Using Hypothetical Reasoning as a Method for Belief Ascription*

Hans Chalupsky
Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14216
(716) 645-2879
hans@cs.buffalo.edu

Running Head: Belief Ascription with Hypothetical Reasoning

*This is a preliminary version of: Hans Chalupsky, Using hypothetical reasoning as a method for belief ascription, *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, 5(2&3):119–133, April–September 1993. All quotes should be from, and all citations should be to the published version.

Abstract

A key cognitive faculty that enables humans to communicate with each other is their ability to incrementally construct and use models describing the mental states of others. Every such model describing some other cognitive agent will realistically contain only a finite number of sentences in some language of thought, hence, assuming sufficiently powerful inference rules, some of its consequences will remain implicit. To make them explicit, the person holding the model could employ a kind of reasoning that can be paraphrased as "what would I believe if I were the other person believing everything I believe that person believes", a strategy that can be viewed as a simulation of the other person's reasoning using the model of that person in conjunction with the reasoning abilities of the simulator.

If we want to equip an artificial cognitive agent with such a *simulative reasoning* ability we have to cope with problems such as simulation at various levels of nesting, meta-reasoning to make implicit agent model information explicitly available for its use in a simulation, and the defeasibility inherent in this reasoning strategy. This paper will describe how in a propositional semantic network formalism such as SNePS, in which propositions are terms of the representation language, we can employ hypothetical reasoning to achieve an elegant solution to the problems stated above. The relevance logic based belief revision mechanism employed by SNePS will automatically take care of some of the problems associated with the defeasibility of belief ascription by way of simulative reasoning. An example run will show how the presented solution can be used to perform simulative reasoning in the current implementation of SNePS.

1 Introduction

A key cognitive faculty that enables humans to communicate with each other in a fruitful and meaningful way is their ability to incrementally construct and use models describing the mental states of others, for example, what they believe, intend, plan, hope etc. A crucial component of such a model is the part which describes the beliefs of other people (or cognitive agents in general), and in this paper I will concentrate on that part only. A model describing some other cognitive agent will realistically contain only a finite number of sentences expressed in some language of thought, hence, assuming sufficiently powerful inference rules, some of the consequences of these beliefs will remain implicit. To make them explicit the agent holding the model has to use a reasoning strategy that can be paraphrased as "what would I believe if I were in the other agent's position", or more precisely, "what would I believe if I were the other agent believing everything I believe that agent believes". Such a strategy can be viewed as a simulation in which the simulating agent uses its model of the other agent in conjunction with its own reasoning abilities to simulate the other agent's reasoning.

If the task at hand is to build an artificial (or computational) cognitive agent capable of communicating with other agents, be they artificial or human, then we certainly have to equip it with some reasoning mechanism that can make the information that is only implicit in its models of other agents explicit. In this paper I will show how a certain kind of *simulative reasoning* mechanism [Moore, 1977; Creary, 1979] can be used to achieve that goal.

Any reasonable attempt to construct an artificial cognitive agent - in the following, I will refer to it as Cassie[1] - will contain at least the following two modules: A knowledge base and a reasoning mechanism. The knowledge base is a database that explicitly describes by way of some formalism things[2] that are assumed to be known or believed[3] by Cassie. Such a knowledge base could be, for example, a set of sentences of first-order logic or a semantic network of some sort. The reasoning mechanism is necessary to make information that is only implicit in the knowledge base explicit. For example, Cassie might know that dogs are animals, and then learn during a conversation that Fido is a dog. From that, she should be able to infer that Fido is an animal. Any state-of-the-art knowledge representation system can handle this standard sort of inference. Now consider the case that Cassie knows some other agent Lucy, and Lucy tells Cassie: "Fido is a dog". This utterance might prompt Cassie to add the following to her knowledge base: Lucy believes that (some) Fido is a dog. If we then ask Cassie whether Lucy believes that (some) Fido is an animal, she should in most circumstances answer yes, even though she did not have an explicit belief about that.

The two forms of inference described above look very similar, however, the reasoning in the second case has to be quite different from the first case. Assuming that Cassie employs simulative reasoning she had to go through the following steps: First, she had to hypothetically assume Lucy's belief that (some) Fido is a dog, even though she herself might not even know a dog named Fido or believe that Fido is actually a cat that just very much looks like a dog, then Cassie had to attribute the belief that dogs are animals to Lucy on the strength of the assumption that this is common knowledge known by just about any fool, and then she had to simulate Lucy's reasoning by attributing her own reasoning strategies to Lucy thus leading to the conclusion that from Lucy's beliefs it follows by simulation that (some) Fido is an animal. Finally, she could ascribe the conclusion of the simulation to Lucy as another belief explicitly stored in her model of Lucy.

A computational model for reasoning of the kind described above has to account for at least the following problems:

- Models of other agents have to be generated automatically and accurately. Just as in the simple common knowledge attribution above we might have to make parts of an agent model explicit before they can be used in the simulation of that agent. I will call this meta-reasoning, because it takes place outside the actual simulation context.

- Simulation has to work at arbitrary levels of nesting, for example, if Cassie wants to simulate Lucy's simulation of John, or, more realistically, Lucy's simulation of Cassie.

- Most importantly, the model has to account for the defeasibility that is inherent in this kind of reasoning, because in general we can never be sure whether the result of a simulation is actually believed by the simulated agent.

- Finally, the simulative reasoning mechanism should be closely integrated with the agent's own reasoning.

