Journal of Philosophical Logic (2006) 35: 65–81
DOI: 10.1007/s10992-005-9002-y

DOV GABBAY, ROLF NOSSUM and JOHN WOODS

CONTEXT-DEPENDENT ABDUCTION AND RELEVANCE

Received on 25 June 2002

ABSTRACT. Based on the premise that what is relevant, consistent, or true may change from context to context, a formal framework of relevance and context is proposed in which

- contexts are mathematical entities
- each context has its own language with relevant implication
- the languages of distinct contexts are connected by embeddings
- inter-context deduction is supported by bridge rules
- databases are sets of formulae tagged with deductive histories and the contexts they belong to
- abduction and revision are supported by a notion of consistency of formulae and sets of formulae which are relative to a context, and which can, in turn, be seen as constituents of agendas.

1. Introduction

The goal of this paper is to give an account of how agendas can be pursued in the face of a not necessarily consistent input load of sentences. Each input sentence is taken to belong to a context, and whether an item of input is relevant to a given agenda has to be determined on line.

We start off by giving a multi-language model of contextual reasoning where each context has a separate language and deductive apparatus.

Distinct contexts are related by language embeddings and inter-context deduction rules. Deductive consequence is thus defined from sets of sentences belonging to different contexts.

We give an account of relevant implication which is geared towards mechanized abduction and revision, and which admits a strong cut theorem. Then we endow each local context logic with this notion of relevance, obtaining a combined logical system of relevance and context.

We proceed by giving the combined logic features which support abduction and revision of sets of contextual sentences. Specifically, a labelling scheme is introduced where sentences are decorated with labels containing

- a context which the sentence belongs to
- the deductive history of the sentence
- a sign to keep track of relevant deductions

An abductive process is defined, in which a database of contextual sentences is revised incrementally according to a load of input sentences. In case an input sentence is not consistent with the existing set of sentences, abduction yields in general several alternative consistent reconciliations of the old with the new.

We apply the abductive revision process to the pursuit of agendas. Taking an agenda as a set of contextual sentences to be satisfied by the accumulated database resulting from abductively relevant revision according to an input load, we obtain a priority criterion for selection among alternatives at each revision step: Those alternative databases are preferred which advance the agenda the most.

Finally, we remark on possible variations of the contextual model, in which the contexts under consideration correspond more closely to the databases being abduced, and point out connections with the work in [10].

2. Main Aspects Exemplified

Suppose an unmanned vehicle is pursuing some goals in a challenging environment, for instance on the surface of Mars.[1] Let us exemplify the main aspects of this paper by reference to the situation of such a vehicle:

Context The operation of a vehicle on Mars may be described logically in terms localized to specific contexts, such as power management, communication, locomotion, articulation of objects on the ground, time of (martian) day, etc: in this way the logical description of the operation of such a vehicle can be modularised. Such basic contexts combine into aggregate contexts, such as articulation of objects with time of day (cameras operate better in daylight), or locomotion with power management (aim for objects reachable on existing power reserve), etc. Our formulation of contexts in Section 3 is developed on the basis of [14].

Agenda An agenda consists of a finite set of goals to be achieved, and goals may be represented as logical propositions to be satisfied (land safely, identify objects, inspect each one, etc.). We present agendas in Section 4 as a variation on the general framework developed in [10].

Relevance Typically, goals are to be achieved once each. Conditions and prerequisites for performing actions are analogous to checklists: each prerequisite is to be checked exactly once. Logically speaking, this is modelled by a logic of relevant implication, where the order of nested antecedents does not matter, but the number of repeated antecedents does matter. Our treatment of relevance in Section 5 is an adaptation of Anderson–Belnap relevant logic as described in [10].

Abduction Unexpected events and adversity may be construed as an input load of new facts, not necessarily consistent with theories describing safe conditions for landing on Mars, establishing contact with mission control, moving about, picking up rocks, etc., nor with assumptions that the rover may have made on the basis of previous events. The task of reconciling new facts with previously accumulated theories and assumptions in the pursuit of agendas is formulated abductively in Sections 6 and 7, by means of a novel combination of context logic and relevant logic.

