Belief flow in assertion networks

Sujata Ghosh[1,2⋆], Benedikt Löwe[2,3,4], Erik Scorelle[2]

[1] Department of Mathematics
Visva Bharati
Santiniketan, West Bengal, India

[2] Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
{sujata,bloewe,escorell}@science.uva.nl

[3] Mathematisches Institut
Rheinische Friedrich-Wilhelms-Universität Bonn
Beringstraße 1, 53115 Bonn, Germany

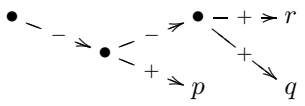
[4] Department Mathematik
Universität Hamburg
Bundesstrasse 55, 20146 Hamburg, Germany

Abstract. We define an abstract model of belief propagation on a graph based on the methodology of the revision theory of truth together with the *Assertion Network Toolkit*, a graphical interface designed to test our semantics.

1 Introduction

Formulas as labelled graphs.

Formulas of ordinary propositional logic can be seen as trees, with the propositional variables as terminal nodes. For example,



reads as $\neg(p \,\&\, \neg(q \,\&\, r)) \equiv \neg p \vee (q \,\&\, r) \equiv p \rightarrow (q \,\&\, r)$.

In order to compute the truth value of a molecular formula of propositional logic, we take an assignment of truth values for the propositional variables (the terminal nodes of the tree) and let the truth value develop backwards along the edges of the tree by the usual rules of propositional logic.[1] In that sense, we can visualize truth as flowing from the leaves to the root of the tree.

⋆ The first author acknowledges the travel support of the ILLC in Amsterdam for a research visit in June 2006, and a **Rubicon** grant of the NWO (680-50-0504) for her visit in the academic year 2006/07.

[1] Exact definitions will follow in the proof of Theorem 2.

Cyclic graphs.

In a logic that allows self-reference, we are not dealing with trees anymore, but with arbitrary, possibly cyclic graphs. The simple flow of truth values from the leaves to the roots may now become a complicated and potentially never-ending pattern. Consider the Liar sentence which corresponds to the graph

While this creates serious problems for sets of sentences with self-reference, the revision theory of truth developed by Herzberger, Gupta, and Belnap [He82a,GuBe93] and the Gaifman Pointer Semantics [Ga88,Ga92] have extended the idea of a set of sentences as a graph to the case involving circularity. These theories consider infinite, possibly (if the graph is infinite) transfinite sequences of revision along the edges of the graph, and then singles out those patterns that are stable as the semantics of the set of sentences. The two characteristic components of revision theory are *backward propagation of truth along the edges* ("revision") and *identification of the stable values*. This was discussed in the survey talk *Revision Forever!* at ICCS 2006 [Lö06].[2]

Even though the revision theory of truth has a wide range of applications (*cf.* [Lö06, § 6]), its basic concepts are determined by the fact that it is a theory of truth: the nodes in the graph represent statements and we want to investigate the circular nature of truth in situations with self-reference. A typical example for this would be the *Nested Liars*:

A: "Everything that B says is false!"
B: "Everything that A says is false!"

A simple analysis tells us that exactly one of A and B speaks the truth. Without additional information, we cannot determine which one is the liar, but we can rule out certain truth-value patterns. While the nested liars are not paradoxical in the strict sense, as they admit consistent valuations, they still pose a problem: the nested liars are completely symmetric, but none of the consistent valuations is. If you think of the nested liars as an abstract, disembodied system of sentences, then they *should* have a symmetric valuation, but they don't!

But in real life, we are not dealing with abstract, disembodied systems of sentences. Behind the letters A and B, there are agents in a communication situation, and depending on who they are and how much we trust their judgement, the situation might turn out not to be symmetric after all. Even on Smullyan's logic island, if one of them is a Knight and the other one is a Knave, we might have independent information that leads us to believe which one is the Knight.

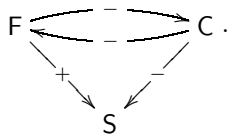
Thinking of nodes in the graph as agents rather than sentences leads from an analysis of truth to an analysis of a belief network. Consider the following

[2] For more details, *cf.* [Kü+05].

example: 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 on the phone to his friend 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, interpreting an arrow labelled + as "everything uttered is true", an arrow labelled − as "everything uttered is false", and S as "the sun is shining":



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. Based on this preference, we experience *beliefs* flowing from the leaves through the graph: the contextually based stronger belief in the positive arrow $\mathsf{F} \to \mathsf{S}$ leads us to disbelieve the negative arrows $\mathsf{C} \to \mathsf{S}$ (since it is in conflict with $\mathsf{F} \to \mathsf{S}$) and $\mathsf{C} \to \mathsf{F}$ (since we believe one of the statements that F makes, the utterance "Everything your friend says is false" must be disbelieved), and then in the next step to believe the arrow $\mathsf{F} \to \mathsf{C}$.