There is of course a vast amount of relevant literature concerned with the formalization of and reasoning about knowledge and belief. On the formal side of the spectrum are the various epistemic and doxastic logics, many of them modal in nature and inspired by Hintikka's famous treatment of knowledge and belief [Hintikka, 1962]. The modal approaches generally model agents as logically omniscient which is an undesirable idealization. Because of that, they do not deal with the defeasibility involved in the reasoning about the mental states of others. [Moore, 1985] gives a theory of knowledge rather than belief and thus does not have to deal with defeasibility at all, because agents can only know things that are true which can never conflict with anything else that is true (at least in a consistent world). [Konolige, 1986] develops a model for belief rather than knowledge, but nevertheless avoids the issue of defeasibility. In his model agents are only deductively instead of logically omniscient, which is a much weaker condition. The more recent treatment of [Zaverucha, 1992] addresses the issue of defeasibility in a very interesting approach that integrates modality, relevance and defeasibility in a single logic. Unfortunately, Zaverucha's axioms for absolute belief still imply that agents believe all theorems of his logic. This is not the case in a syntactic theory such as [Haas, 1986]. A drawback of syntactic theories is that they need to employ a complicated quotation machinery to deal with nested beliefs, and that they seem to be unintuitive. Haas' theory also does not deal with defeasibility. On the more psychologically oriented end of the spectrum are systems such as Viewgen [Ballim and Wilks, 1991], which its authors call a "highly pragmatic approach" [Ballim *et al.*, 1991]. Viewgen, too, does not deal with issues of defeasibility and belief revision, and it only does very limited reasoning with the beliefs it ascribes to other agents.

In the following, I will describe how in a propositional semantic network formalism such as SNePS we can achieve an elegant solution to the problems described above without resorting to a special belief logic, yes without changing its underlying logic at all. In SNePS, the Semantic Network Processing System [Shapiro and Rapaport, 1987; Shapiro and Rapaport, 1992], propositions (represented by proposition nodes) are terms of the language which allows us to express deduction rules which quantify over arbitrary propositions that might be believed by other agents; hence, we will be able to tell our computational agent Cassie how to perform simulative reasoning in much the same way as we tell it that dogs are animals.

SWM [Martins, 1983; Martins and Shapiro, 1988] is the formal logic underlying SNePS. It is based on relevance logic [Anderson and Belnap, 1975] and used to define the Multiple Belief Reasoner, or MBR, which is an abstract definition of a belief revision system. The way in which MBR handles contextual reasoning and belief revision will allow us to handle some of the problems associated with the defeasibility of simulative reasoning.

Finally, SNePS comes with an inference package called SNIP which implements a proof procedure for SWM. Using this package I can show an actual example run that demonstrates how simulative reasoning can be carried out using the approach presented below.

2 Agent Models

Simulative reasoning is a mechanism that uses and extends agent models, hence it is intimately tied to the representation of such models. So, what should an agent model look like? How can common knowledge, private knowledge, expertise shared by a group of experts etc. be represented to accurately model the beliefs of other agents? Below I will describe a SNePS representation for specific beliefs as expressed in "Lucy believes that Fido is a dog", and motivate why it is sufficient to only consider one simple belief representation for the development of the reasoning mechanism.

As mentioned before one of the major applications of SNePS is the construction of an artificial cognitive agent called Cassie. Cassie's cognitive makeup is a semantic network whose nodes are structured by syntactic formation rules (or case frames) with corresponding

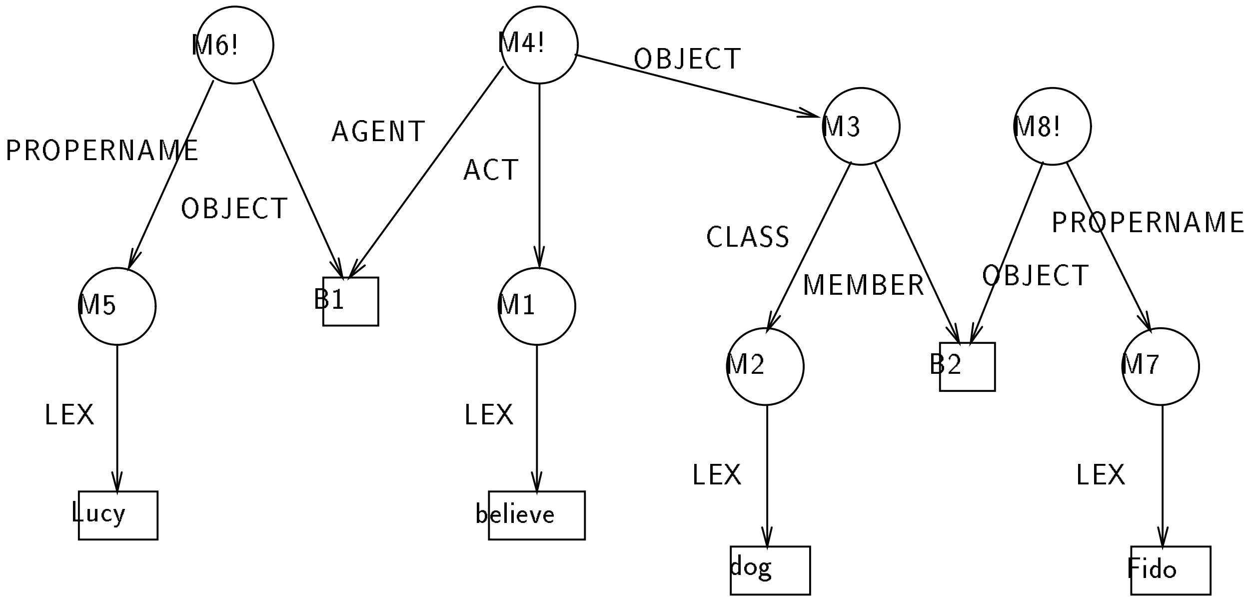

Figure 1: Cassie's *de re* representation of "Lucy believes that Fido is a dog."

semantic interpretations. The various types of nodes together with a set of well-formed networks and their interpretations are explained in detail in [Shapiro and Rapaport, 1987]. What concerns us here is the structure of the case frame that Cassie uses to represent other agents' beliefs, for example, the kind of network she builds in her mind when she processes the sentence "Lucy believes that Fido is a dog." Figure 1 shows how the *de re* interpretation of this sentence is represented in Cassie's mind.

