Pointer semantics with forward propagation
Ghosh, S.; Löwe, B.; Saraf, S.

Published in:
Advances in Intelligent Systems Research

DOI:
10.2991/agi.2009.29

Citation for published version (APA):
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
In this paper, we will discuss a new approach to formally modelling belief change in systems of sentences with inter-dependency. Our approach is based on the paradigm called pointer semantics or revision theory which forms a fundamental way of successfully understanding the semantics of logic programming, but has also been used extensively in philosophical logic and other applications of logic. With a purely unidirectional (backward) flow of change, pointer semantics are not fit to deal with belief change. We propose an extension that allows flow of change in both directions in order to be applied for belief change.

Introduction

Pointer semantics

Pointer semantics are a formal propositional language for finitely many propositions \( \{p_0, \ldots, p_n\} \) defined in terms of each other, and are a well-established tool in logic. The underlying idea of pointer semantics is to

\[
\text{iterate away uncertainties and keep the values that stabilize}, \quad (#)
\]

Starting with an initial hypothesis for the truth values that may be in conflict with each other, you apply the Tarskian definition of truth repeatedly and consider the iteration sequence. Those values that stabilize will be kept, the others discarded. The semantics based on (\#) have been rediscovered several times independently and have found applications in various areas of foundational study and philosophy.\(^1\)

The language of pointer semantics is closely related to logic programming and is the logical reflection of the “Revision Theory of Truth” (GB93). The version of pointer semantics that we shall be using in this paper is essentially that of Haim Gaifman (Gai88; Gai92). The revision rules of pointer semantics let the truth value of a proposition depend on the values of those propositions it points to in the dependency graph (cf. Definition 1), so truth values “flow backward in the dependency graph”.

Pointer semantics for belief systems

It has been suggested several times (e.g., cf. (Löw06)) that the underlying principle (\#) could be used for other processes involving revision, e.g., the process of revising and updating our beliefs in the face of learning new and unexpected facts. But the mentioned “backward flow” of traditional pointer semantics makes sure that truth values of terminal nodes in the dependency graph (which would typically correspond to those newly learned atomic statements) will never change in the revision process. In the context of belief, 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 was not easy to handle.

Aims, motivation and related work

Our aim is to provide a new framework for a semantics of belief change based on the general principle (\#) using the standard definition of revision semantics (Definition 2). In this paper, we cannot give a motivation of the general framework of revision theory and refer the reader to the extensive philosophical discussion in (GB93) or the survey (Löw06). Our aim is to stay as close as possible to the spirit of that framework.

There are many competing formal frameworks that deal with the question “How can we model rational belief change?”, most of which far more developed than what can be outlined in this short note. Transition systems and action description languages have been used in (HD05; HD07); there is a rich literature on using probabilistic logics for modelling belief change; in machine learning, dependency networks, Bayesian networks and Markov logic networks have been used (KDR07; HCH03; KD05); an argumentation theoretic approach can be found in (CnS07). This short list only scratches the surface of the vast literature of very good, intuitive and successful answers to our general...
question. Our approach of staying as close as possible to the
pointer semantics paradigm of (6) cannot yet compete
at the same level of depth and maturity at the moment. So,
we need to explain why one should be interested in a new
approach to belief change based on pointer semantics:

Traditional logic approaches to belief update are at the
level of axiomatics of what we require of a belief change
function, not at the more detailed level of how to actually
make the decisions about the adequate belief change. For
instance, if an agent believes \( p \rightarrow q, p \) and learns \( \{ \lnot q \} \),
then the axiomatic prediction would be that either \( p \rightarrow q \) or
\( p \) has to be dropped, but without further details, it is difficult
to predict which one.

As discussed in (Löw06), pointer semantics carries a lot
of information transcending the pure definition of the se-
manics (Definition 2): you can look at how fast stable val-
ues stabilize, at the various oscillation patterns of oscillating
hypotheses, etc. This information can be used for definitions
of levels or types of stability in order to help with priori-
tizing, promising to provide new insight in possible belief
revision operators. These are tools that might be applica-
directly in modelling belief change, or could serve as a
subsidary tool to support other (established) systems of
formal belief change models for applications in artificial in-
telligence.

