Reasoning about Incomplete Agents[*]

Hans Chalupsky
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
hans@isi.edu

Stuart C. Shapiro
Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260
shapiro@cs.buffalo.edu

Abstract

We show how the subjective and nonmonotonic belief logic **SL** formalizes an agent's reasoning about the beliefs of incomplete agents. **SL** provides the logical foundation of SIMBA, an implemented belief reasoning system which constitutes part of an artificial cognitive agent called Cassie. The emphasis of SIMBA is on belief ascription, i.e., on governing Cassie's reasoning about the beliefs of other agents. The belief reasoning paradigm employed by SIMBA is simulative reasoning. Our goal is to enable Cassie to communicate with *real* agents who (1) do not believe all consequences of their primitive or base beliefs, (2) might hold beliefs different from what Cassie views them to be, and (3) might even hold inconsistent beliefs. **SL** provides a solution to the first two problems and lays the groundwork to a solution for the third, however, in this paper we will focus only on how agent incompleteness can be handled by integrating a belief logic with a default reasoning mechanism. One possible application of **SL** and SIMBA lies in the area of user modeling. For example, Cassie could be in the role of an instructor who, among other things, has to deal with the incomplete beliefs of her students.

Introduction

SIMBA, an acronym for **sim**ulative **b**elief **a**scription, is an implemented belief reasoning system which constitutes part of an artificial cognitive agent who we call Cassie. Its main concern is the formalization of various aspects of belief ascription, i.e., it forms the machinery with which Cassie can reason about the beliefs of other agents. SIMBA's logical foundation is SIMBA Logic, or **SL**, which is a fully intensional, subjective, nonmonotonic belief logic.

It is our long-term goal to give Cassie the ability to communicate with other agents such as humans in natural language, thus we have to make sure that she can deal with real agents. In the design of a belief logic to describe Cassie's reasoning we are faced with at least three major challenges: (1) Real agents are incomplete, i.e., they do not believe all consequences of their primitive or base beliefs, (2) Cassie's beliefs about these agents might be incorrect requiring her to revise her beliefs, and (3) they might hold inconsistent beliefs. **SL** is a logic that provides solutions to the first two problems and lays the groundwork to a solution for the third, but in this paper we will only describe how **SL** can handle incomplete agents by incorporating a default reasoning mechanism into a belief logic. The other aspects of **SL** are described in (Chalupsky 1995). One possible application of **SL** and SIMBA lies in the domain of user modeling. For example, Cassie could be in the role of an instructor, who, among other things, has to deal with the incomplete beliefs of her students.

Incomplete Agents

When Cassie reasons about the beliefs of some real agent she has to take into account that real agents are incomplete. Even if all of Cassie's beliefs about the beliefs of the agent are correct, a consequence of these beliefs realizable by Cassie might be one that the agent has not yet concluded. A real-life example of such a situation is teaching. Many times a teacher teaches the basics of some subject and assumes that the "obvious" conclusions have been drawn by the students, only to find out later at an exam that the assumption was obviously wrong.

In slightly more formal terms, if Cassie believes that Oscar believes P and that $\mathsf{P} \Rightarrow \mathsf{Q}$, it makes sense for her to assume that he also believes Q. But then, he might not. This failure of logical consequence in belief contexts has troubled researchers for a long time. Most standard logics of knowledge or belief solve the problem by either avoiding it (e.g., syntactic logics), or by idealizing agents (e.g., modeling them as logically omniscient). Various attempts have been made to overcome some of these shortcomings of standard treatments, for example, (Levesque 1984; Konolige

[*] This is a prelimiary version of: Hans Chalupsky and Stuart C. Shapiro, Reasoning about Incomplete Agents, *Proceedings of the Fifth International Conference on User Modeling (UM-96)*, 1996, in press. All quotes should be from, and all citations should be to the published version.

1986; Fagin & Halpern 1988; Lakemeyer 1990). However, the success is always achieved at considerable cost. The resulting logics either restrict certain forms of inference, or trade one idealization for another, or make somewhat unintuitive assumptions about the nature of agents' reasoning, thus we think none of them are very well suited as a formal foundation of Cassie's reasoning.

Belief Representation

We view Cassie's "mind" as a container filled with a variety of objects, some of which constitute her beliefs. These beliefs are represented by sentences of $L_{\mathbf{SL}}$, the language of **SL**. $L_{\mathbf{SL}}$ very much looks like the language of standard first-order predicate calculus, but it has a very different semantics. Its sentences are not true or false statements *about* Cassie's beliefs, but they *are* Cassie's beliefs which is why we call **SL** a *subjective* logic. $L_{\mathbf{SL}}$ is primarily a language of proposition-valued function terms such as, for example, $\mathsf{Loves}(\mathsf{John}, \mathsf{Mary})$, whose denotation is intended to be the proposition *John loves Mary*. A sentence is formed by prefixing a proposition term with an exclamation mark as in $!\mathsf{Loves}(\mathsf{John}, \mathsf{Mary})$. The semantics of a sentence is that the agent whose mind contains it (usually taken to be Cassie) believes the proposition denoted by the proposition term.

Cassie's beliefs about the beliefs of other agents are expressed by sentences of the kind $!\mathsf{B}(\mathsf{Sally}, \mathsf{Loves}(\mathsf{John}, \mathsf{Mary}))$. The proposition term of such a sentence is simply a nested application of proposition-valued functions but not a higher-order relation. The nesting can go to arbitrary depth to account for propositions such as *John believes that Sally believes that I believe that*.... A full motivation and formal specification of the syntax and semantics of **SL** is given in (Chalupsky & Shapiro 1994). It should be pointed out that even though Cassie's beliefs might be viewed as a database of belief sentences, our model is not the database approach to belief representation. To form beliefs about the beliefs of other agents Cassie has the full logical arsenal at her disposal, including negation and disjunction. Via introspection she can even have beliefs about her own beliefs, for example, $!\neg\mathsf{B}(\mathsf{I}, \mathsf{Equiv}(\mathsf{P}, \mathsf{NP}))$.