The case frame shown in Figure 1 is due to Rapaport [Rapaport, 1986]. In this paper, I shall not be concerned with the various problems associated with *de re* and *de dicto* belief reports. For a detailed treatment of these issues see [Rapaport, 1986; Rapaport *et al.*, 1986; Wiebe and Rapaport, 1986]. Here it suffices to say that the canonical *de re* reading of the above sentence is "Lucy believes *of* Fido that he is a dog" (cf. [Rapaport, 1986, page 392]), which can be further expanded into "Some individual, which Cassie believes is named

Lucy, believes of some other individual, which Cassie believes is named Fido, that it belongs to the class of dogs", a reading that quite straightforwardly translates into the network shown in Figure 1. Nodes whose names start with M are molecular nodes, if they have an exclamation mark they are called *asserted nodes*, i.e., they represent propositions actually believed by Cassie. Nodes at the tails of LEX arcs are structured individuals. The LEX arcs point to base nodes that provide the natural language interface to the outside world used by the natural language parser and generator. Nodes whose names start with a B are base nodes which represent arbitrary individuals that are further qualified by the propositions asserted about them.

The central belief case frame is the AGENT/ACT/OBJECT case frame that expresses Cassie's belief that some individual B1 believes the proposition M3. The node M3 represents the proposition that some individual B2 is a member of the class dogs. M6 represents Cassie's belief that the individual B1 is named Lucy, and M8 represents Cassie's belief that the individual B2 is named Fido. Notice that in this figure Cassie does not have any beliefs about whether Fido is a dog, because M3 is not asserted.

Figure 2 shows how Cassie can represent a complete agent model of Lucy. Such a model is represented as a set of belief case frames of the kind shown in Figure 1. Notice that Cassie might herself believe some of the beliefs she attributes to Lucy. This is illustrated by the asserted proposition P_{L1}! and the unasserted proposition P_{Lk} (the triangles stand for some arbitrary SNePS networks representing these propositions).

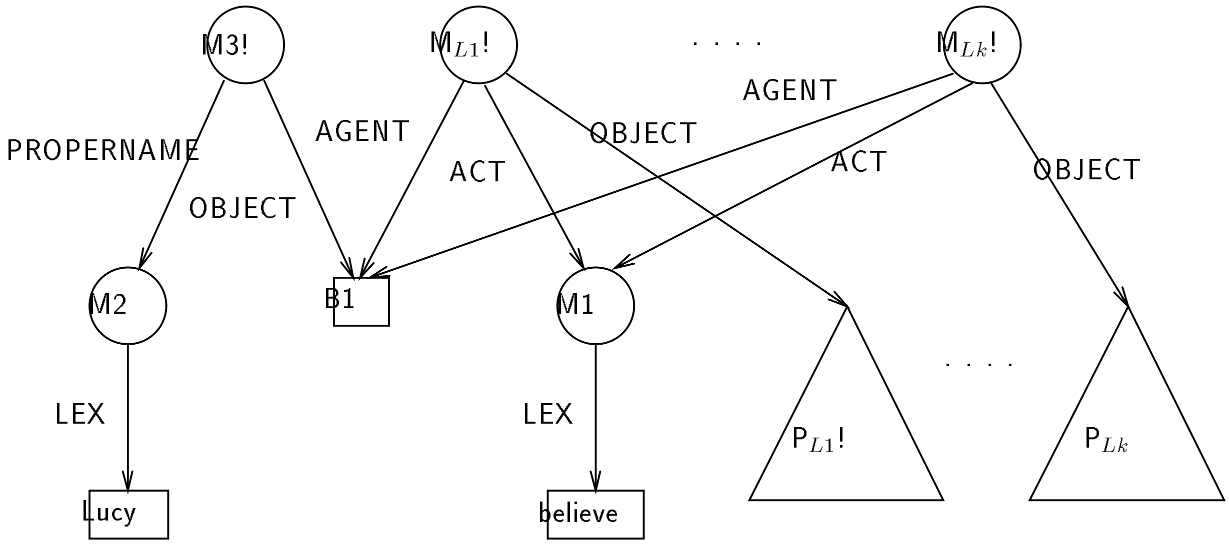

Figure 2: Cassie's representation of Lucy's beliefs

Notation

In the following, I will use the linear notation $B(a, p)$ as an abbreviation for the belief case frame described in Figure 1, where a is a meta-variable ranging over agents, and p is a meta-variable ranging over arbitrary propositions that might be believed by an agent. Propositions will be approximated by sentences written in standard predicate logic notation. I will use a Fitch-style natural deduction system [Fitch, 1952] to describe inference. The sentences at the outermost level of a proof should be read as the propositions believed by Cassie (corresponding to asserted proposition nodes). Hence, the set of top-level sentences $B(Lucy, P_{L1}), \ldots, B(Lucy, P_{Lk})$ will be used to linearly represent Cassie's model of Lucy's beliefs as shown in Figure 2. A proposition P that is believed by Cassie and Lucy (according to Cassie's view of Lucy) will be linearized as $P, B(Lucy, P)$, however, in the actual network the structure representing the commonly believed proposition will be shared.

3 Simulative Reasoning is a Form of Hypothetical Reasoning

Let us reconsider how Cassie can simulate Lucy's reasoning as required by the initial example above: Suppose Cassie's model of Lucy's beliefs looks like this: $B(Lucy, P_1)$, $B(Lucy, P_2)$, $B(Lucy, P_3)$. If Cassie wants to know whether $B(Lucy, Q)$ she has to hypothetically assume Lucy's beliefs as her own and try to infer Q. This amounts to proving the entailment $(P_1 \wedge P_2 \wedge P_3) \supset Q$. If successful, she can conclude $B(Lucy, Q)$ based on the result of the simulation.

