A clausal form of logic of belief

— A logic programming model

Y. J. Jiang

Computer Science Department,
University of Essex,
Wivenhoe Park,
Colchester, C04 3SQ

Abstract

In this paper, we present a clausal logic of belief which formalizes beliefs in an *extended clausal form* of logic. Our aims are to solve the representational problem of quantified beliefs and to allow an efficient resolution-like proof procedure with controlled granularity to be developed. A *levelled* intensional scheme that enables the clausalization of beliefs is proposed. An inferential power bounded resolution rule of beliefs for the formalism is introduced. The formal semantics of the formalism is defined. A general circumscriptive non-monotonic reasoning system for belief revision is described. Finally, a scheme for handling the consistency of beliefs under Tarski's *truth definition theorem* is developed.

Keywords: logic programming, computational models of belief, intension, imputation, non-monotonic reasoning and semantic paradoxes.

1 Introduction

There are currently three approaches to formalization of beliefs. In the semantic approach (eg. [Moore 85], [Halpern&Moses 85]), beliefs are characterized by accessibility relations between possible worlds. In the partition approach (eg. [Kobsa 85]), beliefs are identified with the presence of representation structures in specific nested belief spaces reserved for the respective agent. In the syntactic approach (eg. [Konolige 85]), belief of an agent is equated with derivability of a first order theory of the agent. Despite the fundamental differences of the three approaches, one common characteristic is that they all assume the use of *arbitrary forms* of logic in their representation of beliefs.

Although the form of representation can play a part in the *meaning* of a sentence from a strict cognitive sense (for example, representing the belief 'there does not exist a person who does not like Mary' as 'Everybody likes Mary' could be misleading because an agent of the belief may not know that these two sentences are equivalent), however several problems can arise from the use of arbitrary forms of logic. One problem is the difficulty of developing an *efficient* proof procedure for reasoning about beliefs. The solution is usually based on a natural deduction approach which does not seem to have much success in AI applications.

Another problem is the difficulty of formalizing the intensionality of arbitrary quantifications. (In fact, many schemes, in particular the semantic ones such as [Levelsque 84] and [Halpern&Moses 85] are careful to avoid this problem by sticking with propositional beliefs.) For example, it is easy to see Konolige's [85] semantics of the statement $BEL(Jim, \exists x Unicorn(x))$ as: $\exists x Unicorn(x)$ is in the belief space of Jim; however it is difficult to see how the semantics of the *quantifying-in* belief $\exists x BEL(Jim, Unicorn(x))$ can be formalized in a similar way because the quantification is outside the scope of BEL predicate. Konolige's solution is to treat the second sentence in a similar way as $\exists x P(x)$ where P is any ordinary predicate. However this seems to obscure the semantics of the BEL predicate. For example, given the following universally qualified belief

$\forall x BEL(Jim, Unicorn(x) \rightarrow Likes(Jim, x))$.

and the following quantifying-in belief, $\exists x BEL(Jim, Unicorn(x))$, following Konolige's semantics, it is difficult to see how we can derive:

$$\exists x BEL(Jim, Likes(Jim, x))$$

which we should do. The problem of intensional quantification gets more serious when we try to represent nested beliefs, eg. $BEL(Jim, \exists x BEL(Tom, Unicorn(x))$ in which Jim believes that there is a particular individual in Tom's mind whom Tom believes to be a unicorn although Jim has no idea himself who this individual could be.

A third problem is the granularity of implied beliefs of an agent as discussed in [Levelsque 84]. In the possible world approach, beliefs of an agent are represented by a set of possible worlds that are compatible with what the agent believes. A recuring problem in this approach is *logical ominiscience*, ie. the set of beliefs is closed under logical implication. This means that anyone who can be persuaded of the truth of Peano's postulates knows everything about number theory that anyone else knows. In addition, every valid sentence must be believed by every agent and contradictory beliefs of an agent imply that the agent believes anything. The possible world approach is thus *too coarse-grained* in the sense that it cannot distinguish same logical sets of beliefs. The syntactic and partition approaches on the other hand are *too-fine grained* in the sense that it distinguishes too much on same logical sets of beliefs. For example, in these approaches, an agent may believe A and B but not $A \wedge B$. To avoid these spurious syntactic distinctions, the obvious axioms must be present which could complicate the proof theory of belief.

Thus we propose an alternative approach which formalizes beliefs in a quantifier-free *canonical form* of logic. As argued in [Moore & Hendrix 79], beliefs should be represented in an internal language from a computational perspective rather than an external language of thought although different external languages of the same content may not have the same meaning. Here we further argue that beliefs should be represented in a canonical form of an internal language from a *logic programming* perspective although different forms of the language of the same content may not have the same meaning. We have chosen an extended clausal form of logic as such a canonical form for three reasons. The first is that a clausal form of logic is quantifier-free. This would simplify the task of formalizing the semantics of quantified beliefs. The second is that clausal forms of logic form the basic foundation of current logic programming systems and Japanese Fifth Generation Computer Kernel Languages([Kowalski 83]). In other words, there are well-developed and efficient proof procedures for clausal forms of logic. The third reason is that it seems easier to control the granularity of implied beliefs in a clausal form of logic. However it should be understood that clausal forms of logic are *by no means* the only forms used in logic programming. In fact, the concept of logic programming can be applied to *any forms of logic*. The reason we call our approach a "logic programming model", is simply because the *current most promising* logic programming approach is to operate on a clausal form of logic.