Reasoning as Logical Inference

While the syntax and semantics of **SL** provide the formal basis of Cassie's belief representation, we model her reasoning as logical inference according to a deductive system $D_{\mathbf{SL}}$. An implementation of a proof procedure for $D_{\mathbf{SL}}$ serves as her actual reasoning engine. $D_{\mathbf{SL}}$ is a natural deduction system which consists of a part very similar to natural deduction systems for predicate calculus, and a part that deals with belief reasoning. We will introduce $D_{\mathbf{SL}}$ by way of example as we go along.

The focus of **SL** and SIMBA is on the formalization of Cassie's reasoning about the beliefs of other agents. The reasoning paradigm we use for that is *simulative reasoning* (Creary 1979; Chalupsky 1993; Barnden *et al.* 1994), a mechanism in which Cassie hypothetically assumes the beliefs of some other agent as her own and then tries to infer conclusions from these hypothetical beliefs with help of her own reasoning skills.

Notational Conventions: Sans serif indicates object language terms, for example, $\mathsf{Smart}(\mathsf{Oscar})$, and $italics$ indicate meta-variables ranging over such terms, for example, $p \vee q$. B is the belief function and I is Cassie's ego constant. All object and function constants start with an uppercase letter; variables are written in lower case. Simulation contexts (explained below) are drawn with double vertical lines, hypothetical contexts only have single lines, contexts that could be either have one single and one double line. To abbreviate sentences that appear in reasoning contexts we use their step numbers as aliases. For example, if the line with step number 5 contains the sentence $!\mathsf{Smart}(\mathsf{Oscar})$, then we can use 5 as an abbreviation wherever we want to refer to that sentence.

An Example

Figure 1 shows an example in which Cassie is imagined to be a teacher of basic complexity theory. Oscar is one of her students of whom she assumes that from the material presented in class he has arrived at the following obvious (to her) conclusion: If the complexity classes P and NP are equivalent then the NP-complete SAT problem is computable in polynomial time.

Here is a quick introduction to $D_{\mathbf{SL}}$ derivations: The main structuring device are *inference* or *reasoning contexts* which are drawn as boxes. They come in two kinds: (1) Simulation contexts to simulate a particular agent's reasoning, and (2) hypothetical contexts to carry out hypothetical reasoning. Every context has a name, a pointer to a parent context (or $\top$ for the top-level context) and the agent whose reasoning is carried out listed in the top field. Every application of an inference rule adds another sentence to one of the open contexts (there is no order requirement). To follow a derivation one follows the step numbers on the left of the context boxes in sequence. This scheme is very close to the actual implementation.

The top-level simulation context in the example represents Cassie's primary frame of mind. Every sentence in that context represents (or is) one of her beliefs. Steps 1 to 5 display her beliefs about Oscar's grasp of complexity theory: 1 (the sentence in step 1) represents her belief that he believes that if two classes are equivalent every element of one class is also element of the other. 1 is followed by

	Cassie ($\top$), I			
1	$!B(Oscar, \forall c_1, c_2, e\,(Cl(c_1) \wedge Cl(c_2) \wedge \wedge Equiv(c_1, c_2) \wedge In(e, c_2)) \Rightarrow In(e, c_1))$,	hyp,	$\{1\}$	H
2	$!B(Oscar, Cl(P))$,	hyp,	$\{2\}$	H
3	$!B(Oscar, Cl(NP))$,	hyp,	$\{3\}$	H
4	$!B(Oscar, \forall p\, In(p, P) \Rightarrow PTime(p))$,	hyp,	$\{4\}$	H
5	$!B(Oscar, In(SAT, NP))$,	hyp,	$\{5\}$	H
				open Oscar
18	$!B(Oscar, Equiv(P, NP) \Rightarrow PTime(SAT))$,	der,	$\{1, 2, 3, 4, 5\}$	BI 17
19	$!Smart(Oscar)$,	hyp,	$\{19\}$	H
	...			Exam
99	$!\neg B(Oscar, Equiv(P, NP) \Rightarrow PTime(SAT))$,	hyp,	$\{99\}$	H

	Oscar (Cassie), Oscar, $6/\{1\}, 7/\{2\}, 8/\{3\}, 9/\{4\}, 10/\{5\}$			
6	$!\forall c_1, c_2, e\,(Cl(c_1) \wedge Cl(c_2) \wedge Equiv(c_1, c_2) \wedge \wedge In(e, c_2)) \Rightarrow In(e, c_1)$,	hyp,	$\{6\}$	SH 1
7	$!Cl(P)$,	hyp,	$\{7\}$	SH 2
8	$!Cl(NP)$,	hyp,	$\{8\}$	SH 3
9	$!\forall p\, In(p, P) \Rightarrow PTime(p)$,	hyp,	$\{9\}$	SH 4
10	$!In(SAT, NP)$,	hyp,	$\{10\}$	SH 5
				open OscarHyp
17	$!Equiv(P, NP) \Rightarrow PTime(SAT)$,	der,	$\{6, 7, 8, 9, 10\}$	$\Rightarrow$I 16