Overview of this paper

In our section “Definitions”, we give the standard Definition
2 from pointer semantics (following (Löw06)) and define an
algebra of pointer systems. The latter definition is new to
this paper, allowing to state and prove Propositions 3 and 4
in the case of operators restricted to backward propagation.
The central new definition of this paper is in the section “Be-
lief 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 as expected for a sys-
tem is intended to model systems of belief. We close with
a discussion of future work in our final section.

Definitions

Abstract Pointer Semantics

Fix a finite set of propositional variables \( \{ p_0, \ldots, p_n \} \). An
expression is just a propositional formula using \( \wedge, \vee, \text{ and } \lnot \)
and some of the propositional variables or the empty se-
quence, denoted by \( \bot \).

We fix a finite algebra of truth values \( \mathbb{T} \) with operations
\( \wedge, \vee, \text{ and } \lnot \) corresponding to the syntactic symbols. We as-
sume 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 \( \frac{1}{2} \). We allow \( \bot \) to be applied to the empty
set and let \( \inf(\emptyset) := \frac{1}{2} \).

Our salient example is the algebra \( \mathbb{T} := \{ 0, \frac{1}{2}, 1 \} \) with
the following operations (“strong Kleene”):

| \( \land \) | 0 | 1/2 | 1 |
| \( \lor \) | 0 | 1/2 | 1 |
| \( \lnot \) | 1 | 1/2 | 0 |

The value \( \frac{1}{2} \) stands for ignorance, and thus the infimum is
defined as \( \inf(\{t\}) := t, \inf(\{\frac{1}{2}\} \cup X) := \frac{1}{2}, \inf(\{0,1\}) := \frac{1}{2} \).
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 \( p_i \models E \) is a clause. We intuitively interpret
\( p_i \models E \) as “\( p_i \) states \( E \)”. If \( E_0, \ldots, E_n \) are expressions, a set
of clauses \( \Sigma := \{ p_0 \models E_0, \ldots, p_n \models E_n \} \) is called a pointer
system. An interpretation is a function \( I : \{ p_0, \ldots, p_n \} \rightarrow \mathbb{T} \)
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 ex-
tended to a function assigning truth values to all expressions
(using the operations \( \land, \lor \text{ and } \lnot \) on \( \mathbb{T} \)). We denote this ex-
tended function with the same symbol \( I \).

A clause can be transformed into an equation in \( \mathbb{T} \): if
\( p_i \models E \) is a clause, we can read it as an equation \( p_i = E \)
in \( \mathbb{T} \). If \( q \) is such an equation, we say that an interpretation \( I \)
is a solution of \( q \) if plugging the values \( \{ I(p_0), \ldots, I(p_n) \} \)
into the corresponding variables of the equation results in
the same value left and right of the equals sign. An inter-
pretation 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 in-
dexed by pointer systems is called a revision operator. If
\( \delta \) is a revision operator, we write \( \delta I \) for the revision function
assigned to the pointer system \( \Sigma \) (and sometimes just
write \( \delta \) if \( \Sigma \) is clear from the context). We use the usual
notation for iteration of revision functions, i.e., \( \delta^0(I) := I, \delta^{n+1}(I) := \delta(\delta^n(I)) \).

Definition 1 Given a pointer system \( \{ p_0 \models E_0, \ldots, p_n \models E_n \} \),
we define its dependency graph by letting \( \{ 0, \ldots, n \} \) 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 pro-
positions occurring in \( E_i \), and thus we call a revision operator
\( \delta \) an \textbf{B-operator} (for “backward”) if the value of \( I(p_i) \) only
depends on the values of \( I(p_j) \) for \( p_j \) occurring in \( E_i \).