This example shows that even in the context of belief and agents, the underlying idea of pointer semantics with its backward propagation of truth values is still fruitful.

However, as soon as we interpret the values assigned to nodes and arrows as *belief values* rather than truth values, a new phenomenon occurs that was not present in the case of the theory of truth: *forward propagation of belief along the edges*: If a trusted source A states "φ is false", but the reasoning agent believes in φ, then his belief in φ should influence his trust in A, but also the agent's trust in A should influence his belief in φ.

Aim of the paper. Related work.

Our aim in this paper is to provide a formal model that handles both backward and forward influence of values in the graph while keeping the spirit of revision semantics. Based on this model, we define a belief semantics via stability and check that this captures our intuitions of belief dynamics.

Of course, our model is neither the first attempt to provide a formal background for reasoning about beliefs in a multi-agent setting nor the first one to consider the evolution of numbers on a graph network.

There are many approaches to reasoning about beliefs (and knowledge) with many agents. As an example, consider dynamic epistemic logic [Ge99] or announcement logics [Pl89]. This theory uses graphs in the spirit of modal logic in order to represent states of the world. The dynamics results in changing of the graph (as new belief states are created by actions or deleted by generated knowledge). This is very different from our approach. Another example, closer to traditional belief revision, is the work of Dragoni and Giorgini on multi-source belief revision [DrGi01] which shares some features with our set-up.

Also the idea of attaching numbers representing beliefs to nodes of a graph is not at all new: there is a large body of literature on *Bayesian Belief Nets* (see, e.g., [Pe88,Wi05]) with a lot of interesting technical results. The set-up (assigning probabilities to the nodes and using the Bayesian rule for propagation) is close to our approach (see also § 5), but seems to be lacking the notion of stability that is crucial to our own semantics. We do find the notion of stability in the research area of *graph automata* (the graph version of cellular automata as described in [ToKuMu02]).

A very intriguing connection can be found to the area called *social network analysis* which ranges from empirical social science to computer science [WeBe88,BrEr05].

What is novel about our approach is the combination of number propagation (as known in neural networks or cellular automata) and the semantics derived from stability (as used in the revision theory of truth).

Outline of the paper.

In § 2, we provide the abstract formal background called **assertion network semantics** that will then have to be instantiated by concrete functions and operations. The aim is then to find concrete functions that make assertion network semantics recover our natural intuitions. This is necessarily an empirical study, and we shall see an extensive example in § 3. Since there are so many potential concretizations of our assertion network semantics, we decided to support our work with a software tool called *Assertion Network Toolkit* which is described in § 4. In our conclusion in § 5, we list the further projects coming out of the work described in this paper.

2 The most abstract model

In our paper, a **(directed) labelled graph** G is a triple $G = \langle V, E, \ell \rangle$, where V is a set of nodes, $E \subseteq V \times V$ is the set of edges, and $\ell : E \to \{+, -\}$ is a labelling function. We write

$$\begin{aligned} \mathrm{In}(v) &:= \{e \in E\,;\, \exists w(e = \langle w, v\rangle)\}, \text{ and} \\ \mathrm{Out}(v) &:= \{e \in E\,;\, \exists w(e = \langle v, w\rangle)\}, \end{aligned}$$

and let $\mathrm{indeg}(v) := |\mathrm{In}(v)|$ and $\mathrm{outdeg}(v) := |\mathrm{Out}(v)|$ denote the indegree and outdegree of v, respectively. We denote the set of terminal nodes by $T := \{v \in V\,;\, \mathrm{outdeg}(v) = 0\}$.

In our graphs, the terminal nodes stand for facts (like "the sun is shining"). The non-terminal nodes correspond to agents making statements, either about a fact or about an agent. Labelled edges are these statements, where an edge labelled + corresponds to a positive statement ("is true" or "Everything he or she says is true") and an edge labelled − corresponds to a negative statement ("is false" or "Everything he or she says is false"). This intended semantics is captured in the following definition: a function $I : V \to \{0,1\}$ is called an **interpretation (on a labelled graph)**. Let $* : \{+,-\} \times \{0,1\} \to \{0,1\}$ be defined by $+ * 0 = - * 1 = 0$ and $+ * 1 = - * 0 = 1$. We say that an interpretation **respects** a labelled graph G if for all nonterminal vertices v, we have $I(v) = \bigwedge_{w \in \mathrm{Out}(v)} \ell(v,w) * I(w)$. Obviously, every function $I : T \to \{0,1\}$ has at most one extension to an interpretation that respects G.

