0 \leftarrow E_0, ..., \mathrm{p}_n \leftarrow E_n\}$ be a pointer system. For any $i \leq n$, we have*

$$[\![\neg(\Sigma, \mathrm{p}_i)]\!]_{\delta_{\mathrm{B}}} = \neg [\![\Sigma, \mathrm{p}_i]\!]_{\delta_{\mathrm{B}}}.$$

Proof. In this proof, we shall denote interpretations for the set $\{\mathrm{p}_0, ..., \mathrm{p}_n\}$ by capital letters I and J and interpretations for the bigger set $\{\mathrm{p}_0, ..., \mathrm{p}_n, \mathrm{p}_\neg\}$ by letters $\hat{I}$ and $\hat{J}$. It is enough to show that if $\hat{I}$ is δ_{B}-recurring, then there is some δ_{B}-recurring J such that $\hat{I}(\mathrm{p}_\neg) = \neg J(\mathrm{p}_i)$. If I is δ_{B}-recurring, we call J an **I-predecessor** if J is also δ_{B}-recurring and $\delta_{\mathrm{B}}(J) = I$, and similarly for $\hat{I}$. It is easy to see that every δ_{B}-recurring I (or $\hat{I}$) has an I-predecessor (or $\hat{I}$-predecessor) which is not necessarily unique.

As δ_{B} is a B-operator, we have that if $\hat{J}$ is δ_{B}-recurring, then so is $J := \hat{J} \restriction \{\mathrm{p}_0, ..., \mathrm{p}_n\}$.

Now let $\hat{I}$ be δ_{B}-recurring and let $\hat{J}$ be one of its $\hat{I}$-predecessors. Then by the above, $J := \hat{J} \restriction \{\mathrm{p}_0, ..., \mathrm{p}_n\}$ is δ_{B}-recurring and

$$\hat{I}(\mathrm{p}_\neg) = \delta_{\mathrm{B}}(\hat{J})(\mathrm{p}_\neg) = \neg \hat{J}(\mathrm{p}_i) = \neg J(\mathrm{p}_i).$$

q.e.d.

Proposition 2 *Let $\Sigma_0 = \{\mathrm{p}_0 \leftarrow E_0, ..., \mathrm{p}_n \leftarrow E_n\}$ and $\Sigma_1 = \{\mathrm{p}_0 \leftarrow F_0, ..., \mathrm{p}_m \leftarrow F_m\}$ be pointer systems. For any $i, j \leq n$, we have*

$$[\![(\Sigma_0, \mathrm{p}_i) \vee (\Sigma_1, \mathrm{p}_j)]\!]_{\delta_{\mathrm{B}}} = [\![\Sigma_0, \mathrm{p}_i]\!]_{\delta_{\mathrm{B}}} \vee [\![\Sigma_1, \mathrm{p}_j]\!]_{\delta_{\mathrm{B}}}.$$

Similarly for $\vee$ replaced by $\wedge$.

Proof. The basic idea is very similar to the proof of Proposition 1, except that we have to be a bit more careful to see how the two systems Σ_0 and Σ_1 can interact in the bigger system. We reserve letters I_0 and J_0 for the interpretations on Σ_0, I_1 and J_1 for those on Σ_1 and I and J for interpretations on the whole system, including p_*. If $[\![\Sigma_0, \mathrm{p}_1]\!] = 1$, then any δ_{B}-recurring I must have $I(\mathrm{p}_*) = 1$ by the $\vee$-analogue of the argument given in the proof of Proposition 1. Similarly, for $[\![\Sigma_1, \mathrm{p}_j]\!] = 1$ and the case that $[\![\Sigma_0, \mathrm{p}_i]\!] = [\![\Sigma_1, \mathrm{p}_j]\!] = 0$. This takes care of six of the nine possible cases.