It is worth noting that a recent attempt has been made by [Cerro 86] to allow modal reasoning in Prolog [Clocksin&Mellish 81]. However no concern is given to the problem of formulating clausal forms of logic of abitrarily quantified beliefs. In particular, the problem of intensionality of terms is not addressed. In addition, the semantics of the scheme is based on the possible world approach, hence also suffers the granularity problem of possible world approach. However, Cerro's approach does have many practical applications, especially those involving knowledge base systems for which the granularity problem of modal approach is less serious.

The paper is organized as follows. Section 2 introduces the basic preliminaries. Section 3 describes the problem and solution of clausalizing beliefs. Section 4 discusses the inference mechanism and the formal semantics of the proposed formalism. Section 5 presents an explicit circumscriptive non-monotonic reasoning scheme for handling belief revision. Section 6 proposes a scheme to allow a modified Tarski's *truth definition theorem* to be consistent with our formalism.

2 Clausal form of Logic, Skolemization and Resolution

Every standard first order formula can be transformed into a prenex-normal form which is logically equivalent. A Prenex-normal form can be written as QM, where Q consists of all the quantifiers in the formula ($\exists$ and $\forall$), while M is a quantifier-free well-formed formula (wff)[1].

By introducing Skolem functions, one can eliminate existential quantifiers, and hence the universal quantifiers since they are then implied, i.e. indicated by leaving variables free. Skolemization is achieved by replacing each of the existentially quantified variables by a skolem function f whose arguments are all of those universally quantified variables that precede the existential variable. For example, the formula $\forall x \exists y Likes(x, y)$ can be skolemized into $Likes(x, f(x))$ where f is a skolem function. If there is no universal quantifier preceding an existential quantifier, a skolem function with no arguments called *a skolem constant* will be produced.

Every prenex-normal formula can be transformed into a quantifier-free clausal form of logic which is logically equivalent. To do this, a prenex-normal formula must be skolemized if it contains any existential quantifiers. A clausal form of logic can be one of the following two equivalent forms:

- Implication form

  $P_1 \wedge \ldots \wedge P_n \rightarrow Q_1 \vee \ldots \vee Q_m$

- Disjunctive form

  $\neg P_1 \vee \ldots \vee \neg P_n \vee Q_1 \vee \ldots \vee Q_m$

where P_i and Q_j are (positive) literals. When m=1, the clause is called a *Horn clause*. We will mix the use of these two forms throughout this paper.

The resolution principle [Robinson 65] is a rule of inference that can be applied to clausal forms of logic. The principle can be defined as follows.

Given two clauses,

$S_1 \vee p1 \vee S_2$

[1] This section is mainly extracted from [Chang & Lee 73].

$S_3 \vee \neg p2 \vee S_4$

where S denotes disjunctive literal set and p1a=p2a (ie. p1 and p2 are *unifiable* where "a" is a substitution called the Most General Unifier (MGU)),

we can deduce the following resolvent,

$S_{1,2,3,4}a$

Resolution is the only rule of inference that is necessary in order to find proofs to all theorems. Although resolution is complete [Robinson 65], unlimited applications of resolution may cause many irrelevant and redundant clauses to be generated. Thus many restricted forms of resolutions have been developed. One such an example is the linear resolution strategy in which one of the two resolved clauses is always the most recent resolvent. A more restricted linear resolution strategy called *linear input resolution* on Horn-clauses in which linear resolution is begun with the input goal is used in Prolog.

3 Clausalizing Beliefs

Standard first order logic is concerned with *extensional objects* or things that exist. However in a belief logic, namely, a logic supplemented with a BEL (or B) predicate, we are additionally concerned with *intensional objects* or concepts which may not have extensions (eg. unicorns). Thus unlike predicates of standard first order logic, a B predicate introduces intensional scopes that quantifiers cannot be moved outside. This can be illustrated by the following two sentences:

1. $\exists x B(Simon, Unicorn(x))$
2. $B(Simon, \exists x Unicorn(x))$.

In the first sentence, there is a particular individual in Simon's mind whom Simon believes to be a unicorn; while in the second sentence, Simon does not know which individual is a unicorn but he believes a unicorn exists. This means that we cannot have a prenex-normal form, hence a clausal form of logic to represent quantified beliefs.