Fix \( \Sigma \) and \( \delta \). We call an interpretation \( I \) (\( \Sigma, \delta \))-stable
if there is some \( k \) such that for all \( n \geq k, \delta^n(I) = I \). We call
\( I(\Sigma, \delta) \)-recurring if for every \( k \) there is a \( n \geq k \) such that
\( \delta^n(I) = I \). If \( \Sigma \) is fixed by the context, we drop it from the
notation and call interpretations \( \delta \)-stable and \( \delta \)-recurring.

If \( H \) is an interpretation, we consider the sequence \( H^\omega := \{ \delta^i(H) : i \in \mathbb{N} \} \) of interpretations occurring in
the infinite iteration of \( \delta \) on \( H \). Clearly, if there is a stable interpretation in \( H^\omega \), then this is the only recurring interpretation in \( H^\omega \).

We write \( \text{Rec}_{\Sigma, \delta}(H) \) for the set of recurring interpretations

There the relationship between pointer semantics and the de-
dependency graph has been investigated in (BoI03).
in $H^\infty$. Note that since the set of interpretations in finite, this set must be non-empty. If $I \in \text{Rec}_{\Sigma, \delta}(H)$, then there are $i, j > 0$ such that $I = \delta^j(H) = \delta^{i+j}(H)$. Then for every $k < j$ and every $n$, we have $\delta^{i+n} = \delta^{i+j+n}(H)$, so the sequence $H^\infty$ exhibits a periodicity of length $j$ (or a divisor of $j$). After the first occurrence of an $I \in \text{Rec}_{\Sigma, \delta}(H)$, all further elements of $H^\infty$ are recurring as well, and in particular, there is a recurring $J$ such that $\hat{\delta}(J) = I$. We shall call this an $I$-predecessor and will use this fact in our proofs.

**Definition 2**

$$[\Sigma, p]_I := \inf\{I(p_i) : I \in \text{Rec}_{\Sigma, \delta}(H)\},$$
$$[\Sigma, p]_\delta := \inf\{I(p_i) : \exists H(I \in \text{Rec}_{\Sigma, \delta}(H))\}.$$  

**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.

As a consequence, the natural notion of logical operations is not that between propositions, but that between systems. If $\Sigma = \{p_0 \leftarrow E_0, \ldots, p_n \leftarrow E_n\}$ 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 $p_{\ldots}$,

$$\neg(\Sigma, p_i) := \Sigma \cup \{\neg p_i\}.$$  

If we have two pointer systems

$$\Sigma_0 = \{p_0 \leftarrow E_{0,0}, \ldots, p_n \leftarrow E_{0,n}\},$$

$$\Sigma_1 = \{p_0 \leftarrow E_{1,0}, \ldots, p_n \leftarrow E_{1,n}\},$$

we make their sets of propositions disjoint by considering a set $\{p_0, \ldots, p_n, p_0^*, \ldots, p_n^*\}$ of $n_0 + n_1 + 2$ propositional variables. We then set

$$\Sigma^* := \{p_0^* \leftarrow E_{1,0}, \ldots, p_n^* \leftarrow E_{1,n}\}.$$  

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

$$(\Sigma_0, p_j) \land (\Sigma_1, p_j) := \Sigma_0 \cup \Sigma_1^* \cup \{p_j \land p_j^*\},$$

$$(\Sigma_0, p_j) \lor (\Sigma_1, p_j) := \Sigma_0 \cup \Sigma_1^* \cup \{p_j \lor p_j^*\}.$$  

**Logical properties of Gaifman pointer semantics**

Fix a system $\Sigma = \{p_0 \leftarrow E_0, \ldots, p_n \leftarrow E_n\}$. A proposition $p_i$ is called a terminal node if $E_i = \infty$. It is called a source 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 including edges in the dependency graph. The Gaifman-Tarski operator $\delta_\Sigma$ is defined as follows:

$$\delta_\Sigma(I)(p_i) := \begin{cases} I(E_i) & \text{if } p_i \text{ is not terminal,} \\ I(p_i) & \text{if } p_i \text{ is terminal.} \end{cases}$$  

Note that this operator can be described as follows:

"From the clause $p_i \leftarrow E$, form the equation $Q_i$ by replacing the occurrences of $p_i$ on the right-hand side of the equality sign with the values $I(p_i)$. If $p_i$ is a terminal node, let $\delta(I)(p_i) := I(p_i)$. Otherwise, let $I^*$ be the unique solution to the system of equations $\{Q_0, \ldots, Q_n\}$ and let $\delta(I)(p_i) := I^*(p_i)$."