If I_0 and I_1 are δ_{B}-recurring, then so is the function $I := \delta(I_0) \cup \delta(I_1) \cup \{\langle \mathrm{p}_*, I_0(\mathrm{p}_i) \vee I_1(\mathrm{p}_j) \rangle\}$ (if I_0 is k-periodic and I_1 is ℓ-periodic, then I is at most $k \cdot \ell$-periodic). In particular, if we have such an I_0 with $I_0(\mathrm{p}_i) = ½$ and an I_1 with $I_1(\mathrm{p}_j) \neq 1$, then $I(\mathrm{p}_*) = ½$ (and symmetrically for interchanged rôles). Similarly, if we have recurring interpretations for relevant values 0 and 1 for both small systems, we can put them together to δ_{B}-recurring interpretations with values 0 and 1 for the big system. This gives the truth value ½ for the disjunction in the remaining three cases.

q.e.d.

Note that the proof does not really depend on the particular operator δ_{B}. Other B-operators, as long as the logical connectives are properly represented by our semantical rules for the revised values of $\mathrm{p}_\neg$ and p_*, will be fine here. This shows that δ_{B} interacts with our algebra of pointer systems to give rise to a logical system.

Belief semantics with forward propagation

In (GLS07), the authors gave a revision operator that incorporated both backward and forward propagation. The value of $\delta(I)(\mathrm{p}_i)$ depended on the values of all $I(\mathrm{p}_j)$ such that j is

H_0	$\delta_{\mathrm{B}}(H_0)$	$\delta_{\mathrm{F}}(H_0)$	H_1	$\delta_{\mathrm{B}}(H_1)$	$\delta_{\mathrm{F}}(H_1)$	H_2	$\delta_{\mathrm{B}}(H_2)$	$\delta_{\mathrm{F}}(H_2)$	H_3	$\delta_{\mathrm{B}}(H_3)$	$\delta_{\mathrm{F}}(H_3)$	H_4
0	½	0	0	1	0	½	1	½	1	1	1	1
1	1	½	1	1	½	1	1	½	1	1	½	1
0	0	½	0	0	½	0	0	½	0	0	½	0
½	0	½	0	0	½	0	0	½	0	0	0	0
½	½	½	½	1	½	1	1	1	1	1	1	1

Figure 1: The first three iterations of values of $H_0 = (0, 1, 0, ½, ½)$ up to the point of stability ($H_3 = (1, 1, 0, 0, 1)$).

connected to i in the dependency graph.[2] Here, we split the operator in two parts: the backward part which is identical to the Gaifman-Tarski operator, and the forward part which we shall define now.

In analogy to the definition of δ_{B}, we define δ_{F} as follows. Given an interpretation I, we transform each clause $\mathrm{p}_i \leftarrow E_i$ of the system into an equation $\mathtt{Q}_i \equiv I(\mathrm{p}_i) = E_i$ where the occurrences of the p_i on the left-hand side of the equation are replaced by their I-values and the ones on the right-hand side are variables. We obtain a system $\{\mathtt{Q}_0, ..., \mathtt{Q}_n\}$ of $n+1$ equations in $\mathbb{T}$. Note that we cannot mimic the definition of δ_{B} directly: as opposed to the equations in that definition, the system $\{\mathtt{Q}_0, ..., \mathtt{Q}_n\}$ need not have a solution, and if it has one, it need not be unique. We therefore define: if p_i is a source node, then $\delta_{\mathrm{F}}(I)(\mathrm{p}_i) := I(\mathrm{p}_i)$. Otherwise, let S be the set of solutions to the system of equations $\{\mathtt{Q}_0, ..., \mathtt{Q}_n\}$ and let $\delta_{\mathrm{F}}(I)(\mathrm{p}_i) := \inf\{I(\mathrm{p}_i)\,;\, I \in S\}$ (remember that $\inf \varnothing = ½$). Note that this definition is literally the dual to definition (*) of δ_{B} (i.e., it is obtained from (*) by interchanging "right-hand side" by "left-hand side" and "terminal node" by "source node").

We now combine δ_{B} and δ_{F} to one operator δ_{T} by defining pointwise

$$\delta_{\mathrm{T}}(I)(\mathrm{p}_i) := \delta_{\mathrm{F}}(I)(\mathrm{p}_i) \otimes \delta_{\mathrm{B}}(I)(\mathrm{p}_i)$$

where $\otimes$ has the following truth table:

$\otimes$	0	½	1
0	0	0	½
½	0	½	1
1	½	1	1

.