To solve this problem, we propose a *levelled* intensional scheme that allows an extended clausalization to be achieved. In this scheme, each logical term is associated with a number denoting the level (or depth or nestness) of intensional scope it is *meant* to be in (the default level is 0). This is represented by using a built-in predicate structure called $LEVEL$ so that $LEVEL(x,i)$ denotes a i levelled intensional term x. For clarity reason, we write $LEVEL(x,i)$ in a subscript notation: x_i. To unify two terms, we additionally require the levels of the two terms to be unifiable.

Because quantified terms of different intensional scopes are distinguished/levelled, the intensional scheme allows quantifiers to be moved outside the scope of BEL predicates so as to produce prenex-normal forms which can then be skolemized (if there is any existential quantifier) into clauses. For example, the formula

$$\forall x B(Simon.\exists y B(John, likes(x,y)))$$

after introducing levels of intension can be transformed into the following prenex-normal form

$$\forall x_0 \exists y_1 B(Simon, B(John, Likes(x_0, y_1)))$$

which can then be skolemized into the following clause

$$B(Simon, B(John, Likes(x_0, f(x_0)_1)))$$

where f is a skolem function.

We elaborate more on the levelled intensional scheme with the representation of the four intensional interpretation of the statement 'Jim believes every unicorn likes something' (omitting the logical form before clausalization):

1. 'There is a particular thing Jim believes every unicorn likes'

   $B(Jim, Unicorn(x_1) \rightarrow Likes(x_1, c1_0))$

2. 'Jim believes every unicorn likes a particular type of thing'

   $B(Jim, Unicorn(x_1) \rightarrow Likes(x_1, c2_1))$

3. 'Jim believes every unicorn likes something of his own type'

   $B(Jim, Unicorn(x_1) \rightarrow Likes(x_1, f(x_1)_1))$

4. 'Jim believes every unicorn likes his own particular thing '

   $B(Jim, Unicorn(x_1) \rightarrow Likes(x_1, g(x_1)_0))$

where c1, c2 are skolem constants and f, g are skolem functions.

We adopt the following notations in our formalism:

- A built-predicate is denoted by an upper-case string, eg. BEL/B.
- A non-skolem term is denoted by a lower-case string preceded with a upper-case letter.
- A skolem constant is denoted by a low-case string preceded with a letter of the set {a,b,c,d,e}.
- A skolem function is denoted by a lower-case string preceded with a letter of the set {f,g,h}.
- A variable is denoted by a lower-case string preceded with a letter of the set {x,y,z}.

To demonstrate the expressiveness of our formalism, we describe its representations of some more examples.

The following sentence attributed to Russel is discussed by McCarthy [79]: "I thought that your yacht was longer than it is." It can be expressed in the clausal approach as (omitting tense and pronouns):

$Len(Yt, c1_0)$.
$B(I, Len(Yt, c2_1))$.
$B(I, c2_1 > c1_0)$.

where c1 and c2 are skolem constants. Here B is better understood as Believed rather than *Believe*.

From the example, it can be seen that conjunctive beliefs are modelled as separate clauses in our formalism, eg. the formula $B(I, A \wedge B)$ is represented as two clauses $B(I, A)$ and $B(I, B)$. Such modelling helps to remove the spurious syntactic distinction problem that often persist in the syntatic formalization of beliefs.

To express "Your yacht is longer than Peter thinks it is", we have the following formulae:

$Len(Yt, c1_0)$.
$B(Peter, Len(Yt, c2_1))$.
$c1_0 > c2_1$.

where c1 and c2 are all skolem constants.

Quine [56] discusses an example in which Ralph sees a person skulking about and concludes that he is a spy, and also sees him on the beach, but doesn't recognize him as the same person. The facts can be expressed in our formalism as:

$See(Ralph, Sk(b_1))$.
$B(Ralph, Spy(b_1))$.
$See(Ralph, onbeach(c_1)$.
$B(Ralph, Spy(c_1))$.
$b = c$.

where b and c are skolem constants.

Note that a non-skolem constant can also have a level of intension other than zero. This can be illustrated by the following two sentences:

1. $B(Simon, Loves(Mars_0, Venus_0))$
2. $B(Simon, Loves(Mars_1, Venus_1))$.

In the first sentence, Mars and Venus are extensional objects of the real world; while they are intensional concepts of Simon which may not have extensions in the second sentence.

To see how nested beliefs can be modelled with levels of intension, we illustrate with the following example:

1. $\exists x B(Simon, B(John, Unicorn(x)))$
2. $B(Simon, \exists x B(John, Unicorn(x)))$
3. $B(Simon, B(John, \exists x Unicorn(x)))$.

The intensional differences of these sentences can be explained as follows. In the first sentence, Simon has a particular individual in his mind whom he thinks John believes to be a unicorn. In the second sentence, Simon believes there is a particular individual in John's mind (though Simon may not know which one it is) whom he thinks John believes to be a unicorn. In the third sentence, Simon believes that John believes that there exists a unicorn though Simon has no idea himself or no idea about John's idea about who the unicorn is.