As mentioned above, Cassie's reasoning engine is based on SWM which does not have any specific inference rules for beliefs, hence Cassie does not know how to deal with propositions inside belief case frames. However, we can tell her how to handle belief case frames by giving her special deduction rules that describe how to perform simulative reasoning. For example, a rule that would tell her how to deal with the specific simulation above could be expressed like this: If Lucy believes P_1 and P_2 and P_3, and if from $P_1 \wedge P_2 \wedge P_3$ you can infer Q than conclude that Lucy believes Q, or more formally,

$$(B(Lucy, P_1) \wedge B(Lucy, P_2) \wedge B(Lucy, P_3) \wedge ((P_1 \wedge P_2 \wedge P_3) \supset Q)) \supset B(Lucy, Q)$$

Of course, telling Cassie about such a specific case would not be very useful at all, hence we need to give her a more general description of how to simulate somebody else's reasoning. The basic problem is how to formalize the hypothetical reasoning part without explicitly referring to the propositions believed by the simulated agent. Here is a way how we can do that:

$$\forall a, p\left(\left(\left(\forall p_2(B(a, p_2) \supset p_2)\right) \supset p\right) \supset B(a, p)\right) \qquad \text{(B)}$$

I will call this rule B because it deals with reasoning about other agents' beliefs. Paraphrased in English it means the following: If, from assuming that for all propositions p_2 believed by agent a it follows that p_2, we can infer some proposition p, then we can conclude that a believes p. The inner universally quantified expression can be viewed as a generator that allows Cassie to generate all object beliefs of a as beliefs hypothetically also believed by Cassie in a subproof in which she tries to infer some p.

As should be evident from the way rule B quantifies over arbitrary propositions that this is only a rough predicate logic approximation of an actual SNePS network used to represent it. In SNePS special *rule nodes* are used to represent rule-like information, for example, if we want to tell Cassie that all dogs are animals we add a rule node expressing this information to the network that defines her "mind" (we can can tell her either in natural language or with a SNePS command dependent on the kind of interface we use). Just as such a rule quantifies over individuals that are members of the class dog, rule B quantifies over propositions that are objects of belief case frames. Cassie can then use these rules to infer certain things that are not explicitly represented in her mind, for example, that certain individuals which are members of the class dog are also animals, or, with rule B, new beliefs that follow by way of simulation from beliefs residing in her agent models.

4 Why is $\mathbf{B(a, p)}$ Sufficient?

It should be clear now why in the formalization of rule B it is sufficient to only consider the $B(a, p)$ case frame. Suppose Cassie employs a special case frame to represent mathematical expertise, let us call it $MathB(p)$, to mean that all mathematicians believe (or know) some proposition p. To allow Cassie to use such propositions in a simulation we can again use rules to tell her how, for example, the rule

$$\forall a, p((MathB(p) \wedge Mathematician(a)) \supset B(a, p))$$

would enable her to use all such propositions in the simulation of the reasoning of a mathematician. Another way of expressing this without quantifying over arbitrary propositions would be

$$\forall a(Mathematician(a) \supset B(a, P))$$

where P is some particular proposition commonly believed by mathematicians. Various forms of group and common knowledge can be handled that way, too.

5 SWM and Relevant Implication Introduction

For the example in the next section it is important to understand how implication introduction in a relevance logic differs from implication introduction employed by some natural deduction system for standard first-order predicate logic.

The formal logic underlying SNePS is SWM, which is a relevance logic whose basic building blocks are *supported well-formed formulas* (swffs). An swff is a well-formed formula (wff) with an associated triple called its *support*, which contains an *origin set*, an *origin tag*, and a *restriction set*. The origin set of an swff specifies the set of hypotheses on which the derivation of its wff was based, hence, in the case of a hypothesis, the origin set will be a singleton set containing only the name of the wff itself (wffs will be named by numbers;

hence, origin sets will be sets of numbers). An origin tag can be either *hyp* for hypotheses, or *der* for derived wffs. For a description of restriction sets as well as a third origin tag called *ext* the reader is referred to [Martins, 1983; Martins and Shapiro, 1988].

The main goal of relevance logics is the formalization of a form of implication in which the antecedent suffices *relevantly* for the consequent. Relevance logics generally deny the validity of the so-called paradoxes of material implication: $A \supset (B \supset A)$ and $(A \wedge \neg A) \supset B$, which are valid theorems of classical logic. One method to achieve that, is to restrict the applicability of the inference rule of implication introduction ($\supset$I) in a way presented below. SWM is an Anderson-Belnap-style relevance logic [Anderson and Belnap, 1975] which employs a Fitch-style natural deduction system as its proof method. Natural deduction systems generally have a rule of hypothesis to get started, together with introduction and elimination rules for the various logical connectives. Here are the definitions of the inference rules relevant to the form of hypothetical reasoning employed in applications of rule B:

Rule of Hypothesis (Hyp): At any point in a proof a new hypothesis might be introduced. Such an introduction opens a new subproof. As its origin set, the new hypothesis gets assigned a singleton set $\{k\}$, where k has to be a natural number not previously used to identify any other hypothesis in the proof.

Repetition (R): At any point of the subproof, the hypothesis that initiated it can be repeated with origin set $\{k\}$ and origin tag *hyp* to be used in an inference step. A similar inference rule, *reiteration*, allows results previously proved in some outer subproof to be imported.

Implication Introduction ($\supset$I): If at any point in the subproof we come to a conclusion whose origin set contains k, we are allowed to discharge the hypothesis and introduce the entailment in the outer subproof. The restriction that the origin set has to contain k insures that the hypothesis was actually used in the derivation of the conclusion and hence relevantly implies it. The origin set of the conclusion minus the singleton origin set of the hypothesis defines the origin set of the resulting entailment.