3. Contexts

It is a commonplace that all reasoning takes place within context of some sort or another, and in recent years there has been a thrust in AI towards formal modelling of contextual reasoning. Contexts turn up in various mathematical guises, and some application areas amenable to treatment by context-logical methods are the following:

- ontological encyclopaedia (Cyc) [15]
- federated databases [11]
- visual recognition [1]
- multi-agent belief [3]
- planning [6]
- lexical disambiguation [5]
- understanding metaphors [2]
- default reasoning [4]
- machine learning [7]

The list is unexhaustive and the references only indicative. It is presented to show the wide applicability of these methods. It is not our intention to focus on any particular application area in this paper. We are concerned with how contexts support an abductive reasoning process, and not so much concerned with what an individual context is.

3.1. *Context Languages*

Context may be seen as composed of bits and chunks. There can be atomic contexts, for example Cyc microcontexts, as well as larger contexts made up of smaller ones. How contexts combine to form new ones needs to be addressed, and one way to do that is the following.

Let a set Ψ of contexts be given a priori,[2] and let there be a binary operator $\oplus$ on Ψ such that $c \oplus d$ is a context whenever c and d are contexts.

For each context c let us assume a language Λ_c of propositions used to express facts in that context. Each language is taken to contain relevant implication $\rightarrow$ and falsity $\perp$, and in addition atoms and possibly other connectives.

DEFINITION 1 (Well formed formulae). If A is a formula in Λ_c, then c. A is a well formed formula, and A is called a c-formula.

When contexts combine, various issues arise:

- what is the language of the joint context?
- what are the true statements in the new context?
- are there more than one way of forming the same context?
- how do facts from one context influence those of another?

We take on the first issue by formalizing a translation between certain context languages:

DEFINITION 2 (Language embedding). For all contexts c, there is a partial function l_c that maps $u \oplus c$-formulae into u-formulae for arbitrary $u \in \Psi$.

The embedding from $u \oplus c$ to u describes how the context $u \oplus c$ is represented in the context u.

In [12, 14, 16] the properties of such systems of context are studied in some detail.

3.2. *Inter-Context Deduction*

Let us see how one moves between contexts composed by the $\oplus$ operator. Deduction is by the rules for relevant logic as explained in Section 5, extended with the following bridge rules:

$$\frac{u.l_c(A)}{u \oplus c.A} R_{up_c} \qquad \frac{u \oplus c.A}{u.l_c(A)} R_{dw_c}$$

These rules will form part of the basis of our abduction regime. They support relevant deduction within a family of contexts. Facts of context u can be exported to a richer context $u \oplus c$ by rule R_{up_c}, and reasoning can proceed there according to the local rules of $\Lambda_{u \oplus c}$. Results so obtained can be imported back into u by rule R_{dw_c} for further processing.

4. Agendas

In its most basic sense, as developed in [10], an agenda is a tuple $\langle E, N \rangle$ with N being an *endpoint* and E a set of *effectors*. Intuitively N is any realizable state of affairs and E any set of realizable conditions sufficient for the realization of N. An agenda thus charts a realizable future state of affairs in a causally structured world. In such a world, future realizables lie in wait for the emergence of conditions that achieve their realization. At this level of generality, an agenda is a function that maps effectors to endpoints.

Given the theoretical aims of [10], it is desirable to propose two variations of this basic idea. Firstly, since in some situations it is useful to identify agendas with agents, thus making it possible to speak of an agenda pursued by a practical agent, for example, a class of agendas-*for* is described. An agenda $\langle E, N\rangle$ *for* a practical agent **X** is an agenda in the basic sense, in whose endpoint N the agent is said to have an interest, and is said to be disposed toward their realization. In each case, it allowed that the interest expressed in the endpoint, and the disposition towards taking steps to bring it about, might be tacit.

With this variation at hand, it is possible to define a risk notion of *agenda-relevance*. Information is relevant for a cognitive agent when in processing it the agent is affected in ways that advance or close one or more of his agendas.

In the other variation developed in [10], a *sentential agenda* is a couple $\langle S^E, S^N\rangle$ of sentences denoting in order the effectors and the endpoints of an agenda $\langle E, N\rangle$. Sentential agendas are introduced for the similarity that they give to proofs. If we think of the conclusion of a proof as a deducible formula awaiting the realization conferred by a deduction, the analogy with sentential agendas is apt. Various features of proofs can then be imported into the account of agendas. (See Sections 8 and 9 below.)

In their use here, agendas are sentential agendas, but with this difference: They are here relativized to contexts.

In the formal part of [10] it is shown that for a base logic of agenda relevance the adaptation of Anderson–Belnap relevant systems presented here is adequate to the task of formal representation. This point is developed in the section to follow.