These sentences can be represented in our formalism as follows:

1. $B(Simon, B(John, Unicorn(c1_0)))$
2. $B(Simon, B(John, Unicorn(c2_1)))$
3. $B(Simon, B(John, Unicorn(c3_2)))$

where c1, c2 and c3 are all skolem constants.

We have so far only shown that beliefs can be represented in an extended clausal form of logic. Our formalism also allows beliefs to be nested within clauses. This can be illustrated with the following example: "Every Person (P) believes that every ET (E) believes that Unicorns (U) exist". One intensional representation of the statement in an arbitrary form of logic could be:

$$\forall x(P(x) \rightarrow B(x, \forall y(E(y) \rightarrow B(y, \exists z U(z)))))$$

This can be represented in our formalism as:

$$P(x_0) \rightarrow B(x_0, E(y_1) \rightarrow B(y_1, U(f(x_0, y_1)_2))).$$

From the above examples, it can be seen that unlike McCarthy [79] which appears to have only two levels of intension, our approach allows theoretically an infinite level of intension depending on the nestness of beliefs. Thus instead of using terms like "de-dicto" and 'de-re" to describe intensions and extensions, we talk about levels of intension in our formalism. In addition, McCarthy uses different notations to express intensions and extensions of same concepts which could complicate the first order deductive calculus, we use same notations for them whose intensions are distinguished by the levels of intensional scope they are in.

Barnden [1986] has criticized the existing models of beliefs including the modal approach (eg. [Helpern&Moses 85]), the quotation scheme (eg. [Perlis 85]) and the concept formation scheme (eg. [Creary 79]), for introducing unwarranted inferences which he call the *imputation* problem. In these models, the belief of Jim "Sue is smart" would be represented in the same way as the nested belief of Tom "Jim believes that Sue is smart". They introduce the unwarranted inference (or "opacity violation") in that Jim's mental state of smartness of Sue is the same as Tom's or as anyone else's. Undesirable imputations of a similar sort arise in the belief models described in Barwise&Perry [83], if they are extended in the natural way to deal with nested beliefs. The extension causes imputations, to ordinary agents, of beliefs about their models' "situation types'.

Barnden then proposed an alternative scheme based on Creary's concept formation scheme [Creary 79]. In Barnden's scheme, the mental state of each agent is *explicitly* denoted as a concept formation. For example, one intensional interpretation of the nested belief of Tom "Jim believes that Sue is smart" in Barnden's scheme is:

$$B(Tom, \$_{Tom}(B(\$_{Tom}Jim, \$_{Tom}\$_{Jim}(S(Sue)))))$$

where $\$_{agent}$ denotes the mental state of the agent. Although Barnden's scheme may be theoretically sound, it does not seem to be computationally viable.

It is felt that the cause of imputation of existing models of belief lies on their basis of only two levels of intension. Thus in contrast with Barnden's scheme, the intensional scheme in our formalism can distinguish the mental states of nested agents *implicitly* by their intensionally levelled terms, eg.:

$$B(Tom, B(Jim_1, S(Sue_2)_2))$$

This approach is more viable computationally than Barden's explicit scheme since intensional structures can be regarded as functional structures – a similar analogy to a Prolog *structure* [Clocksin & Mellish 81].

It should be noted that our intensional scheme and clausalization mechanism can be generalized to intensional predicates other than BEL such as WANTS, SEEKS, AWARE etc. This can be illustrated by the following example discussed in [Hobbs etal 77]: "Everyone seeks a frog". We can have the following four interpretations of the statement represented in our clausal forms as shown below.

- "There is a particular frog everyone seeks"

  $$Person(x) \rightarrow SEEKS(x, c1_0).$$

- "Everyone seeks a particular type of frog"

  $$Person(x) \rightarrow SEEKS(x, c2_1).$$

- "Everyone seeks his own particular frog"

  $$Person(x) \rightarrow SEEKS(x, f(x)_0).$$

- "Everyone seeks his own particular type of frog"

  $$Person(x) \rightarrow SEEKS(x, g(x)_1).$$

where c1, c2 are skolem constants and f,g are skolem functions.

4 Inference and Semantics

In this section, we discuss the inference mechanism, semantics, soundness, completeness, consistency and recursiveness of our formalism.

4.1 Inference

In the previous section, we have described the syntactic logical form of our formalization of beliefs, namely, an extended clausal form of logic supplemented with a BEL/B predicate. However it is pointless to talk about beliefs outside the context of a world in which the beliefs may be true or false. This means that we need rules of inference to reason about beliefs of an agent to obtain his implied beliefs.

The only inference rule in our approach is a linear resolution-like principle (such as Linear Input Resolution or , SL resolution [Hayes&Kowalski 79]) for all agents of beliefs. Exactly what this principle is, is not the concern of this paper.