Figure 3 gives a schematic representation of a proof in which all three of the above inference rules were applied. A line in such a proof typically consists of a line number that also serves as a name for the wff, the actual wff, its support consisting of an origin tag and an origin set, and finally, a description of the inference rule that led to the particular wff together with the names of the wffs that justified its application.

	$\ldots$		
m	$\quad p$	$\langle \text{hyp}, \{k\} \rangle$	Hyp, unique k
	$\quad \vdots$		
	$\quad p$	$\langle \text{hyp}, \{k\} \rangle$	R(m)
	$\quad \vdots$		
n	$\quad q$	$\langle ot(q), \alpha \cup \{k\} \rangle$	
	$p \supset q$	$\langle \text{der}, \alpha \setminus \{k\} \rangle$	$\supset$I(m,n)

Figure 3: Rules for relevant implication introduction

The inference rules of SWM guarantee that the origin set of any derived wff is the set of hypotheses which were relevantly used in its derivation. This property in conjunction with what [Martins and Shapiro, 1988] call a *contextual interpretation* for SWM make it ideally suited to be applied by a belief revision system.

Whether a certain wff is said to be believed by the system is defined relative to a *context.* A context is a set of hypotheses, and a wff is said to be believed in a context if its origin set

1	$\forall x(Dog(x) \supset Animal(x))$	$\langle$hyp, $\{1\}\rangle$	
2	$Dog(Rover)$	$\langle$hyp, $\{2\}\rangle$	
3	$B(Lucy, Dog(Fido))$	$\langle$hyp, $\{3\}\rangle$	
4	$\forall a, p(((\forall p_2(B(a,p_2) \supset p_2)) \supset p) \supset B(a,p))$	$\langle$hyp, $\{4\}\rangle$	B
	$B(Lucy, Animal(Fido))$?		Query
5	$((\forall p_2(B(Lucy,p_2) \supset p_2)) \supset Animal(Fido)) \supset B(Lucy, Animal(Fido))$	$\langle$der, $\{4\}\rangle$	UI(4), $a/Lucy$, $p/Animal(Fido)$
6	$\forall p_2(B(Lucy,p_2) \supset p_2)$	$\langle$hyp, $\{6\}\rangle$	
7	$B(Lucy, Dog(Fido)) \supset Dog(Fido)$	$\langle$der, $\{6\}\rangle$	UI(6), $p_2/Dog(Fido)$
	$B(Lucy, Dog(Fido))$	$\langle$hyp, $\{3\}\rangle$	Reiteration(3)
8	$Dog(Fido)$	$\langle$der, $\{3,6\}\rangle$	$\supset$E(7,3)
9	$Dog(Fido) \supset Animal(Fido)$	$\langle$der, $\{1\}\rangle$	UI(1), $x/Fido$
10	$Animal(Fido)$	$\langle$der, $\{1,3,6\}\rangle$	$\supset$E(8,9)
11	$Animal(Rover)$	$\langle$der, $\{1,2\}\rangle$	UI(1), $\supset$E
12	$(\forall p_2(B(Lucy,p_2) \supset p_2)) \supset Animal(Fido)$	$\langle$der, $\{1,3\}\rangle$	$\supset$I(6,10)
13	$B(Lucy, Animal(Fido))$	$\langle$der, $\{1,3,4\}\rangle$	$\supset$E(5,12)

Figure 4: A simulative reasoning example

is a subset of that context. Removing a hypothesis from a context, because, for example, a contradiction was derived, will automatically disbelieve all wffs that have that hypothesis in their origin set, because the status of belief is determined dynamically with respect to the context. This mechanism is generally called assumption based truth maintenance [Martins, 1983; de Kleer, 1986; Martins and Shapiro, 1988].

At any point in time a *current context* is defined as the set of hypotheses currently believed by the system (or Cassie). In our proof notation the current context at any point in the proof is defined as the set of hypotheses in the outermost proof plus all the hypotheses that initiated currently active subproofs.

6 An Example

Figure 4 shows a simple example that demonstrates how rule B in conjunction with relevant implication introduction can be used to perform simulative reasoning. The line numbers on the left serve as names for the wffs. All hypotheses have origin tag *hyp* and as origin set a singleton set with their name. The top-level sentences represent Cassie's view of the world, i.e., that all dogs are animals (1), Rover is a dog (2), Lucy believes that Fido is a dog (3), and rule B (4), all of which are hypotheses. Note again that these sentences are only predicate logic approximations of actual SNePS networks used to represent propositions of that kind, for example, sentence (3) is actually represented by a node of the same structure as node `M4` in figure 1. The reason for the use of this notation is to highlight the reasoning aspects of the example which would be more difficult to understand using the actual networks.

What we want to show is whether Cassie believes that Lucy believes that Fido is an animal. To do that we first apply universal instantiation (UI) to B in order to get sentence (5) which has the query as a consequent. Note that this is not an instantiation of some kind of sentence schema (p in B gets substituted by the sentence *Animal(Fido)*), but rather an ordinary instantiation obtained by substituting a variable by a proposition which is an ordinary term in SNePS that can be used in a substitution (see above).

Now we are ready to start the simulation. To do that we open a subproof in which we hypothetically assume that all propositions p_2 believed by Lucy are true or also believed by Cassie (6). This will allow us to generate all of Lucy's beliefs as beliefs hypothetically believed by Cassie within the subproof, to which we then can apply all the standard inference rules in order to simulate Lucy's reasoning. (7) is an instantiation of (6) which after reiterating (3) and applying modus ponens (or $\supset$E) allows us to derive that Fido is a dog (8). At that point Cassie hypothetically believes (8). Using an instance of (1) and applying $\supset$E we can derive that Fido is an animal (10). Similarly, using Cassie's belief that Rover is a dog (2) we can derive that Rover is an animal (11).