Let us briefly motivate this table. The values for agreement ($0 \otimes 0$, $½ \otimes ½$, and $1 \otimes 1$) are obvious choices. The two values for complete disagreement ($0 \otimes 1$ and $1 \otimes 0$) are also relatively clear: as long as you do not want to give one of the two directions primacy over the other, you have little choice but give the value ½ of ignorance. For reasons of symmetry, this leaves two values $½ \otimes 0 = 0 \otimes ½$ and $½ \otimes 1 = 1 \otimes ½$ to be decided. We opted here for the most informative truth table that gives classical values the benefit of the doubt. The other options would be the tables

$\otimes_0$	0	½	1
0	0	½	½
½	½	½	1
1	½	1	1

$\otimes_1$	0	½	1
0	0	0	½
½	0	½	½
1	½	½	1

$\otimes_2$	0	½	1
0	0	½	½
f½	½	½	½
1	½	½	1

.

[2] Actually, the system was even more complicated, as interpretations did not only provide values for the vertices, but also for the edges in the dependency graph.

Each of these connectives will give rise to a slightly different semantics. We opted for the first connective $\otimes$, as the semantics based on the other three seem to have a tendency to stabilize on the value ½ very often (the safe option: in case of confusion, opt for ignorance).

Properties of our belief semantics

As mentioned in the introduction, we should not be shocked to hear that a system modelling belief and belief change does not follow basic logical rules such as Propositions 1 and 2. From its beginnings (Hin62) to modern doxastic systems, logics of belief have always dealt with the peculiarities of the logical structure of belief. Let us take the particular example of conjunction: the fact that belief is not closed under the standard logical rules for conjunction is known as the *preface paradox* and has been described by Kyburg as "conjunctivitis" (Kyb70; HB99). While the phenomenon of "conjunctivitis" focusses on the fact that something strange happens with the belief of large conjunctions of individually believes propositions, in other contexts (that of the modality of "ensuring that"), we have a problem with simple binary conjunctions (Sch08).

Of course, the failure of certain logical rules in reasoning about belief is closely connected to the so-called "errors in reasoning" observed in experimental psychology, e.g., the famous Wason selection task (Was68). What constitutes rational belief in this context is an interesting question for modellers and philosophers alike (Ste97; Chr07; Cou08).

The fact that reasoning about beliefs is such a complicated topic makes it unlikely that there is one objectively accurate semantics that describes reasoning about beliefs in pointer structures. Instead, different semantics will be relevant in different situations. Which semantics to choose is a decision that the modeller has to make with as much information as possible. This is one of the reasons why we decided for a very lean and easy to survey system (as opposed to the more complicated system of (GLS07)). Let us focus on some concrete examples to validate our claim that the semantics we propose do agree with intuitive understanding, and thus serve as a quasi-empirical test for our system as a formalization of reasoning in self-referential situations with evidence.

Concrete examples

So far, we have just given an abstract system of belief flow in our pointer systems. In order to check whether our system results in intuitively plausible results, we have to check

H_0^*	$\delta_{\mathrm{B}}(H_0^*)$	$\delta_{\mathrm{F}}(H_0^*)$	H_1^*	$\delta_{\mathrm{B}}(H_1^*)$	$\delta_{\mathrm{F}}(H_1^*)$	H_2^*	$\delta_{\mathrm{B}}(H_2^*)$	$\delta_{\mathrm{F}}(H_2^*)$	H_3^*	$\delta_{\mathrm{B}}(H_3^*)$	$\delta_{\mathrm{F}}(H_3^*)$	H_4^*	$\delta_{\mathrm{B}}(H_4^*)$	$\delta_{\mathrm{F}}(H_4^*)$	H_5^*
0	½	0	0	1	0	½	1	½	1	1	1	1	1	1	1
1	1	½	1	1	½	1	1	½	1	1	½	1	1	½	1
0	0	½	0	0	½	0	0	½	0	0	½	0	0	½	0
½	0	½	0	0	½	0	0	½	0	0	0	0	0	0	0
0	½	½	½	½	½	½	1	½	1	1	½	1	1	1	1
1	1	½	1	1	½	1	1	½	1	1	½	1	1	½	1
0	0	½	0	0	½	0	0	½	0	0	½	0	0	½	0
½	0	1	½	0	½	0	0	½	0	0	0	0	0	0	0
½	½	½	½	½	½	½	½	½	½	1	½	1	1	1	1