5. Relevance

We now turn to the structure of the local logics of relevant implication and deduction. They are adaptations of Anderson–Belnap relevant logic as described in [10] chapter 14. Briefly, each language has falsity $\bot$ and relevant implication $\to$, plus its own supply of propositional constants. There is also a set of atomic tags, and sets of tags are called labels. These are used to trace the deductive history of formulas in databases.

5.1. *Local Consequence Relations*

The following is a Hilbert system for relevant implication with $\to$ and $\bot$.

DEFINITION 3 (Hilbert system for relevant implication).

1. $A \to A$
2. $(A \to B) \to ((C \to A) \to (C \to B))$
3. $(A \to (B \to C)) \to (B \to (A \to (C))$
4. $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
5. $\bot \to A$
6. $\dfrac{\vdash A \to B}{\vdash (B \to C) \to (A \to C)}$
7. $\dfrac{\vdash A \quad \vdash A \to B}{\vdash B}$

The notion of deducibility $\vdash A$ is defined in the usual way. By axiom 3 the order of premises in relevant implication is immaterial:

THEOREM 1 (Permutation theorem for the Hilbert system).

$$\frac{\vdash (A_1 \to \ldots \to (A_n \to B)\ldots)}{\vdash \left(A_{\rho(1)} \to \ldots \to \left(A_{\rho(n)} \to B\right)\ldots\right)}$$

where ρ is any permutation of $\{1, \ldots, n\}$.

Proof. See [10]. □

On the strength of the permutation theorem, we can define deducibility from a multiset of formulae as follows:

DEFINITION 4 (Deducibility from a multiset of formulae).

Let $\Delta = \{A_1, \ldots, A_n\}$ be a multiset of wffs. Define $\Delta \vdash B$ as $\vdash (A_1 \to \ldots \to (A_n \to B)\ldots)$.

This notion of deducibility admits the following cut theorem:

THEOREM 2 (Cut theorem for the Hilbert system). *Let Δ, Γ be multisets of wffs.*

If $\Delta, A \vdash B$ and $\Gamma \vdash A$ then $\Delta \cup \Gamma \vdash B$.

Proof. See [10]. □

6. Goal-Directed System of Context and Relevance

We can now take the logics of relevant implication as our local context logics. A formula A derived with label α in context c will be denoted

$$\alpha : c.A$$

Since the soundness and completeness results of [14] carry over, we have at our disposal the full range of deductive machinery, including the labelled deduction rules of each local relevance logic and the global bridge rules for incremental augmentation and depletion of context.

A database Δ is a set of contextual sentences tagged with signed labels, the signs serving to signal availability for use in the abductive process:

$$\Delta = \{\pm\alpha : c.A\}$$

A contextual sentence $d.B$ is a consequence of a database Δ just in case it is a consequence of the contextual facts contained in it, disregarding their deductive histories:

$$\Delta \vdash d.B \quad \text{iff} \quad \{c.A | \pm\alpha : c.A \in \Delta \text{ for some } \alpha\} \vdash d.B$$

Notice the signed labels attached to contextual formulae in databases. These signs and labels will determine the value of the Success predicate defined below, and by extension the Abduce function which is defined further on. Signs and labels pertain only to the notion of consequence of databases, and do not come into play in individual deduction steps of relevance logic.

Let us give the joint logic a goal-directed formulation (see [8]) to facilitate the mechanisms of abduction and revision. To begin with, let us observe that every wff $c.B$ of the logic has the form $c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q)\ldots)$, where Q is atomic or $\bot$. The subformula Q is called the *head* of the formula, and $A_1, \ldots A_n$ comprise the *body*.

If we want to prove $c.Q$ from the database, we can search in a goal directed way, and look for c-clauses in the database with head Q, and then try to prove the wffs in the body. The subformulas in the body may have the form $Y = X_1 \rightarrow (X_2 \rightarrow \ldots \rightarrow (X_m \rightarrow R)\ldots)$ with R atomic or $\bot$. To prove $c.Y$ we use the deduction theorem, add $c.X_1, \ldots, c.X_m$ to the database and try to prove $c.R$. The goal directed discipline is defined operationally in terms of a predicate Success(Δ, $c.Q$) for a database Δ and atomic Q, which specifies whether, and if so how, $c.Q$ is reachable from Δ.

DEFINITION 5 (The Success predicate). (Q denotes an atom, and Q' denotes either Q or $\bot$)

(a) immediate success: Success($\Delta, c.Q$) = 1 if $\pm\alpha : c.Q' \in \Delta$ for some label α, and for all other clauses $\pm\beta : c.B \in \Delta$ the sign of β is +.
The + signals that those clauses are used as premises.