Because sentence (10) contains the name of the hypothesis on top of the subproof in its origin set, we are allowed to use implication introduction to arrive at (12). Notice that we could not do the same with sentence (11), because the hypothesis (6) was not used in its derivation. Classical implication introduction would have allowed this inference which could have led us to conclude that Lucy believes that Rover is an animal which does not follow from the assumptions in this example. Finally, applying modus ponens again to (12) and the instance of B (5) we arrive at the desired conclusion.

7 Discussion of Rule B

Let me briefly discuss how rule B provides solutions to the various problems mentioned in the introduction. While the example above only demonstrated a simulation at one level of nesting there is no restriction to that extent. With a very similar strategy we could have shown that (Cassie believes) Lucy believes that Sally believes that John believes that Fido is an animal. Instead of only one subproof we would have had three nested subproofs with propositions getting unwrapped from their belief case frames step by step while moving towards deeper levels of nesting, and finally simulation results getting wrapped again with proper belief case frames while moving out to the top-level proof.

Meta-reasoning is provided automatically by the way nested beliefs are represented in SNePS, and by the way inference is handled in SWM. If the belief that Lucy believes Fido is a dog had not been explicitly available, but had rather followed from Sally believes that Fido is a dog and Lucy believes everything that Sally believes, we would have been able to derive it (in a meta-reasoning step) before its use in the simulation. In a database approach to belief representation, where a certain belief of some agent is represented by membership of a sentence in some agent database, meta-reasoning would have been complicated if not impossible. On a similar note, the presented mechanism is general enough to deal with disjunctive belief or negative belief, for example, if Cassie knew that Lucy believes Fido is a dog or Lucy believes that Fido is a cat. However, the current implementation of SNIP cannot handle reasoning by cases.

Cassie in conjunction with rule B does certainly not model other agents as logically omniscient. Other agents are only assumed to believe what follows from Cassie's models of them by exhaustive application of Cassie's inference rules. In this respect, the presented approach is similar to [Konolige, 1986], which uses a deductive closure assumption.

Finally, some aspects of the defeasibility of simulative reasoning are automatically taken care of by SWM. The origin set of a simulation result contains the exact set of assumptions used in its derivation. Should we later have to retract any of those assumptions, for example, if we (or Cassie) find out that Lucy did not believe that Fido is a dog, then the removal of this assumption from the set of beliefs held by Cassie will automatically disbelieve the result of the simulation, because then its origin set is not anymore a subset of Cassie's beliefs.

The only shortcoming of this approach, though a major one, is that private beliefs are not shielded properly. While the use of relevant implication introduction makes sure that the hypothesis on top of the simulation proof and by that at least one of Lucy's beliefs was used to derive the conclusion, it does not restrict the use of additional information. In fact, in the example above we actually used this property when we imported the common belief that all dogs are animals. Had this belief been some expertise only privately known by Cassie, we would have arrived at an unwarranted simulation result. This problem cannot be remedied without extending SWM with inference rules that specifically deal with simulation subproofs.

8 A Demonstration Run

The demonstration run below shows how a slightly modified version of the example in Figure 4 can actually be run in the current implementation of SNePS. The demonstration uses the SNePSLOG front-end [Shapiro *et al.*, 1981; Matos and Martins, 1989] to communicate with SNePS in a predicate-logic-style language, similar to the one used all along. SNePSLOG parses input sentences and translates them into SNePS networks. It uses a special case frame for the representation of predicates which is different from the ones shown previously, however, in this particular example which concentrates on inference this will not make any difference.

The output below was only slightly edited for formatting purposes and split into a few sections to allow room for explanations. Input commands issued to SNePSLOG are entered at the : prompt. Command output (if any) follows the input separated by a blank line. An occasional CPU time message shows how many seconds it took to execute the previous command.

We start by entering the SNePSLOG system and clearing the network. Next we define a new and empty Cassie context which is then declared to be the default context. All sentences (or wffs) entered after that will be added to that context. Simply typing a sentence is interpreted as an assertion of that sentence, i.e., the node representing the sentence will be added as a hypothesis to the current context. Similar to the example in Figure 4 we first tell Cassie that all dogs are animals, that Lucy believes that Fido is a dog, and then rule B. The version of rule B used here is more restricted, because it only deals with unary predicates. This suffices for the sake of this example and reduces the amount of unnecessary inferences made. Finally, we list all the hypotheses entered so far. The exclamation marks at the start of the line tell us that these are all wffs believed in the current Cassie context. Every wff is followed by a (singleton) set of supports, where each support is a triple consisting of an origin tag, an origin set, and a restriction set which in this example will always be empty. Wff numbers correspond to SNePS node labels. Node `M2` is a subpart of some other node, hence there is no `WFF2`.

```
> (snepslog)
Welcome to SNePSLOG (A logic interface to SNePS-2.1)

:  clearkb
Knowledge Base Cleared

:  set-context Cassie ()

:  set-default-context Cassie

:  all(x)(Dog(x) => Animal(x))

   all(X)(DOG(X) => ANIMAL(X))

CPU time :  0.07 GC time :  0.00

:  B(Lucy, Dog(Fido))

   B(LUCY,DOG(FIDO))
```

```
CPU time :  0.06 GC time :  0.00

:  all(a,p,x)( (all(p2,y)(B(a,p2(y)) => p2(y)) => p(x)) => B(a,p(x)))

   all(A,P,X)(((all(P2,Y)(B(A,P2(Y)) => P2(Y))) => P(X)) => B(A,P(X)))

CPU time :  0.17 GC time :  0.00

:  list-wffs

!  WFF1:  all(X)(DOG(X) => ANIMAL(X)) {<HYP,{WFF1},{}>}
!  WFF3:  B(LUCY,DOG(FIDO)) {<HYP,{WFF3},{}>}
!  WFF4:  all(A,P,X)(((all(P2,Y)(B(A,P2(Y)) => P2(Y))) => P(X))
                          => B(A,P(X))) {<HYP,{WFF4},{}>}
```