Our labelled graphs are similar and yet different from Bolander's dependency graphs and the corresponding pointer semantics [Bo03, Chapter 5]: non-equivalent sets of clauses can have the same dependency graph whereas the labels of the edges make the semantics of the labelled graph unequivocal.[3]

The following theorem connects labelled graphs with the above semantics to pointer semantics.

Theorem 1. *Labelled graphs interpret pointer semantics in the following sense: for every set of clauses Σ with propositional variables $\{\mathrm{p}_0, ..., \mathrm{p}_n\}$ there is a labelled graph G with vertices $\{v_0, ..., v_n\}$ such that an interpretation (in the sense of footnote 3) $I : \mathbb{N} \to \{0,1\}$ respects Σ if and only if the interpretation (on the labelled graph) $I^* : V \to \{0,1\}$ defined by $I^*(v_i) := I(i)$ respects G.*

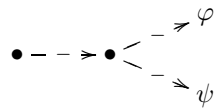
Proof. Fix a set of clauses $\Sigma = \{i{:}E_i\,;\, 0 \leq i \leq n\}$ and write all expressions E_i in disjunctive normal form, *i.e.*, $E_i = \bigvee_k \bigwedge_j \ell_{ijk}$ where ℓ_{ijk} is either a propositional variable or a negation of a propositional variable.

To start, let us notice that our intended semantics for the labelled graphs only deals with conjunction, not disjunction ("everything he says is true", "everything he says is false"). Therefore, we have to define disjunction via de Morgan's laws,

[3] Here, we refer to the pointer semantics of a propositional language as described in [Lö06, § 2]: Every propositional variable is an **expression**; $\bot$ and $\top$ are expressions; if E and F are expressions, then $\neg E$, $E \wedge F$, and $E \vee F$ are expressions. If E is an expression and n is a natural number, then $n{:}\,E$ is a **clause**.

We say that an **interpretation** is a function $I{:}\,\mathbb{N} \to \{0,1\}$ assigning truth values to propositional letters. Obviously, an interpretation extends naturally to all expressions. Now, if $n{:}\,E$ is a clause and I is an interpretation, we say that I **respects** $n{:}\,E$ if $I(n) = I(E)$, and I respects a set of clauses Σ if it respects every element of Σ.

where $\varphi \vee \psi$ is represented by



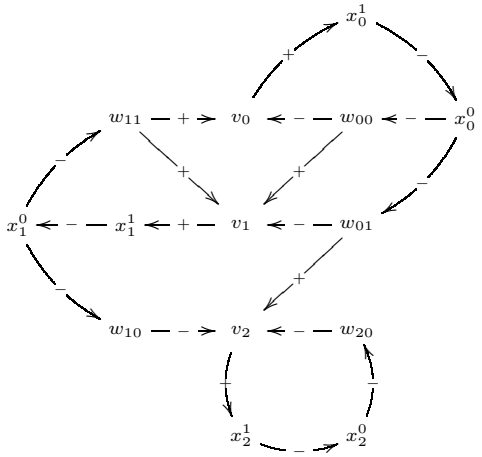
We construct a labelled graph to express Σ. We start with the set of vertices: for each i, we take a vertex v_i representing p_i, then for each conjunctive clause $\bigwedge_j \ell_{ijk}$ occurring in one of the expressions, we take a vertex w_{ik}. In order to deal with the disjunctions, we add vertices x_i^0 and x_i^1 for every expression E_i.

Now the edges are drawn as follows: If for some j, we have $\ell_{ijk} = \mathrm{p}_{i^*}$, we draw an edge $w_{ik} - + \rightarrow v_{i^*}$, if for some j, we have $\ell_{ijk} = \neg\mathrm{p}_{i^*}$, we draw an edge $w_{ik} - - \rightarrow v_{i^*}$, and for all i and k, we draw edges $x_i^0 - - \rightarrow w_{ik}$, $x_i^1 - - \rightarrow x_i^0$, and $v_i - + \rightarrow x_i^1$. Obviously, this general construction produces a labelled graph with the desired property.[4] q.e.d.

As mentioned in the introduction, we shall now combine the idea of (stable) revision semantics with forward propagation of belief along edges. Fix some metric space $\langle \Lambda, d \rangle$ of **values**. We call a function $H : E \cup V \to \Lambda$ a **hypothesis**. This can be interpreted as a state of beliefs (about the communication situation) of the reasoning agent.