	OscarHyp (Oscar), Oscar			
11	$!Equiv(P, NP)$,	hyp,	$\{11\}$	H
12	$!(Cl(P) \wedge Cl(NP) \wedge Equiv(P, NP) \wedge \wedge In(SAT, NP)) \Rightarrow In(SAT, P)$,	der,	$\{6\}$	$\forall$E 6
13	$!(Cl(P) \wedge Cl(NP) \wedge Equiv(P, NP) \wedge \wedge In(SAT, NP))$,	der,	$\{7, 8, 10, 11\}$	$\wedge$I 7,8,10,11
14	$!In(SAT, P)$,	der,	$\{6, 7, 8, 10, 11\}$	$\Rightarrow$E 12,13
15	$!In(SAT, P) \Rightarrow PTime(SAT)$,	der,	$\{9\}$	$\forall$E 9
16	$!PTime(SAT)$,	der,	$\{6, 7, 8, 9, 10, 11\}$	$\Rightarrow$E 14,15

Figure 1: Oscar's reasoning is incomplete

a hyp origin tag and by its origin set or *hypothesis support* (this support structure is derived from (Martins & Shapiro 1988)). Since 1 is a hypothesis, its origin set just contains the sentence itself. The H on the right of the box indicates that this sentence was introduced with the rule of hypothesis which is the only means to add new, otherwise unjustified beliefs to a reasoning context. Cassie also believes that Oscar believes that P and NP are classes, that for every instance of P there is an algorithm that solves it in polynomial time, and that SAT is in NP.

What follows is a simulation of Oscar's reasoning in the Oscar context. It is not really necessary to follow this example in all its detail, it is just supposed to present the general flavor of our system and show the incompleteness problem. In the Oscar context Cassie assumes the object propositions of her beliefs about Oscar as her own beliefs to simulate his reasoning. An exact definition of the simulation rules will be given later. Since the sentence in question is an entailment, she has to perform hypothetical reasoning in the context OscarHyp to derive it. When sentences are derived they get a der origin tag, and their hypothesis support is in most cases computed by simply taking the union of the premise supports. Finally, Cassie derives 17 and ascribes it to Oscar as 18 in her top-level context. The hypothesis support of 18 was computed with help of the map stored at the top of the Oscar context. In this example we view this last belief introduction step as a sound inference rule that is not different from rules such as Modus Ponens, etc.

A few weeks later Cassie gives an exam. While she grades Oscar's exam she finds out – much to her dismay – that he obviously does not believe sentence 18, otherwise he would have solved one of the exam problems correctly (this is especially disappointing in light of 19). Cassie's new belief is introduced in step 99, but that directly contradicts the simulation result of step 18. What is she supposed to believe now?

If we do not take special action now, Cassie will be able to derive and believe *any arbitrary sentence* by using contradiction elimination. It is certainly completely undesirable to have Cassie's own top-level reasoning collapse just because one of the agents she knows about is incomplete. There are two scenarios that can explain the resulting contradiction:

1. Some of Cassie's initial belief hypotheses about Oscar's beliefs are incorrect. This case needs to be handled by belief revision which is supported by **SL** but outside the scope of this paper.

2. Oscar's reasoning is incomplete. It is easily imaginable that each of Cassie's belief hypotheses about Oscar's beliefs is directly "observable" by reading Oscar's exam paper, only Oscar's belief in the obvious conclusion is not manifested anywhere, even worse, it is directly observable that he does not believe the conclusion in question. This case cannot be solved by belief revision, because there is nothing to revise. All the initial beliefs are correct and should not be retracted. The problem is that Oscar's reasoning is incomplete, and what needs to be done is to block the incorrect simulation result in light of the striking evidence to the contrary.

Simulation Results are Default Conclusions

Our solution to the problem above is to treat simulation results as *default* conclusions. A default conclusion can be shadowed if it contradicts any belief based solely on proper belief hypotheses.

To handle the default character of simulation results at the logic level we introduce the concept of a *simulation assumption*. A simulation assumption is a special kind of hypothesis that is justified by a derivation from a set of proper hypotheses. In a sense an assumption is a hermaphrodite, because it is hypothesis and derived sentence simultaneously. This characterization of an assumption was introduced by Cravo and Martins (1993) in their formalization of default reasoning, and the following treatment owes a great deal to their work.

In the example above we assumed the proposition of every derivable sentence to also be believable. Thus, believability was a monotonic property. Using the concept of simulation assumptions, we can define a nonmonotonic variant of believability based on the primitive notion of derivability. This new version will allow us to shadow simulation results as well as handle mutually contradicting simulations.

Formalization

Below are those inference rules of $D_{\mathbf{SL}}$ that are particularly sensitive to the distinction between hypotheses and assumptions. In every rule it is assumed that l is the step number of the immediately prior inference step, that the sentence at line $l+1$ is the conclusion, and that all other sentences are premises. A new assumption support element is added to the right of the hypothesis support of every sentence. It contains the set of simulation assumptions on which the derivation of a particular sentence is based. In every inference step hypothesis and assumption supports are combined separately. τ, ω, and α are meta-variables (indices are used where necessary), where τ ranges over origin tags, ω over hypothesis supports, and α over assumption supports.