To distinguish the different reasoning capability of different agents of beliefs, we assign an Inferential Power (IP) to each agent. We could have designed a more clever and complicated measurement (such as deduction, learning and memory abilities of an agent plus resource-bound factors) as the inferential power of an agent, for simplicity and illustration reason however, we have chosen the maximum inferential depth of resolutions allowed to an agent as the inferential power of the agent. So given IP(Simon,3), Simon can only invoke at most, a depth of three resolution inferences in a deductive process. By *inferential depth of resolution*, it means the depth of a AND/OR-search tree of a linear resolution proof.

For example, given the following beliefs of Simon and ID(Simon,3),

$P \leftarrow Q$
$Q \leftarrow R$

$R \leftarrow S$
S
we can infer that Simon also believes S, Q, R but not P.

To answer a query about an agent's belief, we negate the belief and prove refutationally that the negated belief is inconsistent with the belief space of the agent. For example, to prove $BEL(Simon,p)$, is to show $BEL(Simon,\neg P)$ to be inconsistent with Simon's beliefs. This means that in addition to the normal resolution rule, we need to define the following resolution rule regarding beliefs.

> Given
>
> $S_1 \vee B(agent, S_2 \vee P \vee S_3) \vee S_4$
>
> $S_5 \vee B(agent, S_6 \vee \neg Q \vee S_7) \vee S_8$
>
> where $Pa = Qa$ (where a is the MGU between the two literals) and S denotes disjunctive literal set.
>
> we can obtain the following resolvent (within the inferential power of the agent),
>
> $(S_{1,5} \vee B(agent, S_{2,6,3,7}) \vee S_{4,8})a.$

Our inference mechanism can be compared and contrasted with Konolige's approach [85] which uses multiple rules of inference for arbitrary forms of a first order logic supplemented with a B predicate. In his approach, the inferential power of each agent is determined by the set of rules of inference (or sequents) he has. As argued in the introduction, multiple rules of inference for arbitrary forms of logic may not be supported efficiently.

In addition, unlike the too-coarse-grained possible-world approach, our formalism represents beliefs as syntactic clausal structures to be manipulated and the consequential closure of an agent's beliefs is controlled by his inferential power. However unlike other systactic schemes (eg. [Konolige 85]) which suffers the problem of too-fine granularity, we use a canonical clausal form which helps to remove many spurious syntactic distinctions. In this sense, our formalism can be seen as a balance between the fine-grained syntactic approach and the coarse-grained semantic approach.

Finally, it may be noted that our inference mechanism does not have any axioms. However certain axioms may be useful. Two such ones are the positive introspection $B(a,p) \rightarrow B(a, B(a,p))$ and the negative introspection $\neg B(a,p) \rightarrow B(a, \neg B(a,p))$. Belief introspection [Konolige 85a] is useful because it allows an agent to reflect upon the workings of his own cognitive function. This can be done in our formalism by issuing recursive queries to the inference system with perhaps reduced inferential power for each agent. However this discussion is outside the scope of this paper.

4.2 Semantics

The basic syntax of our belief logic is a first order clausal form of logic supplemented with a BEL/B predicate which can take a clause as argument in the form of $B(agent, clause)$ where $clause$ can be a variable or a clause instance. In addition to the normal semantics of a standard first order clausal form of logic which we will not describe, the B predicate presents an additional semantics which allows $B(agent, clause)$ to be logically implied by a set of clauses believed by the agent. It is this type of semantics we will describe in this paper.

Because our intensional mechanism have replaced all the quantifiers by intensionally levelled variables and skolem functions, the semantics of our formalism can *uniformly* be defined as follows:

> $B(agent, clause)$ is true iff the clause is a member of the belief space of the agent.

A clause c is a member of the belief space of an agent a iff it is a IP-controlled consequence of the members of the belief space of a. This effectively determines the soundness and completeness results of the formalism which can be stated as follows.

The soundness result of the formalism is:

> if a clause c follows from the IP-controlled inference of an agent a, then $B(a, c)$ is true.

The completeness result of the formalism is:

> if $B(a, c)$ is true, then the clause c follows from the IP-controlled inference of the agent a.

Because the belief space of an agent is consequentially-closed under the inferential power of the agent, the consistency of a belief space need to be defined differently from that of a standard first order logic. In particular, we can have contradictory beliefs without causing an agent to believe anything (as is the case in the possible world approach). In our formalism, a belief space of an agent (closed under the inferential power of the agent) is consistent iff it does not contain two literals $p1$ and $\neg p2$ such that $p1$ and $p2$ are unifiable. This means that an inconsistent set of clauses in a standard first order logic, can be consistent in a belief space of an agent provided the agent's inferential power will not allow contradictions to be deduced. For example, a person may believe the following statements "Every Professsor has a PhD", "Every MD does not have a PhD" and "John is a Professor and has a MD" without realizing that they are inconsistent because he has a limited inferential power or he may have chosen relevant theories for the statements in such a way that he will not establish inconsistent beliefs. The latter will be discussed in the next section.