Now we are ready to start inference. A sentence followed by a question mark is interpreted as a query. When SNePSLOG sees a query it starts a deduction to find out whether the sentence in question is already believed or follows from what is in the network. To deduce a certain sentence (or node) SNIP (the SNePS Inference Package) tries to backward-chain through rule nodes with matching consequents. An inference tracing facility allows us to see how subgoals get pursued. To save space, irrelevant inference traces have been edited out. In the example below the only rule whose consequent matches the query is rule B. After performing the proper variable substitutions SNIP "wonders" whether the antecedent of rule B holds in the belief space (BS) defined by context Cassie. This antecedent is in itself an entailment. To find out whether it holds, SNIP forms a new context that contains the antecedent of the entailment as an additional assumption and then tries to prove the consequent in the new context. This context contains all hypotheses of the Cassie context plus the new assumption `WFF7`. It does not have a name and hence is referred to by explicit mention of the hypotheses defining it. Reasoning in this context corresponds to the subproof in the example above.

```
:  B(Lucy,Animal(Fido))?

I wonder if B(LUCY,ANIMAL(FIDO))
holds within the BS defined by context CASSIE

I wonder if (all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))) => ANIMAL(FIDO)
holds within the BS defined by context CASSIE

Let me assume that all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))

I wonder if ANIMAL(FIDO)
holds within the BS defined by hypotheses (WFF1 WFF3 WFF4 WFF7)

..........

I wonder if DOG(FIDO)
holds within the BS defined by hypotheses (WFF1 WFF3 WFF4 WFF7)

..........

I wonder if B(LUCY,DOG(FIDO))
holds within the BS defined by hypotheses (WFF1 WFF3 WFF4 WFF7)
```

Now we have finally reached the end of a deduction chain. Because Lucy believes that Fido is a dog, and because of our assumption it follows (hypothetically) that Fido is a dog, and hence Fido is an animal, and after implication introduction which proves the antecedent of rule B we can conclude by way of simulation that Lucy believes that Fido is an animal. SNIP then, unsuccessfully, tries a few more options and finally reports the result.

```
I know B(LUCY,DOG(FIDO))

Since all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))
and B(LUCY,DOG(FIDO))
I infer DOG(FIDO)

Since all(X)(DOG(X) => ANIMAL(X))
and DOG(FIDO)
I infer ANIMAL(FIDO)

Since ANIMAL(FIDO)
was derived under the assumption:  all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))
I infer (all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))) => ANIMAL(FIDO)

Since all(A,P,X)(((all(P2,Y)(B(A,P2(Y)) => P2(Y))) => P(X))
```

```
                         => B(A,P(X)))
and (all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))) => ANIMAL(FIDO)
I infer B(LUCY,ANIMAL(FIDO))

..........

B(LUCY,ANIMAL(FIDO))

CPU time :  5.89 GC time :  0.00
```

Let us now look at all the wffs in the network: WFF6 is the newly derived sentence that we asked for with all the initial hypotheses as its origin set. WFF8 is the introduced entailment which corresponds to sentence (12) in Figure 4. All the other wffs do not have exclamation marks, because they were derived in the subproof and have WFF7 in their origin set which is not a member of the current Cassie context; hence, they are not believed by Cassie. Some of the wffs have multiple supports now, because they were derived in more than one way.

```
:  list-wffs

!  WFF1:  all(X)(DOG(X) => ANIMAL(X)) {<HYP,{WFF1},{}>}
!  WFF3:  B(LUCY,DOG(FIDO)) {<HYP,{WFF3},{}>}
!  WFF4:  all(A,P,X)(((all(P2,Y)(B(A,P2(Y)) => P2(Y))) => P(X))
                             => B(A,P(X))) {<HYP,{WFF4},{}>}
!  WFF6:  B(LUCY,ANIMAL(FIDO)) {<DER,{WFF4,WFF3,WFF1},{}>}
!  WFF8:  (all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))) => ANIMAL(FIDO)
           {<DER,{WFF4,WFF3,WFF1},{}>,<DER,{WFF3,WFF1},{}>}
   WFF2:  DOG(FIDO) {<DER,{WFF7,WFF3},{}>}
   WFF5:  ANIMAL(FIDO) {<DER,{WFF7,WFF4,WFF3,WFF1},{}>, <DER,{WFF7,WFF3,WFF1},{}>}
   WFF7:  all(P2,Y)(B(LUCY,P2(Y)) => P2(Y)) {<HYP,{WFF7},{}>, <DER,{WFF7},{}>}
```

Should we (or Cassie) later find out that the assumption that Lucy believes that Fido is a dog was wrong, then we can simply remove it from the Cassie context and all the results dependent on it (such as WFF6 and WFF8) will get automatically disbelieved too.

```
:  remove-from-context Cassie (wff3)
```

```
:  list-wffs

!  WFF1:  all(X)(DOG(X) => ANIMAL(X)) {<HYP,{WFF1},{}>}
!  WFF4:  all(A,P,X)(((all(P2,Y)(B(A,P2(Y)) => P2(Y))) => P(X))
                             => B(A,P(X))) {<HYP,{WFF4},{}>}
   WFF2:  DOG(FIDO) {<DER,{WFF7,WFF3},{}>}
   WFF3:  B(LUCY,DOG(FIDO)) {<HYP,{WFF3},{}>}
   WFF5:  ANIMAL(FIDO) {<DER,{WFF7,WFF4,WFF3,WFF1},{}>, <DER,{WFF7,WFF3,WFF1},{}>}
   WFF6:  B(LUCY,ANIMAL(FIDO)) {<DER,{WFF4,WFF3,WFF1},{}>}
   WFF7:  all(P2,Y)(B(LUCY,P2(Y)) => P2(Y)) {<HYP,{WFF7},{}>, <DER,{WFF7},{}>}
   WFF8:  (all(P2,Y)(B(LUCY,P2(Y)) => P2(Y))) => ANIMAL(FIDO)
           {<DER,{WFF4,WFF3,WFF1},{}>,<DER,{WFF3,WFF1},{}>}
```