(b) immediate failure: Success(Δ, $c.Q$) = 0 if either of these two conditions hold:

(i) $\pm\ \alpha : c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q')\ldots) \notin \Delta$ for any sequence A_i.

(ii) $\pm\ \alpha : c.Q' \in \Delta$ but there are also some other remaining unused c-clauses $- : c.B \in \Delta$.

The presence of negative labels indicates that some clauses are not used as premises.

(c) deduction theorem case for success:

$$\begin{aligned}&\text{Success}(\Delta, c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q)\ldots)) = 1 \text{ if}\\ &\text{Success}(\Delta \cup \{-\alpha_1 : c.A_1, \ldots., -\alpha_n : c.A_n\}, c.Q) \ = 1.\end{aligned}$$

where the α_1 are fresh atomic labels.
The negative labels indicate that these clauses are still unused as premises.

(d) R_{up} case for success: Success(Δ, $c.Q$) = 1 if $c = u \oplus d$ for some contexts u, d, and Success(Δ, $u.l_d(Q)$) = 1.

(e) R_{dw} case for success: Success(Δ, $c.Q$) = 1 if $Q = l_d(A)$ for some context d and some $c \oplus d$-formula A, and

$$\text{Success}(\Delta, c \oplus d.A) = 1.$$

These two cases shift context according to the R_{up} and R_{dw} bridge rules.

(f) unification case for success: Success(Δ, $c.Q$) = 1 in at most n + 1 steps, if for some clause $\pm\alpha : c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q')\ldots) \in \Delta$, the following holds: Δ can be split into n parts $\Delta = \bigcup_i \Delta_i$ such that for each i Success(Δ', $c.A_i$) = 1 in at most n steps, where $\Delta' = \bigcup_i \Delta'_i$ and Δ'_i is obtained from Δ_i by switching declarative units $-\alpha : c.A$ into $+\alpha : c.A$ where they occur in Δ_i.
Switching sign from – to + means that the clause is relevantly taken into account.

(g) unification case for failure: Success (Δ, $c.Q$) = 0 in at most n + 1 steps, if for each clause $\pm\alpha : c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q')\ldots) \in \Delta$, the following holds: for each decomposition as defined in the previous case, there exists i such that Success (Δ'_i, $c.A_i$) = 0 in at most n steps.

This completes the definition of the Success predicate.

We now turn to abduction. Note that $\neg$ Success($\{\alpha : c.\,A\}$, $c.B \rightarrow A$), because in relevance logic $A, B \not\vdash A$. This example is one of the main non-theorems in relevance logic and it is important that it remains a non-theorem.

However, suppose we receive reliable information that $\Delta = \{\alpha : c.\,A\}$ should prove $c.B \rightarrow A$. This means that Δ is not exactly right and it really should be Δ'. How do we find Δ'? The process of abduction is supposed to help us do that.

Abduction is typically triggered by an unsuccessful attempt at deriving a contextual sentence from a database. Therefore, the abductive mechanism has steps corresponding to the cases in the definition of the Success predicate; each case of failure gives rise to an abductive step.

Therefore, the abduction process is a multifunction algorithm which takes a database Δ and a goal G such that $\Delta \not\vdash G$, and yields a family $\{\Delta_1', \Delta_2', \ldots\}$ of related databases (to Δ) such that for each Δ_i' we have $\Delta_i' \vdash G$. A fuller study of abduction is beyond the scope of this paper, but let us agree on a policy for the case $\Delta = \{\alpha : c.\,A\}$ and $G = c.B \rightarrow A$. We have the deduction theorem:

$$c.A \vdash c.B \rightarrow A \quad \text{iff} \quad c.A, c.B \vdash c.A$$

To make amends, we can delete $c.B$, but that policy has two weaknesses.

1. It is contrary to the spirit of relevance to delete $c.B$.
2. Since we need to modify $\Delta = \{\alpha : c.A\}$, so that $c.B \rightarrow A$ becomes provable, we would need to add to Δ the instruction to delete $c.B$ when it comes. Say, for example, add *Delete* $(c.B)$ to Δ. But then we need a logic of deletion, which is not a topic of the present article.