Figure 2: The first three iterations of values of $H_0^* = (0, 1, 0, ½, 0, 1, 0, ½, ½)$ up to the point of stability ($H_4^* = (1, 1, 0, 0, 1, 1, 0, 0, 1)$).

a few examples. Keep in mind that our goal should be to model human reasoning behaviour in the presence of partially paradoxical situations. In this paper, we can only give a first attempt at testing the adequacy of our system: an empirical test against natural language intuitions on a much larger scale is needed. For this, also cf. our section "Discussion and Future Work".

The Liar As usual, the liar sentence is interpreted by the system $\Sigma := \{\mathrm{p}_0 \leftarrow \neg \mathrm{p}_0\}$. Since we have only one propositional variable, interpretations are just elements of $\mathbb{T} = \{0, ½, 1\}$. It is easy to see that $\delta_{\mathrm{B}}(0) = \delta_{\mathrm{F}}(0) = \delta_{\mathrm{T}}(0) = 1$, $\delta_{\mathrm{B}}(½) = \delta_{\mathrm{F}}(½) = \delta_{\mathrm{T}}(½) = ½$, and $\delta_{\mathrm{B}}(1) = \delta_{\mathrm{F}}(1) = \delta_{\mathrm{T}}(1) = 0$. This means that the δ_{T}-behaviour of the liar sentence is equal to the Gaifman-semantics behaviour.

The Miller-Jones Example Consider the following test example from (GLS07):

> Professors Jones, Miller and Smith are colleagues in a computer science department. Jones and Miller dislike each other without reservation and are very liberal in telling everyone else that "everything that the other one says is false". Smith just returned from a trip abroad and needs to find out about two committee meetings on Monday morning. He sends out e-mails to his colleagues and to the department secretary. He asks all three of them about the meeting of the faculty, and Jones and the secretary about the meeting of the library committee (of which Miller is not a member).
>
> Jones replies: *"We have the faculty meeting at 10am and the library committee meeting at 11am; by the way, don't believe anything that Miller says, as he is always wrong."*
>
> Miller replies: *"The faculty meeting was cancelled; by the way don't believe anything that Jones says, as he is always wrong."*
>
> The secretary replies: *"The faculty meeting is at 10 am and the library committee meeting is at 11 am. But I am sure that Professor Miller told you already as he is always such an accurate person and quick in answering e-mails: everything Miller says is correct."* (GLS07, p. 408)

Trying to analyse Smith's reasoning process after he returns from his trip, we can assume that he generally believes the secretary's opinions, and that he has no prior idea about the truth value of the statements "the faculty meeting is at 10am" and "the library meeting is at 11am" and the utterances of Miller and Jones. We have a vague intuition that tells us that in this hypothetical situation, Smith should at least come to the conclusion that the library meeting will be held at 11am (as there is positive, but no negative evidence). His beliefs about the faculty meeting are less straightforward, as there is some positive evidence, but also some negative evidence, and there is the confusing fact that the secretary supports Miller's statement despite the disagreement in truth value.

In (GLS07, p. 409), the authors analysed this example with their real-valued model and ended up with a stable solution in which Smith accepted both appointments and took Jones's side (disbelieving Miller). In our system, we now get the following analysis: A pointer system formulation is given as follows.

$$\begin{array}{lcl} \mathrm{p}_0 & \leftarrow & \neg \mathrm{p}_1 \wedge \neg \mathrm{p}_4, \\ \mathrm{p}_1 & \leftarrow & \neg \mathrm{p}_0 \wedge \mathrm{p}_2 \wedge \mathrm{p}_4, \\ \mathrm{p}_2 & \leftarrow & \text{␣}, \\ \mathrm{p}_3 & \leftarrow & \mathrm{p}_0 \wedge \mathrm{p}_2 \wedge \mathrm{p}_4, \\ \mathrm{p}_4 & \leftarrow & \text{␣}, \end{array}$$

where p_0 is Miller's utterance, p_1 is Jones's utterance, p_2 is "the library meeting will take place at 11am", p_3 is the secretary's utterance, and p_4 is the "the faculty meeting will take place at 10am".