This more complicated description will provide the motivation for the forward propagation operator $\delta^I$ in the section "Belief semantics with forward propagation".

The operator $\delta_\Sigma$ gives rise to a logical system, as the semantics defined by $\delta_\Sigma$ are compatible with the operations in the algebra of pointer systems.

**Proposition 3** Let $\Sigma = \{p_0 \leftarrow E_0, \ldots, p_n \leftarrow E_n\}$ be a pointer system. For any $i \leq n$, we have

$$[\neg(\Sigma, p_i)]_{\Sigma} = \neg[\Sigma, p_i]_{\Sigma}.$$  

**Proof.** In this proof, we shall denote interpretations for the set $\{p_0, \ldots, p_n\}$ by capital letters $I$ and $J$ and interpretations for the bigger set $\{p_0, \ldots, p_n, \ldots\}$ by letters $I$ and $J$. It is enough to show that if $I$ is $\delta_\Sigma$-recurring, then there is some $\delta_\Sigma$-recurring $J$ such that $I(p_{\ldots}) = \neg J(p_i)$. If $I$ is $\delta_\Sigma$-recurring, we call $J$ an $I$-predecessor if $J$ is also $\delta_\Sigma$-recurring and $\delta_\Sigma(J) = I$, and similarly for $I$. It is easy to see that every $\delta_\Sigma$-recurring $I$ (or $J$) has an $I$-predecessor (or $I$-predecessor) which is not necessarily unique.

As $\delta_\Sigma$ is a $\Sigma$-operator, we have that if $J$ is $\delta_\Sigma$-recurring, then so is $J := J[I(p_{0, \ldots, p_n})$ is $\delta_\Sigma$-recurring and

$$I(p_{\ldots}) = \delta_\Sigma(J)(p_{\ldots}) = \neg J(p_i) = \neg J(p_i).$$

**q.e.d.**

**Proposition 4** Let $\Sigma_0 = \{p_0 \leftarrow E_0, \ldots, p_n \leftarrow E_n\}$ and $\Sigma_1 = \{p_0 \leftarrow E_0, \ldots, p_n \leftarrow E_n\}$ be pointer systems. For any $i, j \leq n$, we have

$$[\Sigma_0, p_i]_\Sigma \lor [\Sigma_1, p_j]_\Sigma = [\Sigma_0, p_i]_{\Sigma} \lor [\Sigma_1, p_j]_{\Sigma}.$$  

**Similarly for \lor replaced by \land.**

**Proof.** The basic idea is very similar to the proof of Proposition 3, except that we have to be a bit more careful to see how the two systems $\Sigma_0$ and $\Sigma_1$ can interact in the bigger system. We reserve letters $I_0$ and $I_1$ for the interpretations on $\Sigma_0$, $I_0$ and $I_1$ for those on $\Sigma_1$ and $I$ and $J$ for interpretations on the whole system, including $p_\ldots$. If $[\Sigma_0, p_i] = 1$, then any $\delta_\Sigma$-recurring $I$ must have $I(p_i) = 1$ by the $\lor$-analogue of the argument given in the proof of Proposition 3. Similarly, for $[\Sigma_1, p_j] = 1$ and the case that $[\Sigma_0, p_i] = [\Sigma_1, p_j] = 0$. This takes care of six of the nine possible cases.