We can avoid these difficulties by reasoning as follows:

Since $c.B \rightarrow A$ is supposed to succeed from Δ and $c.\,A$ can be proved from $\Delta_1 \subseteq \Delta$ without $c.B$, it must be that $c.B$ is relevant to support the base Δ_1 which yields $c.A$. Hence we must add to our database the data $\{\alpha : c.B \rightarrow X \,|\, \alpha : c.X \in \Delta\}$. Thus the abduced database will be $\{\alpha : c.A, \alpha : c.B \rightarrow A\}$.

We also wish to be able to get

$$c.B \rightarrow A$$

from the database

$$\{\alpha : c.A, \beta : c.B \rightarrow A\}$$

but this still does not succeed because only one copy of A is used. One can of course add $\gamma : c.\,A \rightarrow (A \rightarrow A)$ to use the additional A, but we think it is better to pass to a logic where 'used' is determined by content rather than by form.

Now we give the formal definition of abduction.

DEFINITION 6 (The Abduce Function). The function $Ab+(\Delta, c.A)$ tells us what to add to the database to make $c.A$ succeed from Δ.

$Ab+(\Delta, c.A)$ is a set of alternative update actions, each of which changes Δ to a new database Δ' which proves $c.A$. We write it as

$$Ab^{+}(\Delta, c.A) = \{\Delta'_i\}$$

The definition of $Ab+$ is recursive on the computation stages of Success(Δ, $c.A$). When Success (Δ, $c.A$) = 1 we define Ab^{+} (Δ, $c.A$) = $\{\Delta\}$, otherwise the Δ_i' arise as follows:

1. if Success (Δ, $c.A$) = 0 by immediate failure: let $\Delta = \{\pm\alpha_1 : c_1.A_1, \ldots, \pm\alpha_n : c_n.A_n\}$ and let β be a new atomic label. Add $\Delta \cup \{-\beta : c.A_1 \rightarrow (\ldots \rightarrow (A_n \rightarrow Q)\ldots)\}$ to the set of abduced databases.
2. if the deduction theorem case for Success fails:

$$\text{Success}(\Delta, c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q)\ldots))$$
$$= \text{Success}(\Delta \cup \{-\alpha_1 : c.A_1, \ldots, -\alpha_n : c.A_n\}, c.Q) = 0$$

   then

$$Ab^{+}(\Delta, c.A_1 \rightarrow (A_2 \rightarrow \ldots \rightarrow (A_n \rightarrow Q)\ldots))$$
$$= Ab^{+}(\Delta \cup \{-\alpha_1 : c.A_1, \ldots, -\alpha_n : c.A_n\}, c.Q)$$

3. if the R_{up} case for Success fails: in other words, seeking to derive $c.Q$ from Δ we find Success (Δ, $u.\ l_d(Q)$) = 0 for all u, d such that $c = u \oplus d$. This may be because there are no such u, d, and then this case doesn't give rise to any abduction. Otherwise we select some u', d' such that $c = u' \oplus d'$, and add $\Delta \cup \{-\alpha : u'.l_{d'}(Q)\}$ to the set of abduced databases.
4. if the R_{dw} case for Success fails: we seek to derive $c.Q$ from Δ, but find Success(Δ, $c \oplus d.A$) = 0 for all d, A such that A is a $c \oplus d$-formula and $Q = l_d(A)$. Again failure may be because there are no d, A which satisfy the criteria, otherwise we select d', A' which do, and add $\Delta \cup \{-\alpha : c \oplus d'.A'\}$ to the set of abduced databases.

5. recall the unification case for failure: Success(Δ, $c.Q$) = 0, where we decompose $\Delta = \bigcup_i \Delta_i$ such that Success $(\Delta_i{}'c.A_i) = 0$ for certain databases Δ_i'. Include all databases abduced by

$$Ab^+(\Delta_i', c.Q)$$

for $1 \leq i \leq n$.

This completes the definition of abduction.

THEOREM 3 (Soundness of Abduction).

$$\textit{If } \Delta' \in Ab^+(\Delta,\ A) \textit{ then Success}(\Delta',\ A) = 1.$$

Proof. By induction on the recursive definition of $Ab+$. □

Note that the abduction function offers us several options for what to add to the database. We need a special logic for deciding which option to take. This logic we will call the *background logic* (for the abduction mechanism). We remark on the background logic for the pursuit of agendas in Section 10 of this paper. Background logics are treated in depth in [9]. For the moment the reader should note that the notion of relevance is involved in the background logic. We choose to abduce a set which is relevant. This now involves a circularity of concepts. To explain relevance we need a basic logic which involves the mechanism of abduction, which in turn needs the notion of relevance for its background logic.