For an edge $e = \langle v, w \rangle$, let $M_e := \mathrm{In}(v) \cup \{v, e, w\} \cup \mathrm{Out}(w)$ and $n_e := \mathrm{Card}(M_e)$. For a vertex v, let $M_v := \mathrm{In}(v) \cup \{v\} \cup \mathrm{Out}(v)$ and $n_v := \mathrm{Card}(M_v)$. If $M_x = \langle x_1, ..., x_m \rangle$, we write $H(M_x) := \langle H(x_1), ..., H(x_m) \rangle$. For every $x \in E \cup V$,

[4] In order to illustrate the procedure, let us give an example: Consider the set of clauses $\{0{:}(\neg\mathrm{p}_0 \wedge \mathrm{p}_1) \vee (\neg\mathrm{p}_1 \wedge \mathrm{p}_2), 1{:}\neg\mathrm{p}_2 \vee (\mathrm{p}_0 \wedge \mathrm{p}_1), 2{:}\neg\mathrm{p}_2\}$. By the rules given above, this transforms into the following labelled graph:



we fix an evaluation function $\Psi_x : \Lambda^{n_x} \to \Lambda$ which we can now use to define a **revision sequence** in the spirit of revision theory:

Fix an initial hypothesis H. We define the sequence $\langle H_i\,;\, i \in \omega\rangle$ at some $x \in E \cup V$ by simultaneous recursion as follows:

$$H_0(x) := H(x)$$
$$H_{i+1}(x) := \Psi_x(H(M_x)).$$

Inspired by the stability concept of revision theory, we can now define a partial stability semantics for our labelled graph. Suppose you have an initial hypothesis H, some $x \in E \cup V$ and some $\lambda \in \Lambda$ such that $\lim_{i\to\infty} H_i(x) = \lambda$, then we say that **the value of x is stably λ in situation H**. We call this the **assertion network semantics** A_H defined by

$$\mathrm{A}_H(x) := \begin{cases} \lambda & \text{if the value of } x \text{ is stably } \lambda \text{ in situation } H, \text{ and} \\ \text{undefined} & \text{if } \langle H_i(x)\,;\, i \in \omega\rangle \text{ diverges.} \end{cases}$$

Theorem 2. *The stable truth predicate of revision semantics is a special case of assertion network semantics, i.e., for every set of clauses Σ there is a labelled graph G and there are evaluation functions such that A_H coincides with the (partial) stable truth predicate on Σ.*

Proof. We shall give evaluation functions Ψ_x such that the assertion network semantics recover the (partial) stable truth predicate defined by

$$\mathrm{S}_H(\mathrm{p}_i) := \begin{cases} \mathsf{true} & \text{if } \mathrm{p}_i \text{ is stably true in the sequence starting with } H, \\ \mathsf{false} & \text{if } \mathrm{p}_i \text{ is stably false in the sequence starting with } H, \text{ and} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Here, we are using Theorem 1 by dealing with revision semantics in labelled graphs instead of dependency graphs.[5]

As values, we choose $\Lambda := \{\mathsf{true}, \mathsf{false}\}$ with the discrete metric. In the classical revision semantics, edges don't play a rôle, so we shall now identify the values of an edge $e = \langle v, w\rangle$ and w.

For a given hypothesis H, we define the following variant H^* on the edges of the labelled graph that incorporates the values of the labels: $H^*(e) := H(e)$ if $\ell(e) = +$ and $H^*(e) := \neg H(e)$ if $\ell(e) = -$. Now, for each $e = \langle v, w\rangle \in E$, we let

$$\Psi_e(H(M_e)) := \begin{cases} \mathsf{true} & \text{if for all } e^* \in \mathrm{Out}(w), \text{ we have } H^*(e^*) = \text{true, and} \\ \mathsf{false} & \text{otherwise.} \end{cases}$$

Clearly, with the functions Ψ_e, the assertion network semantics gives back the stable semantics as given above. q.e.d.

[5] The partial semantics S_H corresponds to the lightface version of the stable truth predicate in [Lö06, p. 27] as opposed to the boldface version integrating over all starting hypotheses. *Cf.* [Kü+05].

3 Concretization

The goal of this enterprise was to develop a semantics to deal with beliefs in assertion situations with reference according to our intuitions. Now, the abstract assertion network semantics depends on the proper choice of the evaluation functions Ψ_x, and without fixing these functions, it is not concrete enough to allow checking the resulting semantics against our intuitions.

The search for a concretization of the abstract model that conforms to our intuitions is an empirical question, and we shall return to this in § 4.