All the above results can be similarly applied to recursive/nested beliefs. To do this, each belief space of an agent is organized hierarchically with each sub-space of a space denoting the next level of agents in a nested belief. Thus $B(a1, B(a2, c))$ is true iff c is a member of the belief space a2 which is a sub-space of a1. However the inferential power of all the sub-agents in a nested belief is determined by the outmost agent. A more detailed description of nested beliefs can be found in [Jiang 86].

It may be noted that the basic semantics of our formalism is similar to the semantics of Konolige's [85] deductive model of belief except that we use a canonical clausal form of logic and a single rule of inference as the computational structures of an agent. From a *logic programming* perspective, our formalism may be treated as a more-viable computational model of belief than Konolige's.

5 Circumscriptive Non-monotonic Reasoning

Unlike a piece of knowledge, a belief may not be true in the real world. This means that with new beliefs in hand, an agent can retract his old beliefs. This form of reasoning is sometimes called belief revision. One possible approach of revising beliefs is to build inference paths of beliefs so that a retracted belief can be traced back from such inference paths. Its purpose is to detect contradictions, identify their causes and try to resolve the contradictions by revising beliefs. Two examples of this approach are London's dependency networks [78] and Doyle's truth maintenance system [79]. Another possible approach of revising beliefs is the explicit non-monotonic reasoning. Its purpose is not to detect the cause of inconsistency but to ensure that the belief system is always consistent. In this approach, non-monotonic beliefs are *explicitly* represented as non-monotonic rules and can only be derived if they are consistent within a certain belief space. Although it is felt that inference paths in linear resolution systems on clausal forms of logic are easier to find than arbitrary forms of logic, in this paper however, we are only concerned with the problems addressed in the latter approach.

There are currently two important types of non-monotonic reasoning. One is McCarthy's circumscription rule [80]. Another is Reiter's default rule [Reiter 80][2]. Both can be specified in the following meta-axiom:

$$x \wedge C(y) \rightarrow y$$

where $C(y)$ is true if $\neg y$ cannot be proved.

The difference between these two approaches lies in the area of consistency checking. In McCarthy's approach, it is the whole belief space including all the non-monotonic rules; while it is only the area of the belief space that excludes default rules in Reiter's approach.

The problem with McCarthy's approach is that it is *too fine-grained* in the sense that it tends to fail to conclude anything. This can be seen from the following two default beliefs:

1. if x is a professor and there is no proof that x is a Mr then we may infer that x is a Dr (Ph.D):

$$\begin{aligned} Dr(x) \leftarrow & \ Prof(x) \\ \wedge & \ C(\neg Mr(x)) \end{aligned}$$

2. if x has a MD and there is no proof that x is a Dr then we may infer x is a Mr:

$$\begin{aligned} Mr(x) \leftarrow & \ MD(x) \\ \wedge & \ C(\neg Dr(x)) \end{aligned}$$

In addition, we assume the belief that no one can both be a Dr and a Mr:

3. $\leftarrow Dr(x) \wedge Mr(x)$.

[2] Moore [85a] may argue that this is best called autoepistemic reasoning.

If we assume that John has MD, and is a professor, then in proving $Dr(John)$, we need to prove $C(\neg Mr(John))$ or $Mr(John)$ which lands us in the proving of $C(\neg Dr(John))$, or $Dr(John)$; hence back to the original goal, ie. looping.

Reiter's approach on the other hand is *too coarse-grained* in the sense that it tends to conclude everything. For instance, though it does not loop for the above example, Reiter's approach would give us both $Dr(John)$ and $Mr(John)$, ie. a contradiction (to belief (3)).

To solve these problems, we propose a neutral approach (based on [Bowen&Kowalski 82]) that covers a range or varing granularities of non-monotonic reasoning systems. Instead of being restricted in one single predefined theory/area of belief space, an agent can perform consistency checking within an explicitly defined theory. This means that in our formalism, a belief space can be divided into various overlapping areas/theories. Allowing various explicit specification of theories is cognitively feasible because an agent may only use a subset of his beliefs which he thinks is relevant to achieve a certain process of reasoning (eg. to establish the proof that John is guilty) and use another subset of his beliefs (maybe overlapping with the former set) to achieve another process of reasoning. This relevance of beliefs to an agent is sometimes called *circumscriptive relevance* [Konolige 85]. For this reason, we call our belief revision approach *circumscriptive non-monotonic reasoning*.

To define the explicit theory of a clause, we further extend the syntax of our basic logic of belief by allowing each clause to have a distinguishing label or number. The idea of attaching a distinguishing number to a wff was initiated by Godel [Enderton 72], thus born the name of *godelization* of a wff. Godel showed that the godelizations of wffs are representable. This means that we can use a number to denote a wff. In other words, by making assertions on godelizations of wffs, we can indirectly express assertions about other assertions. Thus based on the idea of godelization, we can express the explicit theory of a clause by qualifying the godelization of the clause with the theory. For example, if the clause c is in the theory $t1$, we can represent this as:

$$THEORY(\#c, t1)$$

where $\#c$ is the godelization of clause c.