Negation Introduction ($\neg$I): From a contradiction that is solely based on hypotheses we can deduce the negation of any element of $\omega \cup \{!h\}$, i.e., the negation of any hypothesis on which the derivation of the contradiction was based. Following Cravo and Martins we will call such a contradiction a *real* contradiction as opposed to an *apparent* contradiction which is partly based on assumptions. No equivalent rule exists for apparent contradictions.

$$\begin{array}{r|llll||l}
\hline
 & C\ (\ldots), a & & & & \\
\hline
 & \ldots & & & & \\
m & !p \wedge \neg p, & \tau, & \omega \cup \{!h\}, \{\} & & \\
 & \ldots & & & & \\
l+1 & !\neg h, & \mathsf{der}, & \omega \setminus \{!h\}, \{\} & & \neg\mathrm{I}\ \ m \\
\hline
\end{array}$$

Simulation Hypothesis (SH, SH'): The rule of simulation hypothesis comes in two variants. SH: If the belief sentence in the parent context is not based on any assumptions

$$\begin{array}{r|llll||}
\hline
 & C\ (\ldots), a & & & \\
\hline
 & \ldots & & & \\
m & !\mathsf{B}(b, p_i), & \tau_i, & \omega_i, \{\} & \\
 & \ldots & & & \\
m' & !\mathsf{B}(b, p_j), & \tau_j, & \omega_j, \alpha_j \neq \emptyset & \\
 & \ldots & & & \\
\hline
\end{array}$$

$$\begin{array}{r||lll||l}
\hline
 & D\ (C), b, & & & \\
 & \ldots !p_i/\omega_i\{\}, \ldots !p_j/\omega_j\alpha_j & & & \\
\hline
 & \ldots & & & \\
l{+}1 & !p_i, & \mathsf{hyp}, & \{!p_i\}, \{\} & \mathrm{SH}\ \ m \\
 & \ldots & & & \\
l'{+}1 & !p_j, & \mathsf{sim}, & \{\}, \{!p_j\} & \mathrm{SH}'\ \ m' \\
\hline
\end{array}$$

then its object proposition will be introduced as a proper hypothesis in the simulation context. SH': If the parent sentence did depend on assumptions, then the object proposition will be introduced as an *a priori* simulation assumption which is indicated by the new sim origin tag and the assumption origin set. In both cases the proper mapping between origin sets of the parent sentence and the simulation hypothesis is stored at the top of the simulation context.

Belief Introduction (BI**):** This is the only rule of $D_{\mathbf{SL}}$ that actually derives simulation assumptions. Whenever some sentence $!q$ is derived in a simulation context for some agent b, and the hypothesis support of the new sentence is contained in the set of simulation hypotheses $shyps(D, m)$[1] introduced up to that point, then we can introduce the belief sentence $!\mathsf{B}(b, q)$ as a simulation assumption in the parent context. The new belief sentence gets a sim origin tag to identify it as an assumption, and its origin sets are computed by mapping the origin set of $!q$ back into the parent context via the map stored at the top of the simulation context (we are sloppy here, since the possibility of multiple derivations requires a slightly more complicated mapping scheme). Finally, $!\mathsf{B}(b, q)$ gets added to its own assumption support which makes it into the dual gender entity that is half hypothesis and half derived result.

$$\begin{array}{r|l||l}
\hline
 & C\,(\ldots),\, a & \\
\hline
 & \ldots & \\
l{+}1 & !\mathsf{B}(b,q),\ \mathsf{sim},\ \bigcup\limits_{!p_i\in\omega\cup\alpha}\omega_i,\ \{!\mathsf{B}(b,q)\}\cup\bigcup\limits_{!p_i\in\omega\cup\alpha}\alpha_i & \mathrm{BI}\,m \\
\hline
\end{array}$$

$$\begin{array}{r||l||}
\hline
 & D\,(C),\, b, \\
 & !p_1/\omega_1\alpha_1, \ldots, !p_n/\omega_n\alpha_n \\
\hline
 & \ldots \\
m & !q, \quad \tau, \quad \omega \subseteq shyps(D, m),\ \alpha \\
 & \ldots \\
\hline
\end{array}$$

As motivated above, the top-level reasoning context of a $D_{\mathbf{SL}}$ derivation models Cassie's primary state of mind. Over time sentences will get added to that context either as derived results or as hypotheses, and some hypotheses will also get removed as a result of belief revision. Thus the set of believable sentences changes over time. To get a handle on these changes we will look at individual snapshots of reasoning contexts called *belief states*:

Def 1 *A* **belief state** *σ is a quadruple $\langle\!\langle a, H, A, \sigma_p\rangle\!\rangle$, where (1) "$a$" is a reasoning agent, (2) H is a set of sentences taken to be hypotheses, (3) A is a set of sentences taken to be a priori simulation assumptions, and (4) σ_p is either $\top$ or a parent or simulator belief state.*

The support of a sentence can be viewed as a summary of things necessary to derive it. In the following we will make heavy use of sentence supports, hence, we define the following notation:

Def 2 *A* **supported sentence** *$\langle !p, a/?, \tau, \omega, \alpha\rangle$ is a quintuple, where (1) $!p$ is an arbitrary $L_{\mathbf{SL}}$-sentence, (2) $a/?$ is either some agent "a" or the unspecified agent "$?$", (3) τ is an origin tag which can be either* hyp, der, *or* sim, *(4) ω is the set of hypotheses, and (5) α is the set of simulation assumptions on which the derivation of $!p$ is based.*

Adding the agent element to the support is necessary, since inference rules such as introspection (not presented here) encode the agent of a reasoning context in the derived sentence. If no such rule was used in the derivation of a sentence its support contains the unspecified agent "?".