For the time being, let us now concentrate on one rather natural example. We choose $\Lambda := [-1, 1]$ where we interpret 1 as "the agent firmly believes", -1 as "the agent firmly disbelieves", and 0 as "the agent has no evidence either way". We'll focus on the belief change in the edges (representing the statements) and the terminal nodes, and consider the nonterminal nodes as auxiliary. In the following, since we want to keep our values between -1 and 1, we shall be using the function $\mathsf{R}(r) := \min(\{\max(\{-1, r\}), 1\})$, and if e is an edge, we let

$$s_e := \begin{cases} 1 & \text{if } \ell(e) = +, \text{ and} \\ -1 & \text{if } \ell(e) = -. \end{cases}$$

1. If $x \in V \backslash T$, and $\{e_0, ..., e_{n_x}\} = \mathrm{Out}(x)$, then we let

$$\Psi_x(\lambda_1, ..., \lambda_{n_x}) := \frac{\sum_{0 \leq i \leq n_x} \lambda_i}{n_x}.$$

2. For a terminal node t, the value of Ψ_t depends on the value of t and the values of the incoming edges. Let $\{e_0, ..., e_{n_t}\} = \mathrm{In}(t)$, then

$$\Psi_t(\lambda, \lambda_0, ..., \lambda_{n_t}) := \mathsf{R}\left(\lambda + \frac{\sum_{1 \leq i \leq n_t} s_i \lambda_i}{n_t}\right),$$

   where λ is the value of t and λ_i is the value of e_i.
3. Finally, for an edge $e = \langle v, w \rangle$, we let the value depend on the value λ of e, the value $\widehat{\lambda}$ of w and the values $\lambda_0, ..., \lambda_{n_e}$ of $\mathrm{In}(v)$.

$$\Psi_e(\lambda, \widehat{\lambda}, \lambda_0, ..., \lambda_{n_e}) := \begin{cases} \frac{1}{2} \cdot \left(\mathsf{R}\left(\lambda + \frac{\sum_{1 \leq i \leq n_e} s_i \lambda_i}{n_t}\right) + \widehat{\lambda}\right) & \text{if } \ell(e) = +, \\ \frac{1}{2} \cdot \left(\mathsf{R}\left(\lambda + \frac{\sum_{1 \leq i \leq n_e} s_i \lambda_i}{n_t}\right) - \widehat{\lambda}\right) & \text{if } \ell(e) = -. \end{cases}$$

Note that this choice of evaluation functions is not unique, but rather natural, including all of the relevant information in the computation of Ψ_x without discrimination, except that the original value of a node gets more influence than the incoming information.

To see this choice of evaluation functions at work, let us consider the following communication situation:

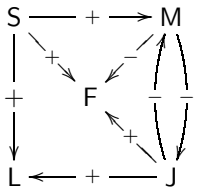
Professors Jones, Miller and Smith are colleagues in a computer science department. Jones and Miller dislike each other without reservation and are very liberal in telling everyone else that "everything that the other one says is false". Smith just returned from a trip abroad and needs to find out about 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."*

This situation is described by the following diagram:



Given the described situation, we assign an initial hypothesis to this graph as follows: the agent has no evidence for any of the statements except that he believes that the secretary (being well-informed about administrative matters) is correct about the two meetings, so we assign the value 0 to everything except for the edges $\mathsf{S} \to \mathsf{L}$ and $\mathsf{S} \to \mathsf{F}$ where we assign $+0.5$. When we run the revision, all values converge (up to two decimal digits precision) after at most 16 steps of revision, giving the following stability pattern:

S	M	J	F	L	S → L	S → F	J → L	J → F	M → F	J → M	M → J	S → M
0	0	0	0	0	+0.5	+0.5	0	0	0	0	0	0
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
+0.3	−1.0	+1.0	+1.0	+1.0	+1.0	+1.0	+1.0	+1.0	−1.0	+1.0	−1.0	−1.0

This corresponds perfectly to our intuitions: if we give belief primacy to the two factual announcements of the secretary, then Jones is speaking the truth twice and Miller is uttering a falsehood. To get an even better feeling of how the system behaves, we shall now gradually change the initial hypothesis by increasing all values that are in favour of Miller (i.e., the values of $\mathsf{M} \to \mathrm{F}$, $\mathsf{M} \to \mathsf{J}$, and $\mathsf{S} \to \mathsf{M}$. Let us define the hypothesis H^λ by

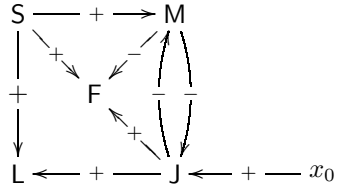
$$H^\lambda(x) := \begin{cases} +0.5 \text{ if } x \in \{\mathsf{S} \to \mathsf{F}, \mathsf{S} \to \mathsf{L}\}, \\ \lambda \quad \text{if } x \in \{\mathsf{M} \to \mathrm{F}, \mathsf{M} \to \mathsf{J}, \mathsf{S} \to \mathsf{M}\}, \text{ and} \\ 0 \quad \text{otherwise.} \end{cases}$$

The special case H^0 is the case discussed above. We shall now slowly increase λ and observe the stability behaviour of the system. For $\lambda = +0.1$, all limits stay

the same, even though the rate of convergence is a bit slower in some cases. But for $\lambda = +0.2$, we can observe the first change, as the value of F now converges to -1.0 (instead of $+1.0$). When λ is $+0.5$, the values of both F and L converge to -1.0.