However unlike [Bowen&Kowalski 82], these theories are organized in hierarchical structures in our formalism so that a clause at a higher-level theory can be inherited by a lower-level theory unless it is false there. Making the theories explicit has the advantage that subtle differences in meanings can be expressed by appropriate organization of theories.

Thus to solve the above looping and contradiction problems, we can represent the theory hierarchies in such a way that T2 is one level below T1 and T2 has belief #2 but T1 does not. This effectively assigns a higher priority to belief #1 over #2. This can be shown in our formalism as follows:

$\#1 : Dr(x) \leftarrow Prof(x) \wedge C(\neg Mr(x), T1).$
$\#2 : Mr(x) \leftarrow MD(x) \wedge C(\neg Dr(x), T2).$
$SUB-THEORY(T2, T1).$
$THEORY(\#2, T2).$
$\neg THEORY(\#2, T1).$
$\leftarrow Dr(x) \wedge Mr(x)$

Finally, it should be noted that our explicit control over the area of belief space for handling belief revision can be generalized to ordinary proofs in the spirit of *relevance logics* [Anderson&Belnap 75]. In this case, every proof must be associated with an area of a belief space. To achieve this, we introduce another meta-predicate PROVE so that $PROVE(p, t)$ stands for "p can be proved in the theory t". Relevance proofs are useful in knowledge representation. For example, it distinguishes $Guilty(x)$ from $PROVE(Guilty(x), t)$ as we should do because we may believe a person to be guilty, but we may not think we can *prove* it given the evidences we think are *relevant* to the case.

6 The consistency of self-referential paradoxes

Another expressive feature of our formalism is its ability in representing self-referential beliefs, eg. "John believes that his belief is false" and mutual-referential beliefs, eg. "Simon believes that Tom's belief is true and Tom believes that Simon's belief is false". For example, the above example of mutual belief can be represented in our godelized clausal form of belief logic as follows:

$B(Simon, \#2 : TRUE(\#3))$
$B(Tom, \#3 : \neg TRUE(\#2))$

As argued before, it is pointless to talk about people's beliefs outside the context of a world in which the beliefs may be true or false. However this could introduce inconsistencies to belief spaces. This is shown by Tarski [36] in his No Truth Definition Theorem, which states that

$$TRUE(\#a) \leftrightarrow a$$

is inconsistent. This can be seen from the example if we assume Simon's belief to be true and Simon's inferential power is capable of making the following inferences:

$$\begin{aligned} TRUE(\#2) &\rightarrow TRUE(TRUE(\#3)) \\ &\rightarrow TRUE(\#3) \\ &\rightarrow TRUE(\neg TRUE(\#2)) \\ &\rightarrow \neg TRUE(\#2) \end{aligned}$$

ie. a paradox inconsistency has arisen.

Tarksi's solution is attaching numerical subscripts or levels to 'true'. In Tarksi's approach, a truth $true_m$ is restricted to apply to sentences containing no predicate $true_n, n \geq m$. This requirement effectively blocks the derivation of contradiction. However Tarski's approach suffers the problems of inefficiency due to different operations at different levels [Warren 81], and limited expressiveness in representing beliefs [Perlis 85] such as "I have a false belief".

Kripke's solution [75] thus introduces truth gaps to account for paradoxes. However the solution incurs the invalidity of the excluded-middle principle, ie. $(p \vee \neg p)$. This could make the design of an efficient proving system difficult. Thus we take an alternative approach in the spirit of Perlis [85].

In our approach, the Tarski's Truth Definition Theorem is modified so that the axiom

$$TRUE(p) \leftrightarrow p$$

(called the *positive axiom*) holds for all positive p but may not be true for negative p. For the negative p, we adopt Gilmore's reduction rule [74] in our approach in such a way that the axiom

$$TRUE(\neg TRUE(p)) \rightarrow TRUE(\neg p)$$

(called the *negative reduction axiom*) holds for all p. To allow further reductions, we introduce another axiom (called the *restricted negative axiom*):

$$TRUE(\neg p) \rightarrow \neg TRUE(p)$$

which holds if the dereference of p involves no TRUE predicate. By dereference of p, we mean that if p is a label, then the dereference of p is the wff named by p; otherwise, the dereference of p is p itself. The restricted negative axiom allows us to deduce $\neg TRUE(EQ(1,2))$ from $TRUE(\neg EQ(1,2))$ because the dereference of $EQ(1,2)$ contains no $TRUE$ predicate; while from $TRUE(\#3)$ in the above example, we cannot deduce $\neg TRUE(\#2)$ because the dereference of #3, ie. $\neg TRUE(\#2)$, contains a TRUE predicate. Note that for the above axioms to work correctly, we need to represent a clause in the disjunctive clausal form mentioned in Section 2.