We identify our starting hypothesis with $H := (½, ½, ½, 1, ½)$ (here, as usual, we identify an interpretation with its sequence of values in the order of the indices of the propositional letters). Then $\delta_{\mathrm{B}}(H) = (½, ½, ½, ½, ½)$ and $\delta_{\mathrm{F}}(H) = (½, ½, 1, 1, ½)$, so that we get $H' := \delta_{\mathrm{T}}(H) = (½, ½, 1, 1, ½)$. Then, in the second iteration step, $\delta_{\mathrm{B}}(H') = (½, ½, 1, ½, ½)$ and $\delta_{\mathrm{F}}(H') = (½, ½, 1, 1, ½)$, so we obtain stability at $\delta_{\mathrm{T}}(H') = H'$.

Examples of nonlogical behaviour

In what follows, we investigate some logical properties of the belief semantics, viz. negation and disjunction, focussing on stable hypotheses. To some extent, our results show that the operator δ_{T} is rather far from the logical properties of δ_{B} discussed in Propositions 1 and 2.

Negation Consider the pointer system Σ given by

$$\begin{array}{lcllcl} \mathrm{p}_0 & \leftarrow & \neg \mathrm{p}_3, & \mathrm{p}_1 & \leftarrow & \text{␣}, \\ \mathrm{p}_2 & \leftarrow & \text{␣}, & \mathrm{p}_3 & \leftarrow & \mathrm{p}_1 \wedge \mathrm{p}_2. \end{array}$$

The interpretation $H := (0, 1, 0, ½)$ is δ_{T}-stable, as $\delta_{\mathrm{B}}(H) = (½, 1, 0, 0)$, $\delta_{\mathrm{F}}(H) = (0, ½, ½, 1)$, and thus $\delta_{\mathrm{T}}(H) = H$.

Now let us consider the system $\neg(\Sigma, \mathrm{p}_3)$. Remember from the proof of Proposition 1 that stable interpretations for the small system could be extended to stable interpretations for the big system by plugging in the expected value for $\mathrm{p}_\neg$. So, in this particular case, the stable value for p_3 is ½, so we would expect that by extending H by $H_0(\mathrm{p}_\neg) := ½$, we would get another stable interpretation.

But this is not the case, as the table of iterated values given in Figure 1 shows. Note that H_0 is not even recurring.

Disjunction Consider the pointer systems Σ and Σ^* and their disjunction $(\Sigma, \mathrm{p}_4) \vee (\Sigma^*, \mathrm{p}_1^*)$ given as follows:

$$\begin{array}{lcllcl} \mathrm{p}_0 & \leftarrow & \neg\mathrm{p}_3, & \mathrm{p}_0^* & \leftarrow & \neg\mathrm{p}_3^*, \\ \mathrm{p}_1 & \leftarrow & \textvisiblespace & \mathrm{p}_1^*, & \leftarrow & \textvisiblespace, \\ \mathrm{p}_2 & \leftarrow & \textvisiblespace & \mathrm{p}_2^*, & \leftarrow & \textvisiblespace, \\ \mathrm{p}_3 & \leftarrow & \mathrm{p}_1 \wedge \mathrm{p}_2, & \mathrm{p}_3^* & \leftarrow & \mathrm{p}_1^* \wedge \mathrm{p}_2^*, \\ \mathrm{p}_* & \leftarrow & \mathrm{p}_4 \vee \mathrm{p}_1^*. & & & \end{array}$$

Note that Σ and Σ^* are the same system up to isomorphism and that Σ is the system from the previous example. We already know that the interpretation $H = (0, 1, 0, ½)$ is δ_{T}-stable (therefore, it is δ_{T}-stable for both Σ and Σ^* in the appropriate reading.