Independent Evidence

Our assertion networks allow us to formalize (abstract) independent evidence. For any given labelled graph $\langle V, E, \ell \rangle$ with initial hypothesis H, we define **independent evidence for** $v \in V$ **of strength** n to be represented by a labelled graph $\langle V^*, E^*, \ell^* \rangle$ with $V^* = V \cup \{x_0, ..., x_{n-1}\}$, $E^* := E \cup \{\langle x_i, v \rangle ; i < n\}$, and ℓ^* extending ℓ with $\ell^*(\langle x_i, v \rangle) = +$, and an initial hypothesis H^* extending H with $H^*(x_i) = H^*(\langle x_i, v \rangle) = +1.0$.[6] In the following diagram, independent evidence of strength 1 has been added to Jones (J):



Adding independent evidence has the expected effect. For instance, if we add independent evidence of strength 1 to J as in the above picture, then the effect of increasing λ in the initial hypothesis H^λ is weakened. In order to get a stable value -1.0 for F, you have to go to $H^{+0.3}$ (instead of $H^{+0.2}$ without independent evidence), and in order to get stable values -1.0 for both terminal nodes, you have to move to $H^{+0.6}$ (instead of $H^{+0.5}$ in the case without independent evidence).

4 The Assertion Network Toolkit

As mentioned in § 3, the search for the right functions Ψ_x in order to capture our intuitions is a largely empirical endeavour. In order to test candidate functions,

[6] The proper modelling of independent evidence requires forward propagation of belief as the following example shows:

Consider the graph $A \multimap B$ with $H(B) = -1$. If we use a revision theoretic model that only allows backward propagation (*e.g.*, the one given in the proof of Theorem 2), then the outcome would be disbelief in A, $A \to B$ and B. In a semantics with only backward propagation (*i.e.*, $H_{i+1}(B)$ depends only on $H_i(B)$, we could not determine the difference between this scenario and all of the following scenarios:

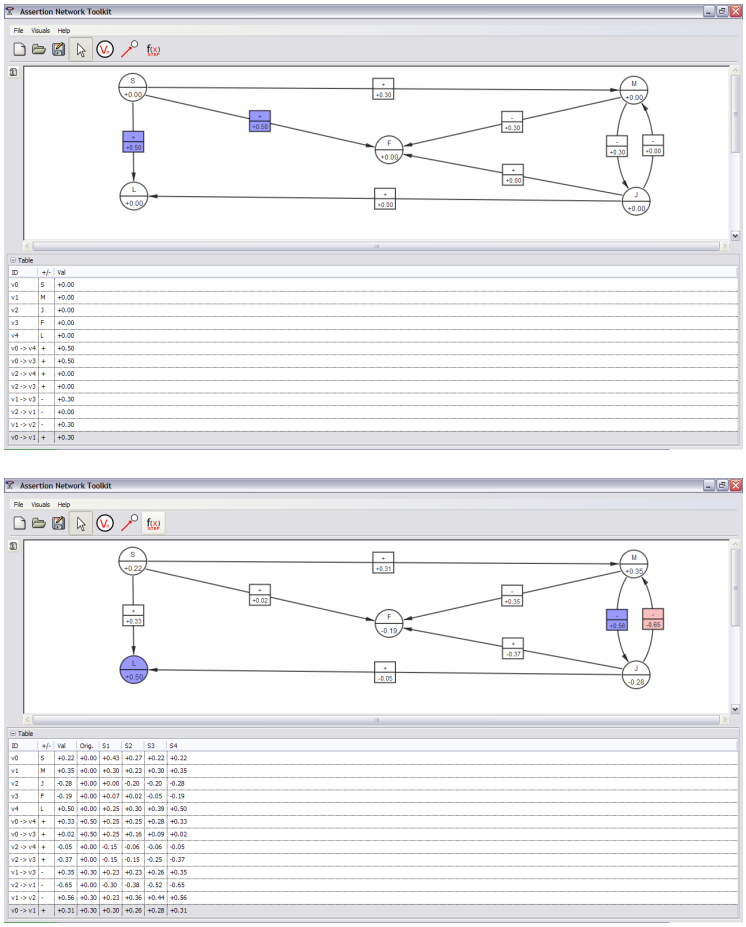


Fig. 1.

we have to go through a large number of examples, in particular more complicated examples that require a sufficiently large number of steps to converge. The example discussed in § 3 has five vertices and eight edges. Already in this case, doing all computations by hand is cumbersome, for larger examples, it is virtually impossible.