This circularity is not a problem for our model construction. We begin with a simple notion of abduction, say $\mathcal{A}_1$, and use it to model relevance $\mathcal{R}_1$. Then we use $\mathcal{R}_1$ to improve the notion of abduction to $\mathcal{A}_2$, etc, and we get better and better notions of relevance and abduction.

7. Revision of Contextual Databases

With the deductive and abductive machinery in place, we can now develop a discipline of incremental revision of databases containing contextual formulae of relevance logic. Given a database

$$\Delta = \{\alpha_i : c_i \,.\, A_i\}$$

and a sequence of inputs

$$d_j \,.\, B_j$$

the task of revision is to reconcile Δ with d_j, B_j, consistently with the rules of relevance logic and intercontextual deduction, resulting in Δ'.

If $\Delta \cup \{-\alpha : d_j.B_j\}$, where α is a fresh label, is d_j-consistent, then we let

$$\Delta' = \Delta \cup \left\{-\alpha : d_j.B_j\right\};$$

otherwise we must find Δ' through some revision policy. If newer inputs have priority, then revision can be thought of as abducing $d_j.B_j$ from Δ, taking the revised database as some variant of the old one in which the new input is derivable. There are other revision strategies as well. We proceed to discuss how such a regime of database revision supports advancement and closure of contextual agendas.

8. Local Agendas

As indicated in Section 4, a local agenda is formalised as a triple $\langle c, \Sigma, N\rangle$, where c is a context, Σ is a finite sequence of contextual sentences in Λ_c called the *effectors* of the agenda, and N is the *endpoint* of the agenda. Intuitively, satisfying all the effectors is sufficient for reaching the endpoint. In this formulation, agendas rather resemble proofs, the effectors playing the roles of intermediate results or lemmata to be attained on the way to closure. Nevertheless, such proofs are not restricted to proceed exclusively within the local relevance logic of context c, since application of bridge rules is not precluded.

Agendas may be trivial, such as if the effectors are necessarily realized, and we may then represent it as $\langle c, \langle\rangle, N\rangle$. An agenda which is impossible to carry out may be written $\langle c, \langle\bot\rangle, N\rangle$.

An agent *advances* an agenda when he acts so as to satisfy an effector, and *closes* it when satisfying all the effectors. We'll study the effects on nontrivial agendas as a contextual database of facts are revised.

The sequence of effectors of an agenda admits various interpretations:

- there is the issue of whether the effectors must be satisfied in the order they are listed
- it may have repeated effectors, in which case there is the issue of whether each occurrence must be satisfied separately or not
- whether the agent can act in such a way as to satisfy an effector more times than it is listed in the agenda, or not

These issues remind us of substructural logic. If the effectors are required to be satisfied exactly the number of times they are listed, then closing an agenda is going to be similar in spirit to constructing a proof in relevant logic, for instance. Indeed, that is how we will interpret agendas here.

We shall focus on agendas where the effectors are essentially multisets, i.e. we disregard the order they are listed in, but we do care if an effector is mentioned once or more than once. Each effector must be ticked off as many times as it is listed in the agenda. With this understanding, the analogy between closing agendas and constructing proofs in relevance logic becomes clear: Compare an agenda $\langle c, \Sigma, N \rangle$ where $\Sigma = \langle E_1, \ldots, E_j \rangle$ is treated as a multiset, with

$$\vdash c.E_1 \rightarrow \big(E_2 \rightarrow \ldots \big(E_j \rightarrow N\big) \ldots\big).$$

Advancing the agenda by ticking off E_1 corresponds to using Modus Ponens (rule 7 of the Hilbert system) in the local relevance logic of Λ_c, resulting in

$$\vdash c.\big(E_2 \rightarrow \ldots \big(E_j \rightarrow N\big) \ldots\big),$$

and completing all the effectors results in

$$\vdash c.N.$$

Let us move towards the concrete, and stipulate the existence of:

- Finitely many databases $\{\Delta_m\}$, each containing contextual facts with deductive histories, of the form $\pm\alpha : c.A$.
- A load of inputs $\langle c_j.A_j \rangle_{j \geq 0}$ to the databases. An input is applied to all the databases in parallel.
- Finitely many agendas $\{\langle c_i, \Sigma_i, N_i \rangle\}$ to pursue, each in one specific context.