The natural extension of H to the full system with nine propositional variables would be $H_0^* := (0, 1, 0, ½, 0, 1, 0, ½, ½)$, as p_* should take the value $H(\mathrm{p}_4) \vee H(\mathrm{p}_1^*) = ½ \vee 0 = ½$. However, we see in Figure 2 that this interpretation is not stable (or even recurring).

Discussion and future work

Testing the behaviour of our system on the liar sentence and one additional example cannot be enough as an empirical test of the adequacy of our system. After testing more examples and having developed some theoretical insight into the system and its properties, we would consider testing the system experimentally by designing situations in which people reason about beliefs in self-referential situations with evidence, and then compare the predictions of our system to the actual behaviour of human agents.

Such an experimental test should not be done with just one system, but with a class of systems. We have already discussed that our choice of the connective $\otimes$ combining δ_{B} and δ_{F} to δ_{T} was not unique. Similarly, the rules for how to handle multiple solutions ("take the pointwise infimum") in the case of forward propagation are not the only way to deal with this formally. One natural alternative option would be to split the sequence H^∞ into multiple sequences if there are multiple solutions. For instance, if we are trying to calculate $\delta_{\mathrm{F}}(H)$ and we have multiple solutions to the set of equations, then $\delta_{\mathrm{F}}(H)$ becomes a set of interpretations (possibly giving rise to different recurrences and stabilities, depending on which possibility you follow). There are many variants that could be defined, but the final arbiter for whether these systems are adequate descriptions of reasoning processes will have to be the experimental test.

References

Thomas Bolander. *Logical Theories for Agent Introspection*. PhD thesis, Technical University of Denmark, 2003.

David Christensen. *Putting Logic in its place. Formal Constraints on Rational Belief*. Oxford University Press, 2007.

Marian E. Counihan. *Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning*. PhD thesis, Universiteit van Amsterdam, 2008. ILLC Publications DS-2008-10.

Haim Gaifman. Operational pointer semantics: Solution to self-referential puzzles I. In Moshe Y. Vardi, editor, *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, March 1988*, pages 43–59. Morgan Kaufmann, 1988.

Haim Gaifman. Pointers to truth. *Journal of Philosophy*, 89(5):223–261, 1992.

Anil Gupta and Nuel Belnap. *The revision theory of truth*. A Bradford Book. MIT Press, Cambridge, MA, 1993.

Sujata Ghosh, Benedikt Löwe, and Erik Scorelle. Belief flow in assertion networks. In Uta Priss, Simon Polovina, and Richard Hill, editors, *Conceptual Structures: Knowledge Architectures for Smart Applications, 15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, July 22-27, 2007, Proceedings*, volume 4604 of *Lecture Notes in Computer Science*, pages 401–414. Springer, 2007.

Alan H. Goldman. A note on the conjunctivity of knowledge. *Analysis*, 36:5–9, 1975.

James Hawthorne and Luc Bovens. The preface, the lottery, and the logic of belief. *Mind*, 108:241–264, 1999.

Hans G. Herzberger. Notes on naive semantics. *Journal of Philosophical Logic*, 11(1):61–102, 1982.

Jaakko Hintikka. *Knowledge and Belief*. Cornell, 1962.

Henry Kyburg. Conjunctivitis. In Marshall Swain, editor, *Induction, Acceptance, and Rational Belief*, page 5582. Reidel, 1970.

Benedikt Löwe. Revision forever! In Henrik Schärfe, Pascal Hitzler, and Peter Øhrstrøm, editors, *Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures, ICCS 2006, Aalborg, Denmark, July 16-21, 2006, Proceedings*, volume 4068 of *Lecture Notes in Computer Science*, pages 22–36. Springer, 2006.

Benjamin Schnieder. On what we can ensure. *Synthese*, 162:101–115, 2008.

Edward Stein. *Without good reason. The rationality debate in philosophy and cognitive science*. Clarendon Library of Logic and Philosophy. Clarendon Press, 1997.

Peter Wason. Reasoning about a rule. *Quarterly Journal of Experimental Psychology*, 20(3):273–281, 1968.