Now we are ready to define a derivation relation between belief states and supported sentences:

Def 3 *$\langle\!\langle a, H, A, \sigma_p\rangle\!\rangle \vdash \langle !q, a/?, \tau, \omega, \alpha\rangle$ iff there exists a derivation of the form such that $hyps(D, l) \subseteq H$ and $shyps'(D, l) \subseteq A$.*[2]

$$\begin{array}{||l||}
\hline
C\,(\top),\, \mathsf{I} \\
\hline
\ldots \\
 \\
\hline
\end{array}
\quad \ldots \quad
\begin{array}{r||l||}
\hline
 & D\,(\ldots),\, a \\
\hline
 & \ldots \\
l & !q, \quad \tau, \quad \omega, \alpha \\
\hline
\end{array}
\quad \ldots$$

This definition of derivability is applicable to belief states describing top-level contexts as well as nested simulation contexts. Often it is also convenient to work with deductive closures:

Def 4 *Let $\sigma = \langle\!\langle a, H, A, \sigma_p\rangle\!\rangle$. Its* **deductive closure** *is the set $\sigma^\star = \{\langle !q, a/?, \tau, \omega, \alpha\rangle \mid \sigma \vdash \langle !q, a/?, \tau, \omega, \alpha\rangle\}$.*

The following theorem states an important fact about support computation: The hypotheses and assumptions collected in the support of a derived sentence are sufficient (though not necessary) to derive it.

The 1 *If $\langle\!\langle a, H, A, \sigma_p\rangle\!\rangle \vdash \langle !q, a/?, \tau, \omega, \alpha\rangle$ then $\langle\!\langle a, \omega, A \cap \alpha, \sigma_p\rangle\!\rangle \vdash \langle !q, a/?, \tau, \omega, \alpha\rangle$.*

The main enterprise below is to define sets of *reasonable assumptions* motivated by a particular belief state. Following standard default logic terminology we will call such a set an *extension* (Reiter 1980).

Def 5 *Let $\sigma = \langle\!\langle a, H, A, \sigma_p\rangle\!\rangle$. Any set E such that $E \subseteq \{\langle !p, a/?, \mathsf{sim}, \omega, \alpha\rangle \in \sigma^\star \mid \alpha \neq \emptyset\}$ is called an* **extension set** *for σ.*

Thus, any set of (not necessarily reasonable) simulation assumptions derivable from a belief state can be an extension

[1] $shyps(D, m)$ collects all sentences introduced into context D via the rule of SH up to step m.

[2] $hyps(D, l)$ collects all sentences introduced into context D via H or SH up to step l, and $shyps'$ collects all a priori simulation assumptions introduced via SH'.

set. Note, that the a priori assumptions A are always trivially derivable from σ.

Def 6 *Let $\sigma = \langle\!\langle a_\sigma, H, A, \sigma_p \rangle\!\rangle$ and E an extension set for σ. A supported sentence $\langle !p, a_s/?, \tau, \omega, \alpha \rangle$ renders its proposition p* **believable** *in σ extended by E iff:*

1. *a_σ and $a_s/?$ are compatible, i.e., they are either identical or $a_s = ?$, and*
2. *$\omega \subseteq H$, and*
3. *$\alpha \subseteq H \cup sent(E)$.*[3]

Regardless of what the *correct* extensions of a belief state will turn out to be, we are now ready to define the following *degrees of believability* of a sentence relative to a belief state and a set of arbitrary extension sets:

Def 7 *Let $\sigma = \langle\!\langle a_\sigma, H, A, \sigma_p \rangle\!\rangle$ and $\mathcal{A}(\sigma)$ a set of arbitrary extension sets for σ. A supported sentence $\langle !p, a_s/?, \tau, \omega, \alpha \rangle$ renders its proposition p*

- **certain**, *written with a bold exclamation mark $\mathbf{!}p$, iff its support renders it believable in belief state $\sigma = \langle\!\langle a_\sigma, H, \emptyset, \sigma_p \rangle\!\rangle$,*
- **plausible**, *written $\tilde{\mathbf{!}}p$, iff either it is certain, or $\mathcal{A}(\sigma) \neq \emptyset$ and for every $E \in \mathcal{A}(\sigma)$ its support renders it believable in σ extended by E,*
- **possible**, *written $\approx\!\!\mathbf{!}p$, iff either it is plausible, or $\mathcal{A}(\sigma) \neq \emptyset$ and for at least one $E \in \mathcal{A}(\sigma)$ its support renders it believable in σ extended by E,*
- *or* **unbelievable**, *written $\oplus p$, if its support does not render it possible.*

The symbols $\mathbf{!}$, $\tilde{\mathbf{!}}$, $\approx\!\!\mathbf{!}$, and $\oplus$ are intended to illustrate "certain", "approximate", "very approximate", and "out". They are annotations used to indicate the degree of believability of a particular sentence in a particular derivation. The plain exclamation mark as in $!p$ only indicates that the sentence was derivable according to the inference rules of the logic. It classifies the proposition p as a *belief candidate*, but whether Cassie actually believes p depends on its believability according to the current state of her various reasoning contexts.

Def 8 *A belief state $\langle\!\langle a, H, A, \sigma_p \rangle\!\rangle$ is* **consistent** *iff $\langle\!\langle a, H, A, \sigma_p \rangle\!\rangle \not\vdash \langle !p \wedge \neg p, a/?, \tau, \omega, \{\} \rangle$ for any p.*

A consistent belief state does not support any real contradictions.

Before we go on to formally define extensions, let us quickly summarize what makes a simulation assumption reasonable relative to a belief state:

1. It should be motivated by the belief state, i.e., derivable from it.
2. It should not contradict any of the belief state's hypotheses or any of their sound consequences.
3. It should not contradict any of the other reasonable assumptions motivated by the belief state.

Rather than adapting Reiter's (1980) fixed point definition for extensions, we follow Cravo and Martins and define them in two steps: (1) We find the set of simulation assumptions that each individually are reasonable for a particular belief state without checking for any possible conflicts with other assumptions. Such a set will be called a *prima facie extension*, because prima facie it could be an extension. (2) We partition a prima facie extension into maximal consistent subsets to form the proper extensions. The maximality criterion ensures that we wind up with the smallest number of extensions possible.