For instance, in the Martian rover we visited earlier, the databases would contain preprogrammed facts along with abduced statements. The load of inputs would correspond to data from sensors as well as instructions from mission control, and the agendas would represent the goals that the mission attempts to achieve.

The databases are thought of as alternative realizations of a finite initial segment of the input load. We can imagine the origin of the revision process to be a single empty database. What happens as an input arrives, is applied to each database, thus changing what is known in each context, and has effects on the agendas, can be described as a repeated cycle of the following actions:

1. for every agenda $\langle c_i, \Sigma_i, N_i \rangle$ which is closed by a database Δ_m, record this fact and remove the agenda from consideration

2. if there are no more open agendas, stop
3. the input at hand is $c_j.A_j$
4. revise each database Δ_m abductively according to $c_j.A_j$, resulting in a set $\{\Delta_{mn}\}$ of alternative databases
5. the new set of databases is $\bigcup_{mn}\{\Delta_{mn}\}$, including all the alternative revisions of every database.

9. Global Agendas

We may equally well conceive of agendas where the effectors are individually tagged with their own separate contexts. This corresponds to agendas of agents with a global perspective, spanning more than one context. In this case, agendas are pairs $\langle\Sigma, N\rangle$, where $\Sigma = \langle c_1.E_1, \ldots, c_j.E_j\rangle$ is a multiset of contextual sentences.

The pursuit of global agendas is by the same cycle of events as for local ones.

We note, however, that the analogy between closing global agendas and proving formulas of relevance logic does not immediately carry over from the local case. This is because our language does not contain relevant implication between formulas of different contexts. But we can tell the story differently, comparing the agenda

$$\big\langle\big\langle c_1.E_1, \ldots, c_j.E_j\big\rangle, N\big\rangle$$

with the desire for a proof of

$$\big\{c_1.E_1, \ldots, c_j.E_j\big\} \vdash N.$$

Now, advancing the agenda by satisfying effector $c_1.E_1$ on the one hand reduces the agenda to

$$\big\langle\big\langle c_2.E_2, \ldots, c_j.E_j\big\rangle, N\big\rangle$$

and on the other hand it can be seen as providing

$$\vdash c_1.E_1,$$

which by modus ponens and the cut theorem reduces the proof burden to

$$\big\{c_2.E_2, \ldots, c_j.E_j\big\} \vdash N.$$

Thus the analogy persists.

10. Priorities Among Abduced Databases

It may fairly be felt that maintaining an ever growing set of alternative databases as abduction is triggered by inputs, is overly luxurious. It is reasonable to reflect on the set of alternatives presented by abduction, and select among them, but how? To rephrase, what is the background logic of agenda pursuit?

The effectors in an agenda suggest an answer; let us prioritize those databases which advance the agenda the most. We formalize this in terms of a priority function

$$p(\Delta, \Gamma) = n,$$

which takes a database Δ and an agenda Γ as input, and returns a natural number n as the function value. Lower numbers correspond to better priorities, as per common convention.

This allows the set of alternative databases to be pruned during every cycle of revision, by retaining those databases or that database which is best according to the priority function. We may then formulate the pursuit of an agenda Γ according to a single database Δ as a cyclic process, repeating the following events until closure of the agenda:

1. if our agenda Γ is closed by the database Δ, stop
2. obtain input $c_j.A_j$
3. revise Δ abductively according to $c_j.A_j$, resulting in a set $\{\Delta_m\}$ of alternative databases
4. replace Δ by the database $\Delta' \in \{\Delta_m\}$ for which $p(\Delta',\Gamma)$ is minimal, breaking ties in some manner
5. advance the agenda Γ by removing effectors satisfied by the new database $\Delta = \Delta'$.

11. Variations on Context

As an alternative to the contextual model elaborated above, it is possible to view context as defined by the current database of sentences. This leads to a simplified formalism in some respects, but is vulnerable to problems in others. It is undesirable, for instance, to identify a context with a deductively closed set of sentences. With the formalism above, deductive closure is defined on sets of sentences belonging to different contexts, in such a way that projections onto individual contexts need not be deductively closed.

The characterization of contextual effects by sets of sentences is beyond the scope of this paper, but we refer the reader to [9] for a fuller treatment than is possible here.

12. Concluding Remarks

In conclusion, we have brought together methods from relevance logic, abduction, and context logic, in such a way as to motivate an approach to agendas by taking account of relevant aspects of an input load of contextualised sentences.

An area which invites investigation with methods similar to those we have employed here, is the logic of fictional discourse [13].