If $I_0$ and $I_1$ are $\delta_\Sigma$-recurring, then so is the function $I := I_0 \cup I_1 \cup \{p_\ldots, I_0(p_i) \lor I_1(p_i)\}$ (if $I_0$ is $k$-periodic and $I_1$ is $\ell$-periodic, then $I$ is at most $k \cdot \ell$-periodic). In particular, if we have such an $I_0$ with $I_0(p_i) = \frac{1}{2}$ and an $I_1$ with $I_1(p_i) \neq 1$, then $I(p_i) = \frac{1}{2}$ (and symmetrically for interchanged roles). Similarly, if we have recurring interpretations for relevant values $0$ and $1$ for both small systems, we can put them together to $\delta_\Sigma$-recurring interpretations with values $0$ and $1$ for the big system. This gives the truth value $\frac{1}{2}$ for the disjunction in the remaining three cases.

**q.e.d.**
Belief semantics with forward propagation

In (GLS07), the authors gave a revision operator that incorporated both backward and forward propagation. The value of $\delta(I)(p_i)$ depended on the values of all $I(p_j)$ such that $j$ is connected to $i$ in the dependency graph. 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 $\delta_B$, we define $\delta_F$ as follows. Given an interpretation $I$, we transform each clause $p_i \leftarrow E_i$ of the system into an equation $Q_i \equiv I(p_i) = E_i$ 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 $\{Q_0, \ldots, Q_n\}$ of $n + 1$ equations in $T$. Note that we cannot mimic the definition of $\delta_B$ directly: as opposed to the equations in that definition, the system $\{Q_0, \ldots, Q_n\}$ 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 $\delta_F(I)(p_i) := I(p_i)$. Otherwise, let $S$ be the set of solutions to the system of equations $\{Q_0, \ldots, Q_n\}$ and let $\delta_F(I)(p_i) := \inf\{I(p_i); I \in S\}$ (remember that $\inf\emptyset = \frac{1}{2}$). Note that this definition is literally the dual to definition (*) of $\delta_B$ (i.e., it is obtained from (*) by interchanging “right-hand side” and “left-hand side” and “terminal node” by “source node”).

We now combine $\delta_B$ and $\delta_F$ to one operator $\delta_T$ by defining pointwise

$$\delta_T(I)(p_i) := \delta_F(I)(p_i) \otimes \delta_B(I)(p_i)$$

where $\otimes$ has the following truth table:

- $\otimes 0$ 0 1
- 0 0 $\frac{1}{2}$ 1
- $\frac{1}{2}$ 0 1 1
- 1 $\frac{1}{2}$ 1 1

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 3 and 4. 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 *prefix paradox* and has been described by Kyburg as “conjunctivitis” (Kyb70). 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; Cou04). 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 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 $\Sigma := \{p_0 \leftarrow \neg p_0\}$. Since we have only one propositional variable, interpretations are just elements of $T = \{0, \frac{1}{2}, 1\}$. It is easy to see that $\delta_B(0) = \delta_T(0) = \delta_T(1) = 1$, $\delta_B(\frac{1}{2}) = \delta_F(\frac{1}{2}) = \delta_T(\frac{1}{2}) = \frac{1}{2}$, and $\delta_B(1) = \delta_F(1) = \delta_T(1) = 0$. This means that the $\delta_T$-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
\[
H_0^* = (0, 1, 0, 1, 0, 1, 0) \quad \text{and} \quad H_1^* = (1, 1, 0, 0, 1, 1, 0).
\]

Figure 2: The first three iterations of values of \(H_0^* = (0, 1, 0, 1, 0, 1, 0, 1/2)\) up to the point of stability \((H_1^* = \ldots)\).