9 Conclusion

I demonstrated how an artificial cognitive agent can employ hypothetical reasoning for the defeasible ascription of beliefs to other agents. In a propositional semantic network formalism such as SNePS, in which propositions are ordinary terms of the representation language, we can provide the artificial cognitive agent with a special deduction rule that enables it to hypothetically reason with beliefs held by other agents. Using the beliefs it has explicitly represented in an agent model, and attributing its own reasoning abilities to that agent it can simulate the agent's reasoning and hence extend its model of that agent with explicit information that was only implicit before. The belief revision system of SNePS and its underlying logic SWM automatically takes care of some aspects of the defeasibility inherent in this kind of reasoning. Because the presented reasoning strategy only employs standard inference mechanism available in SNePS, I could provide an actual demonstration run that shows how simulative reasoning using the presented strategy can be actually carried out in the current implementation of SNePS.

Acknowledgment

This work was sponsored in part by DARPA and Rome Laboratory under USAF contract F30602-91-C-0040 via a subcontract from Paramax Corp.

Notes

[1] One of the major applications of SNePS is the construction of an artificial cognitive agent called Cassie which is an acronym for **C**ognitive **A**gent of the **S**NePS **S**ystem - an **I**ntelligent **E**ntity.

[2] Such "things" can be anything that may be imagined by a mind, not just things that exist in the real world.

[3] In this paper I will use the terms *knowledge* and *belief* rather interchangeably, even though technically, I will always mean belief and not be concerned with truth at all. This means a false belief is as good as any other.

References

[Anderson and Belnap, 1975] A. R. Anderson and N. D. Belnap. *Entailment: The Logic of Relevance and Necessity*. Princeton University Press, Princeton, NJ, 1975.

[Ballim and Wilks, 1991] Afzal Ballim and Yorick Wilks. *Artificial Believers: The Ascription of Belief*. Erlbaum, Hillsdale, NJ, 1991.

[Ballim *et al.*, 1991] Afzal Ballim, Yorick Wilks, and John A. Barnden. Belief ascription, metaphor, and intensional identification. *Cognitive Science*, 15(1):133–171, 1991.

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

[de Kleer, 1986] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986.

[Fitch, 1952] F. B. Fitch. *Symbolic Logic: An Introduction.* Ronald Press, New York, 1952.

[Haas, 1986] Andrew R. Haas. A syntactic theory of belief and action. *Artificial Intelligence*, 28:245–292, 1986.

[Hintikka, 1962] Jaakko Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions.* Cornell University Press, Ithaca, NY, 1962.

[Konolige, 1986] Kurt Konolige. *A Deduction Model of Belief.* Morgan Kaufmann, Palo Alto, CA, 1986.

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

[Martins, 1983] João P. Martins. *Reasoning in Multiple Belief Spaces.* PhD thesis, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY, 1983.

[Matos and Martins, 1989] Pedro A. Matos and João P. Martins. SNePSLOG - a logic interface to SNePS. Technical Report 89/03, Instituto Superior Técnico, Technical University of Lisbon, Lisbon, Portugal, 1989.

[Moore, 1977] Robert C. Moore. Reasoning about knowledge and action. In *Proceedings of the Fifth International Conference on Artificial Intelligence*, pages 223–227, Palo Alto, CA, 1977. Morgan Kaufmann. Reprinted in [Webber and Nilsson, 1981].

[Moore, 1985] Robert C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal Theories of the Commonsense World*, chapter 9, pages 319–358. Ablex Publishing Corp., Norwood, NJ, 1985.

[Rapaport *et al.*, 1986] W. J. Rapaport, S. C. Shapiro, and J. M. Wiebe. Quasi-indicators, knowledge reports, and discourse. Technical Report 86–15, Department of Computer Science, SUNY at Buffalo, 1986.

[Rapaport, 1986] William J. Rapaport. Logical foundations for belief representation. *Cognitive Science*, 10:371–422, 1986.

[Shapiro and Rapaport, 1987] S. C. Shapiro and W. J. Rapaport. SNePS considered as a fully intensional propositional semantic network. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier*, pages 263–315. Springer–Verlag, New York, 1987.

[Shapiro and Rapaport, 1992] Stuart C. Shapiro and William J. Rapaport. The SNePS family. *Computers & Mathematics with Applications*, 23(2–5):243–275, January–March 1992.

[Shapiro *et al.*, 1981] S. C. Shapiro, D. P. McKay, J. Martins, and E. Morgado. SNePSLOG: A "higher order" logic programming language. SNeRG Technical Note 8, Department of Computer Science, SUNY at Buffalo, 1981. Presented at the Workshop on Logic Programming for Intelligent Systems, R.M.S. Queen Mary, Long Beach, CA.

[Webber and Nilsson, 1981] Bonnie L. Webber and Nils J. Nilsson, editors. *Readings in Artificial Intelligence*. Tioga, Palo Alto, CA, 1981.

[Wiebe and Rapaport, 1986] J. M. Wiebe and W. J. Rapaport. Representing *de re* and *de dicto* belief reports in discourse and narrative. *Proceedings of the IEEE*, 74(10):1405–1413, 1986.

[Zaverucha, 1992] Gerson Zaverucha. Logical foundations of a modal defeasible relevant logic of belief. In Bernd Neumann, editor, *Proceedings of the Tenth European Conference on Artificial Intelligence*, pages 615–619, New York, 1992. John Wiley & Sons.