As a consequence, we decided to implement a piece of graphical interface software that allows us to play around with the functions and values. This will be used in the future to investigate the advantages and weaknesses of assertion network semantics.

The **Assertion Network Toolkit** is written solely in C/C++ and built on the Boost Graphing Library (BGL), GTK, GTKmm, and the Cairo graphics library. Boost property maps contain all the data of a graph object while the

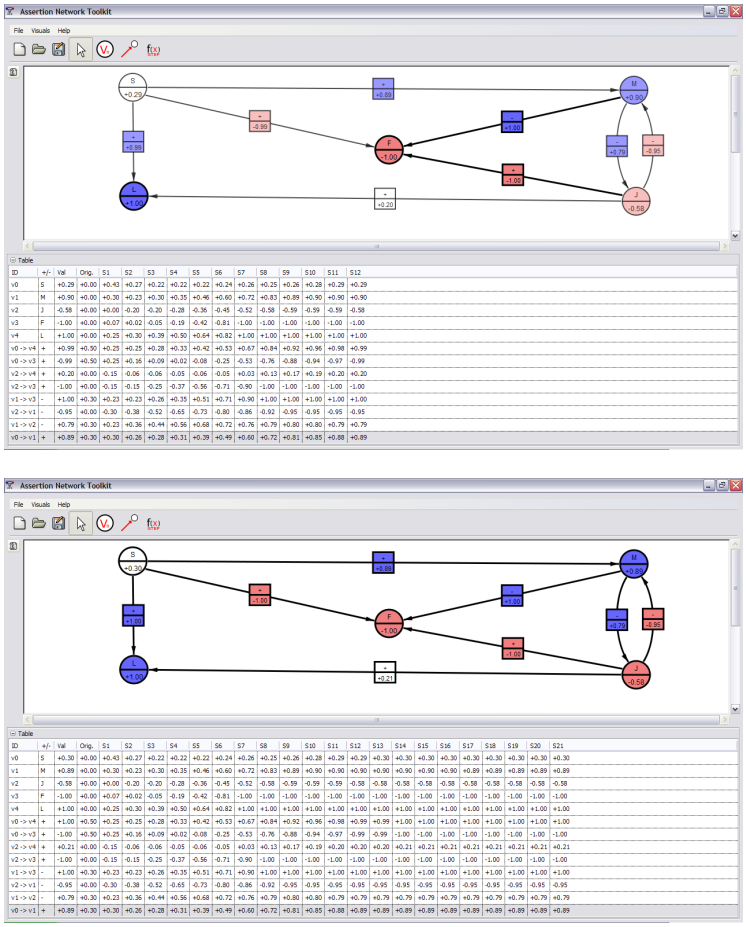


Fig. 2.

drawing area and table widgets serve as a view for the values stored in the maps. Stepping through a function locks the map and updates the view according to the new values. These properties and functions are currently predefined, but due to the flexibility of BGL and the independence of the graph from the views, the next version of the Assertion Network Toolkit will allow users to create graphs with custom properties and define the behavior of a function. The software is platform independent and will remain so as it is developed more.

In the Assertion Network Toolkit, the set-up of a labelled graph is user-friendly and simple. It allows to set the evaluation functions and the initial hypothesis, and then runs the revision according to the functions. In the following example, we look at the graph discussed as an example above with the initial hypothesis $H^{+0.3}$.

In the first picture of Figure 1, we can see the finished graph with the initial values according to $H^{+0.3}$. The edges $\mathsf{S} \to \mathsf{L}$ and $\mathsf{S} \to \mathsf{F}$ are marked blue (darker shade of grey in black-and-white print) as they have a positive value higher than a threshold value set by the user.

We start the revision procedure by clicking the "step" button. Immediately, the two edges marked blue fall below the threshold value and lose their blue marking. After four steps (the next picture), we can now see that the edge $\mathsf{M} \to \mathsf{J}$ and the node L are above the threshold of $+0.5$ and are therefore marked in blue. Similarly, the edge $\mathsf{J} \to \mathsf{M}$ is below the lower threshold value and is consequently marked in red (lighter shade of grey in black-and-white print).

After twelve steps (the first picture of Figure 2), we see that a number of nodes and edges has now reached the threshold values and has been marked red or blue. In addition, we now see that the values of F and L have been constant for the last five iterations. Again, this is a definition of convergence that can be set by the user. In this particular example, we defined no change for five iterations as convergence which is marked by setting the respective vertex or edge in bolded lines.

After twenty-one iterations (the final picture), all vertices and edges have reached that status and are marked in bolded lines.