Def 9 *Let $\sigma = \langle\!\langle a, H, A, \sigma_p \rangle\!\rangle$. Its* **prima facie extension** *$P(\sigma)$ is the set $\{\langle !p, a/?, \mathsf{sim}, \omega, \alpha \rangle \in \sigma^\star \mid \langle\!\langle a, H \cup \alpha, \{\}, \sigma_p \rangle\!\rangle$ is consistent$\}$.*

For elements of the prima facie extension all that is necessary is that they and all the assumptions they depend on could be added to the belief state as hypotheses without leading to a real contradiction.

Cor 1 *If a belief state σ is inconsistent then $P(\sigma) = \emptyset$.*

Since only consistent belief states have interesting extensions, we will from now on always assume that the belief states we work with are consistent.

Def 10 *Let $\sigma = \langle\!\langle a, H, A, \sigma_p \rangle\!\rangle$, and let the simulation hypotheses available for the direct simulation of some agent b be given by the following sets:*

$$
\begin{aligned}
H^\star_{b,\sigma} &= \{!p \mid \sigma \vdash \langle !\mathsf{B}(b,p), a/?, \tau, \omega, \{\} \rangle\} \\
A^\star_{b,\sigma} &= \{!p \mid \sigma \vdash \langle !\mathsf{B}(b,p), a/?, \tau, \omega, \alpha \rangle, \alpha \neq \emptyset\}
\end{aligned}
$$

Then $\sigma_b = \langle\!\langle b, H^\star_{b,\sigma}, A^\star_{b,\sigma}, \sigma \rangle\!\rangle$ is the **simulation belief state** *for agent b in σ.*

The simulation belief state for some agent b specifies the set of hypotheses and assumptions that can be introduced into the simulation context for that agent via the rules SH and SH$'$.

Extensions are intended to partition the simulation assumptions in the deductive closure of a belief state into subsets of reasonable assumptions. What is reasonable is defined in terms of derivability of certain sentences in simulation contexts at arbitrary depths. Even if a belief state σ contains only a finite set of hypotheses, there is no upper bound to the level of nesting of simulation contexts used to derive the elements of its deductive closure, since, for example, hypothetical reasoning can introduce arbitrarily nested belief sentences. For this reason we define unrestricted extensions iteratively, thus considering deeper and deeper nested simulations with every iteration.

[3] *sent* is a projection function that selects the plain sentences from a set of supported sentences.

Def 11 *Let S, T be arbitrary sets and P a predicate. S is a* **maximal subset** *of T such that $P(S)$ iff $S \subseteq T$ and $P(S)$ is true and for any $e \in (T \setminus S)$ $P(S \cup \{e\})$ is false.*

Def 12 *Let $\sigma = \langle\langle a, H, A, \sigma_p \rangle\rangle$. Its* **extensions** *$\mathcal{E}(\sigma)$ are defined incrementally with $\mathcal{E}_i(\sigma)$ referring to their state at iteration $i, i \geq 1$:*

$\mathcal{E}_i(\top) = \{\{\}\}$ for all $i \geq 1$.

$\mathcal{E}_1(\sigma) = \{E \subseteq P(\sigma)\}$ where each E is a maximal subset of $P(\sigma)$ such that

1. *$\langle\langle a, H \cup \mathit{sent}(E), \{\}, \sigma_p \rangle\rangle$ is consistent, and*
2. *$\bigcup_{s \in E} \alpha(s) \subseteq \mathit{sent}(E)$,[4] i.e., E is closed.*

$\mathcal{E}_{i+1}(\sigma) = \{E \subseteq P(\sigma)\}$ where each E is a maximal subset of $P(\sigma)$ such that

1. *$\langle\langle a, H \cup \mathit{sent}(E), \{\}, \sigma_p \rangle\rangle$ is consistent, and*
2. *$\bigcup_{s \in E} \alpha(s) \subseteq \mathit{sent}(E)$, and*
3. *there exists an extension $E_p \in \mathcal{E}_i(\sigma_p)$ such that $\bigcup_{s \in E} \alpha(s) \cap A \subseteq \{!q \mid !\mathsf{B}(b, q) \in \mathit{sent}(E_p)\}$, and*
4. *for each $!\mathsf{B}(b, p) \in \mathit{sent}(E)$ there exists an extension $E_b \in \mathcal{E}_i(\sigma_b)$ such that the set $\{!q \mid \langle !\mathsf{B}(b, q), a/?, \tau, \omega, \alpha \rangle \in \sigma^\star, \alpha \in \mathit{sent}(E)\})$ is a subset of $\{!q \mid \langle !q, b/?, \tau, \omega, \alpha \rangle \in \sigma_b^\star, \alpha \in \mathit{sent}(E_b)\})$.*

$\mathcal{E}(\sigma) = \mathcal{E}_l(\sigma)$ for the smallest $l \geq 1$ for which $\mathcal{E}_l(\sigma) = \mathcal{E}_{l+k}(\sigma)$ for all $k \geq 1$.

Let us comment on the third and fourth condition of the induction step, which insure that the extensions of a simulation belief state are properly constrained by the extensions of its parent belief state and vice versa. Condition three takes care of cases like this: If Cassie believes $!\mathsf{B}(\mathsf{Mary}, \mathsf{P})$ and $!\mathsf{B}(\mathsf{Mary}, \mathsf{Q})$, but these two sentences are in different extensions which means that she can never believe them simultaneously, then no simulation result in the Mary context which is based on both $!\mathsf{P}$ and $!\mathsf{Q}$ should ever be believable there.

Constraining into the opposite direction, condition four handles cases like the following: If $!\mathsf{P}$ and $!\mathsf{Q}$ are in different extensions in the Mary context, then $!\mathsf{B}(\mathsf{Mary}, \mathsf{P})$ and $!\mathsf{B}(\mathsf{Mary}, \mathsf{Q})$ should wind up in different extensions in the parent context of Mary.