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 10am and the library committee meeting is at 11am. 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.

\[
\begin{align*}
p_0 &\leftarrow \neg p_1 \land \neg p_4, \\
p_1 &\leftarrow \neg p_0 \land p_2 \land p_4, \\
p_2 &\leftarrow \neg p_3, \\
p_3 &\leftarrow p_0 \land p_2 \land p_4, \\
p_4 &\leftarrow \neg p_0 \land p_2 \land p_4,
\end{align*}
\]

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 \(H := (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1/2)\) (here, as usual, we identify an interpretation with its sequence of values in the order of the indices of the propositional letters). Then \(\delta_B(H) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, 1/2)\) and \(\delta_F(H) = (\frac{1}{2}, \frac{1}{2}, 1, 1, 1/2)\), so that we get \(H'_1 := \delta_B(H) = (\frac{1}{2}, \frac{1}{2}, 1, 1, 1/2)\). Then, in the second iteration step, \(\delta_B(H') = (\frac{1}{2}, \frac{1}{2}, 1, 1/2)\) and \(\delta_F(H') = (\frac{1}{2}, \frac{1}{2}, 1, 1/2)\), so we obtain stability at \(\delta_B(H') = H'\).

### 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 \(\delta_T\) is rather far from the logical properties of \(\delta_B\) discussed in Propositions 3 and 4.

#### Negation

Consider the pointer system \(\Sigma\) given by

\[
\begin{align*}
p_0 &\leftarrow \neg p_3, \\
p_1 &\leftarrow \neg p_0 \land p_2 \land p_4, \\
p_2 &\leftarrow \neg p_3, \\
p_3 &\leftarrow p_0 \land p_2 \land p_4, \\
p_4 &\leftarrow \neg p_0 \land p_2 \land p_4,
\end{align*}
\]

The interpretation \(H := (0, 1, 0, 1/2)\) is \(\delta_T\)-stable, as \(\delta_B(H) = (\frac{1}{2}, \frac{1}{2}, 0, 0)\), \(\delta_F(H) = (0, \frac{1}{2}, \frac{1}{2}, 1)\), and thus \(\delta_T(H) = H\).

Now let us consider the system \(\neg(\Sigma, p_3)\). Remember from the proof of Proposition 3 that stable interpretations for the small system could be extended to stable interpretations for the big system by plugging in the expected value for \(p_3\). So, in this particular case, the stable value for \(p_3\) is ½, so we would expect that by extending \(H\) by \(H_0(p_3) := \frac{1}{2}\), 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 \(\Sigma\) and \(\Sigma^*\) and their disjunction \((\Sigma, p_4) \lor (\Sigma^*, p_4^*)\) given as follows:

\[
\begin{align*}
p_0 &\leftarrow \neg p_3, \\
p_0^* &\leftarrow \neg p_3, \\
p_1 &\leftarrow \neg p_0 \land p_2 \land p_4, \\
p_1^* &\leftarrow \neg p_0 \land p_2 \land p_4, \\
p_2 &\leftarrow \neg p_3, \\
p_3 &\leftarrow p_0 \land p_2 \land p_4, \\
p_3^* &\leftarrow p_0 \land p_2 \land p_4, \\
p_4 &\leftarrow \neg p_0 \land p_2 \land p_4, \\
p_4^* &\leftarrow p_0 \land p_2 \land p_4.
\end{align*}
\]

Note that \(\Sigma\) and \(\Sigma^*\) are the same system up to isomorphism and that \(\Sigma\) is the system from the previous example. We already know that the interpretation \(H := (0, 1, 0, 1/2)\) is \(\delta_T\)-stable (therefore, it is \(\delta_T\)-stable for both \(\Sigma\) and \(\Sigma^*\) in the appropriate reading).

The natural extension of \(H\) to the full system with nine propositional variables would be \(H_0^* := (0, 1, 0, 1/2, 0, 1, 0, \frac{1}{2}, \frac{1}{2})\), as \(p_4\) should take the value \(H(p_4) \lor H(p_4^*) = \frac{1}{2} \lor 0 = \frac{1}{2}\). 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 $\delta_B$ and $\delta_T$ 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 $H^\infty$ into multiple sequences if there are multiple solutions. For instance, if we are trying to calculate $\delta_F(H)$ and we have multiple solutions to the set of equations, then $\delta_F(H)$ 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