5 Conclusion

In § 2, we have presented an abstract framework for a revision semantics with forward flow of belief. The abstract framework has to be made concrete by the choice of evaluation functions Ψ_x. Whether a given choice of function is appropriate, i.e., conforms with our intuitions about human belief formation, is an empirical question. This empirical question can and should be tested with many examples; for these tests, we provided the *Assertion Network Toolkit* in § 4. In § 3, we have provided one rather natural specification of the abstract framework and have discussed how it behaves for our given example.

The most natural task for the future is the **experimental** project of testing various natural choices of transition functions (including the one given in § 3) in many examples to investigate their behaviour. These experiments will also include a closer investigation of the effects of independent evidence. On the more mathematical side, there is the **logical** task of investigating the logic of our semantics: after fixing transition functions and a notion of *validity* (e.g., $H(v) > 0.75$ or $H(v) = +1.0$), what are the validities exhibited by our semantics? The broadest project is the **comparative** project of connecting our approach to the other models involving graphs and belief (e.g., belief nets). Can one of the models be obtained as a special case of the other (as in Theorem 2)?

References

[Bo03] Thomas **Bolander**, Logical Theories for Agent Introspection, PhD thesis, Technical University of Denmark 2003

[BrEr05] Ulrik **Brandes**, Thomas **Erlebach** (*eds.*), Network Analysis, Methodological Foundations, Springer-Verlag 2005 [Lecture Notes in Computer Science 3418]

[DrGi01] Aldo Franco **Dragoni**, Paolo **Giorgini**, Revising Beliefs received from Multiple Sources, *in:* Mary-Anne Williams, Hans Rott (*eds.*), Frontiers in Belief Revision, Kluwer Academic Publishers, 2001 [Applied Logic Series 22], p. 429-442

[Ga88] Haim **Gaifman**, Operational Pointer Semantics: Solution to Self-referential Puzzles I, *in:* Moshe Vardi(*ed.*), Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge, Pacific Grove, CA, March 1988, Morgan Kaufmann, San Francisco 1988, p. 43–59

[Ga92] Haim **Gaifman**, Pointers to Truth, **Journal of Philosophy** 89 (1992), p. 223–261

[Ge99] Jelle **Gerbrandy**, Dynamic epistemic logic, *in:* Lawrence S. Moss, Jonathan Ginzburg, Maarten de Rijke (*eds.*), Logic, language and computation, Volume 2, Proceedings of the 2nd Conference on Information-theoretic Approaches to Logic, Language and Computation (ITALLC) held at Regent's College, London, July 1996, CSLI Publications, 1999 [CSLI Lecture Notes 96], p. 67-84

[GuBe93] Anil **Gupta**, Nuel **Belnap**, The Revision Theory of Truth, Cambridge MA 1993

[He82a] Hans G. **Herzberger**, Naive Semantics and the Liar Paradox, **Journal of Philosophy** 79 (1982), p. 479–497

[He82b] Hans G. **Herzberger**, Notes on Naive Semantics, **Journal of Philosophical Logic** 11 (1982), p. 61–102

[Kü+05] Kai-Uwe **Kühnberger**, Benedikt **Löwe**, Michael **Möllerfeld**, Philip **Welch**, Comparing inductive and circular definitions: parameters, complexities and games, **Studia Logica** 81 (2005), p. 79-98

[Lö06] Benedikt **Löwe**, Revision Forever!, *in:* Henrik Schärfe, Pascal Hitzler, Peter Øhrstrøm (*eds.*), Proceedings of the 14th International Conference on Conceptual Structures, ICCS06, Aalborg, Denmark, 2006, Springer-Verlag, Berlin, 2006 [Lecture Notes in Artificial Intelligence 4068], p. 22-36

[Pe88] Judea **Pearl**, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann, 1988 [The Morgan Kaufmann Series in Representation and Reasoning]

[Pl89] Jan A. **Plaza**, Logics of Public Communications, *in:* M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, Z. W. Ras (*eds.*), Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems: Poster Session Program, Oak Ridge National Laboratory, 1989 [ORNL/DSRD-24], p. 201-216

[Sc92] John P. **Scott**, Social Network Analysis, Sage, 1992

[ToKuMu02] Kohji **Tomita**, Haruhisa **Kurokawa**, Satoshi **Murata**, Graph automata: natural expression of self-reproduction, **Physica D: Nonlinear Phenomena** 171 (2002), p. 197-210

[WeBe88] Barry **Wellman**, S. D. **Berkowitz**, Social Structures: A network approach, Cambridge University Press, 1988

[Wi05] Jon **Williamson**, Bayesian nets and causality, Philosophical and computational foundations, Oxford University Press, 2005