Figure 2 contains a somewhat contrived example in order to demonstrate various believability situations at once. Because of space restrictions and for simplicity, belief sentences contain only proposition constants such as P or Q as object propositions, and the only inference rule applied in simulation contexts is or-introduction ($\vee$I), since it does not require any premises nor does it repeat any other sentences. Instead of these simplifications, more complicated sentences and inference patterns of the sort shown in Figure 1 could be used. The believabilities in the Cassie context are given according to the belief state σ defined in the example. The belief states that determine the believabilities of the other contexts are not displayed individually. Sentence $\mathsf{8}$ indicates how the problem of the introductory example can be solved. It is a simulation result that directly contradicts $\mathsf{2}$ which is a hypothesis. For that reason $\mathsf{8}$ is not even part of the prima facie extension $P(\sigma)$ and, since because of that it cannot be part of any extension at all, it is unbelievable. The remaining simulation assumptions $\mathsf{9}$, $\mathsf{15}$, and $\mathsf{18}$ are all part of $P(\sigma)$. However, since $\mathsf{9}$ and $\mathsf{15}$ lead to the contradiction in step $\mathsf{17}$, they cannot be in one extension together. For that reason, they and all sentences based on them are only *possibly* believable, since there is at least one extension in which they cannot be believed. The contradiction is of course unbelievable, because no extension contains $\mathsf{9}$ and $\mathsf{15}$. Sentence $\mathsf{18}$ is unproblematic and can be element of all extensions in $\mathcal{E}(\sigma)$. $P(\sigma)$ and $\mathcal{E}(\sigma)$ are both infinite sets which is indicated by the dots.

Intuitively, two sentences that are in different extensions cannot be believed by Cassie "in the same breath". An extension can be viewed as defining a frame of mind. Two sentences might be believable individually even if they are in different frames of mind, their conjunction, however, is only believable if they are in one frame together. Note, that while Cassie simulates Mary's reasoning she is in a different frame of mind, and thus sentence $\mathsf{6}$ is plausible in that context. Only once $\mathsf{6}$ gets exported to the parent context the resulting $\mathsf{8}$ becomes unbelievable. For simplicity, the example did not demonstrate any dependencies between simulation and simulator context. For example, if $\mathsf{6}$ had been unbelievable in the Mary context, then even without the presence of $\mathsf{2}$ sentence $\mathsf{8}$ would have become unbelievable in the Cassie context. This is desirable, since in our view of simulative reasoning Cassie attributes her reasoning skills identically to other agents.

When a sentence such as $\mathsf{8}$ becomes unbelievable it can still participate in derivations, because the believabilities are not taken into account by the deductive system $D_{\mathbf{SL}}$. However, the support computation ensures that every sentence based on it will also be unbelievable. This is a fact that can be exploited by the implementation which we will quickly sketch below.

Approximating Extensions

Our approach shares an ugly problem with default logics in general: The definition of extension is based on the notion of consistency which in a logic with quantification such as

[4] α selects the assumption origin set of a supported sentence.

	Cassie (⊤), I			
1	! B(M, B(S, P)),	hyp,	{1}, {}	H
2	! ¬B(M, B(S, P ∨ Z_S)),	hyp,	{2}, {}	H
				open Mary
8	⊛B(M, B(S, P ∨ Z_S)),	sim,	{1}, {8}	BI 6
9	≉B(M, B(S, P) ∨ Z_M),	sim,	{1}, {9}	BI 7
10	! B(M, B(S, P) ∨ Z_M) ⇒ ⇒ ¬B(L, Q ∨ Z_L),	hyp,	{10}, {}	H
11	! B(L, Q),	hyp,	{11}, {}	H
				open Lucy
15	≉B(L, Q ∨ Z_L),	sim,	{11}, {15}	BI 13
16	≉¬B(L, Q ∨ Z_L),	der,	{1, 10}, {9}	⇒E 9,10
17	⊛B(L, Q ∨ Z_L) ∧ ¬B(L, Q ∨ Z_L),	der,	{1, 10, 11}, {9, 15}	∧I 15,16
18	≁B(L, Q ∨ Z_{L2}),	sim,	{11}, {18}	BI 14

	Mary (Cassie), M, 3/{1}{}			
3	! B(S, P),	hyp,	{3}, {}	SH 1
				open MarySally
6	≁B(S, P ∨ Z_S),	sim,	{3}, {6}	BI 5
7	! B(S, P) ∨ Z_M,	der,	{3}, {}	∨I 3

	Lucy (Cassie), L, 12/{11}{}			
12	! Q,	hyp,	{12}, {}	SH 11
13	! Q ∨ Z_L,	der,	{12}, {}	∨I 12
14	! Q ∨ Z_{L2},	der,	{12}, {}	∨I 12

	MarySally (Mary), S, 4/{3}{}		
4	! P,	hyp,	{4}, {}
5	! P ∨ Z_S,	der,	{4}, {}

$$\begin{aligned}\sigma &= \langle\!\langle \mathsf{I}, \{1, 2, 10, 11\}, \{\}, \top \rangle\!\rangle \\ P(\sigma) &= \{\langle 9, ?, \mathsf{sim}, \{1\}, \{9\}\rangle, \langle 15, ?, \mathsf{sim}, \{11\}, \{15\}\rangle, \\ &\qquad \langle 18, ?, \mathsf{sim}, \{11\}, \{18\}\rangle, \ldots\} \\ \mathcal{E}(\sigma) &= \{\{\langle 9, ?, \mathsf{sim}, \{1\}, \{9\}\rangle, \langle 18, ?, \mathsf{sim}, \{11\}, \{18\}\rangle, \ldots\}, \\ &\qquad \{\langle 15, ?, \mathsf{sim}, \{11\}, \{15\}\rangle, \langle 18, ?, \mathsf{sim}, \{11\}, \{18\}\rangle, \ldots\}\}\end{aligned}$$

Figure 2: Simulation with believabilities

SL is an undecidable property. Since we want to use **SL** not just as a tool for theoretical analysis, but as the foundation for the implementation of an actual belief reasoning engine, this is a serious misfeature.