Notes

[1] At the time of writing, two such vehicles have in fact recently landed on Mars; one of them is operating normally and the other is not.

[2] Typographical convention: Lower-case Latin letters $a \dots e$ and $u \dots z$ denote contexts, upper-case Latin letters $A \dots Z$ denote formulae, lower-case Greek letters $\alpha \dots \omega$ denote labels, and upper-case Greek letters Γ, Δ, Θ, Λ, Π, Σ, Υ, Φ, Ψ, Ω denote various sets, multisets, and sequences.

References

1. Aisbett, J. and Gibbon, G. (1994): A tunable distance measure for coloured solid models, *Artif. Intell.* **65**(1), 143–164.
2. Barnden, J. A. and Lee, M. A. (1999): An implemented context system that combines belief reasoning, metaphor-base reasoning and uncertainty handling, in P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, and F. Castellani (eds.), *Modeling and Using Contexts: Proceedings of the Second International and Interdisciplinary Conference, CONTEXT'99*, Springer Berlin Heidelberg New York, pp. 28–41.
3. Benerecetti, M., Bouquet, P. and Ghidini, C. (1997): A multi context approach to belief report, in S. Buvač and L. Iwanska (eds.), *AAAI Fall 1997 Symposium on Context in KR and NL.* AAAI.
4. Besnard, P. and Schwab, T. (1993): A context-based framework for default logics, in R. Fikes and W. Lehnert (eds.), *Proceedings of the Eleventh National Conference on Artificial Intelligence*, American Association for Artificial Intelligence: AAAI Press, Menlo Park, California, pp. 406–411.
5. Buvač, S. (1996): Resolving lexical ambiguity using a formal theory of context, in K. van Deemter and S. Peters (eds.), *Semantic Ambiguity and Underspecification*, CSLI Lecture Notes.

6. Buvač, S. and McCarthy, J. (1996): Combining planning contexts, in A. Tate (ed.), *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative*, AAA Press, Menlo Park, California.
7. Edmonds, B. (2001): Learning appropriate contexts, in V. Akman, P. Bouquet, R. Thomason and R. A. Young (eds.), *Modeling and Using Context*, Springer Berlin Heidelberg New York, pp. 143–155.
8. Gabbay, D. and Olivetti, N. (2001): *Goal Directed Proof Theory*, Kluwer.
9. Gabbay, D. and Woods, J. (2005): *The Reach of Abduction: Insight and Trial*, volume 2 of *A Practical Logic of Cognitive Systems*, Elsevier; ISBN 0 444 51791 X, pp. 496.
10. Gabbay, D. and Woods, J. (2003): *Agenda Relevance: A study in Formal Paragmatics*, volume 1 of *A Practical Logic of Cognitive Systems*, Elsevier/North-Holland; ISBN 0 444 51385 X.
11. Ghidini, C. (1998): *A semantics for contexual reasoning: Theory and two relevant applications*, Phd dissertation, Dipartamento di Informatica e Studi Aziendali, Univesita degli Studi di Roma "La Sapienza", Roma.
12. Ghidini, C. and Giunchiglia, F. (2001): Local models semantics, or contexual reasoning = locality + compatibility, *Artif. Intell*. **127**(2) pp. 221–259.
13. Nossum, R. (2003): A contexual approach to the logic of fiction, in P. Blackburn, C. Ghidini, R. Turner and F. Giunchiglia (eds.), *Modeling and Using Context*, number 2680 in LNAI, Springer, pp. 233–244.
14. Nossum, R. and Serafini, L. (2002): Multicontext logic of fiction, in J. Calmet, B. Benhamou, O. Caprotti, L. Henoque and V. Sorge (eds.), *Artificial Intelligence, Automated Reasoning and Symbolic Computation*, volume 2385 of *LNAI*, Springer.
15. Reed, S. and Lenat, D. (2002): Mapping ontologies into Cyc. Technical Report Online, Cycorp Inc., http://www.cyc.com/doc/white_papers/mapping-ontologies-into-%cyc_v31.pdf.
16. Serafini, L. and Giunchiglia, F. (2001): ML system: A proof theory for contexts, *J. Logic Lang. Inf.* July.

DOV GABBAY
DFKI, Saarbrücken, Germany
King's College London, London, UK

ROLF NOSSUM
Agder University College
Dept. of Mathematics Gimlemeon
4604 Kristiansand, Norway
E-mail: rolf.nossum@hia.no

JOHN WOODS
University of British Columbia
Canada