However, since we only want to model the reasoning of an agent (as opposed to do theorem proving), we can choose a weaker condition than consistency that is computable and still useful: Instead of checking whether the sentences of an extension are consistent with the hypotheses of a belief state which in general is impossible, we only require them to be *not known to be inconsistent*. This is similar to the approach taken by (Martins & Shapiro 1988).

Whenever in our implementation of SIMBA a sentence gets added to a reasoning context, and that sentence contradicts an already existing one, we recompute approximations of the extensions of all currently open reasoning contexts according to our iterative definition. Since we only have a finite number of sentences and only have to check for overt inconsistency, we do not have to compute closures or go to arbitrary levels of nesting. And since all sentences record in their support on which hypotheses and assumptions they are based, they will automatically change their believability according to the latest extension approximation. With this approach **SL** becomes a dynamic logic of sorts. The quality of the extension approximations can be improved by investing more work in detecting inconsistencies. One way to do is is to do some limited forward inference whenever a new sentence gets derived in order to detect contradictions that "lurk right around the corner". E.g., in the example above sentence 16 needed to be available to see that 9 and 15 were mutually inconsistent assumptions.

Conclusion

We presented **SL**, a nonmonotonic belief logic capable of formalizing an agent's reasoning about the beliefs of incomplete agents. **SL** combines a belief logic with a default reasoning mechanism to allow the shadowing of belief ascription results from simulative reasoning in case of evidence to the contrary. Using a notion of believability based on extensions, an agent built upon **SL** can keep multiple extensions "in mind" simultaneously, in case the simulation of two or more agents leads to mutually contradicting results. By relaxing the consistency condition in the definition of extensions, we get a notion of approximate extensions which is feasible to compute in the implementation of SIMBA.

SL does not itself provide a method to choose between multiple extensions, but it generates a set of candidates from which one could then choose a preferred extension according to some strategy. The derivation of simulation assumptions is always based on belief hypotheses, thus, an example strategy would be to order them according to some measure of epistemic entrenchment of these underlying hypotheses. However, the full logic **SL** does have representations and inference rules to make the believabilities of propositions explicit (cf. (Chalupsky 1995)), therefore, Cassie can base decisions on such believabilities even without a method of choosing between extensions.

It should be pointed out that the way **SL** uses default reasoning is different from what is done in Nested Theorist (van Arragon 1991), a system which concentrates on modeling users capable of default reasoning, rather than users whose reasoning is incomplete. Naturally, our choice of a deductive system as the underlying reasoning model limits us to model deductive reasoning only. In fact, in our treatment the only nondeductive aspect of Cassie's reasoning is simulative reasoning. However, this restriction is merely a matter of emphasis rather than a real limitation. Simulative reasoning is a paradigm that takes an arbitrary reasoning mechanism and attributes it to another agent in order to simulate its reasoning. Our choice was to use deductive reasoning as the basic mechanism, but in principle it could be anything. For example, it would be possible to combine **SL** with the default logic SWMC of Cravo and Martins (1993), thus providing Cassie with the additional ability to reason about the default reasoning of other agents akin to what is done by the Nested Theorist system, but in a more general framework.

References

Barnden, J. A., Helmreich, S., Iverson, E., and Stein, G. C. 1994. Combining simulative and metaphor-based reasoning about beliefs. In Ram, A., and Eiselt, K., eds., *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society,* 21–26. Hillsdale, NJ: Lawrence Erlbaum Associates.

Chalupsky, H., and Shapiro, S. C. 1994. SL: A subjective, intensional logic of belief. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society,* 165–170. Hillsdale, NJ: Lawrence Erlbaum.

Chalupsky, H. 1993. Using hypothetical reasoning as a method for belief ascription. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* 5(2&3):119–133.

Chalupsky, H. 1995. Belief ascription by way of simulative reasoning. forthcoming PhD dissertation.

Cravo, M. R., and Martins, J. P. 1993. SNePSwD: A newcomer to the SNePS family. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* 5(2&3):135–148.

Creary, L. G. 1979. Propositional attitudes: Fregean representations and simulative reasoning. In *Proceedings of the Sixth International Conference on Artificial Intelligence*, 176–181. Palo Alto, CA: Morgan Kaufmann.

Fagin, R., and Halpern, J. Y. 1988. Belief, awareness, and limited reasoning. *Artificial Intelligence* 34:39–76.

Konolige, K. 1986. Belief and incompleteness. In Hobbs, J. R., and Moore, R. C., eds., *Formal Theories of the Commonsense World*. Norwood, NJ: Ablex Publishing. chapter 10, 359–404.

Lakemeyer, G. 1990. A computationally attractive first-order logic of belief. In van Eijck, J., ed., *Logics in AI*. Berlin: Springer-Verlag. 333–347.

Levesque, H. J. 1984. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, 198–202. Palo Alto, CA: Morgan Kaufmann.

Martins, J. P., and Shapiro, S. C. 1988. A model for belief revision. *Artificial Intelligence* 35(1):25–79.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13:81–132.

van Arragon, P. 1991. Modeling default reasoning using defaults. *User Modeling and User-Adapted Interaction* 1:259–288.