Using the *restricted negative axiom*, we can *preserve the excluded-middle principle*

$$TRUE(p) \vee \neg TRUE(p)$$

but not

$$TRUE(p) \vee TRUE(\neg p).$$

In other words, we cannot have both $TRUE(p)$ and $\neg TRUE(p)$ in a consistent belief space, but we can have both $TRUE(\neg p)$ and $TRUE(p)$ in a consistent belief space. The fact that $TRUE(p) \wedge TRUE(\neg p)$ holds, helps to reveal a paradox without letting this create an inconsistency to our formalism. The price is simply that we stick literally with what the statements express, and this inconvenience will be as rare as are these sentences in typical discourse situations. A consistency proof of the modified truth definition theorem can be found in [Jiang 86].

To see how we have solved the paradox inconsistency, we use the earlier example as an illustration.

Suppose we assume that Simon is right, then we have the following inference chain,

$$\begin{aligned} TRUE(\#2) &\rightarrow TRUE(TRUE(\#3)) \\ &\rightarrow TRUE(\#3) \\ &\rightarrow TRUE(\neg TRUE(\#2)) \\ &\rightarrow TRUE(\neg \#2) \end{aligned}$$

which is consistent.

Suppose we assume that Tom is right, then we have the following inference chain,

$$\begin{aligned} TRUE(\#3) &\rightarrow TRUE(\neg TRUE(\#2)) \\ &\rightarrow TRUE(\neg \#2) \\ &\rightarrow TRUE(\neg TRUE(\#3)) \\ &\rightarrow TRUE(\neg \#3) \end{aligned}$$

which is still consistent.

In both cases, paradoxes are *revealed* and at the same time consistencies are *preserved*.

7 Conclusions

In this paper, we have presented a scheme from a logic programming perspective which formalizes beliefs in *an extended clausal form of logic*. We have shown that our formalism is free-from the quantification problem that often persists in existing formalisms. In particular, we have indicated that our formalism allows an efficient resolution-like proof procedure to be developed. A levelled intensional scheme which enables the clausalization of beliefs has been proposed. It has been argued that the intensional scheme is free from the imputation problem. An inferential power bound resolution rule of belief has been introduced. A general circumscriptive non-monotonic reasoning system for handling belief revision has been described. The concept of godelization which increases the expressiveness of our formalism has been introduced. In particular, a modified Tarski's Truth Theorem has been shown to be consistent with our formalism.

There are issues such as common beliefs and implied beliefs of *a group of agents* which have not beed discussed in this paper due to the space limit. In addition, we have neither addressed belief introspection nor *implicit* belief revision. These problems will be sbjected to further research.

As regards to implementation of our formalism, it is felt that it can be done quite easily in Prolog. In particular, an intensionally-levelled term can be represented as a structure of the form (term,level); and a godelized clause can be represented as a structure of the form (label,clause). However this discussion is outside the scope of this paper.

Acknowledgements

This work is supported by the Alvey Directorate and ICL under grant GR/D/45468-IKBS 129. It is part of a collaborative project between ICL and the Univ. of Essex and Manchester. It is a pleasure to acknowledge the many discussions I have had with my collegues in Essex, Manchester and other institutes. In particular, I like to thank Peter Aczel, Alan Rector, David Warren (Manchester), Simon Lavington, Raymond Turner, Sam Steel (Essex), John Barnden (Indiana) and Richard Frost (Glassgow) for their helpful discussions.

References

A. Anderson & N. Belnap (1975) *Entailment:The logic of relevance and necessity*. Vol.1 Princeton Univ. Press.

J. Barnden (1986) *Interpreting propositional attitude reports: towards greater freedom and control* Proc. 7th European Conf. on AI, July, 1986.

J. Barwise & J. Perry (1983) "Situations and attitudes". MIT Press.

K. A. Bowen and R. A. Kowalski (1982) *Amalgamating language and meta-language in logic programming.*. In Logic Programming, ed. Clark and Tarnlund, pages 153-172, Academic Press, 1982.

L.F. Cerro (1986) *MOLOG: A system thay extends Prolog with modal logic*. New Gen. Computing 4 pp.35-50.

C.L. Chang & R.C.T. Lee (1973) *Symbolic logic and mechanical theorem proving*. Academic Press.

W.F. Clocksin and C.S Mellish (1981) *Programming in Prolog*. Springer Verlag.

L.G. Creary (1979) *Propositional attitude: Fregean representation and simulative reasoning*. Proc. of 6th IJCAI. Tokyo, Japan, August.

J. Doyle (1979) *A truth maintenance system.*, AI. Vol. 12, No.3, pp.231-272.

H. Enderton (1972) *A mathematical introduction to logic*. Academic Press. 1972.

P. Gilmore (1974) *The consistency of partial set theory without extensionality*. ed. T. Jech, Axiomatic set theory, Amer. Math. Soc. Providence RI 1974,

J. Hintikka (1962) *Knowledge and belief*. Ithaca, New York, Cornell Univ. Press.

Y.J. Halpern, Y. Moses (1985) *A guide to the modal logics of knowledge and belief: preliminary draft.* IJCAI 85, Vol.1

P.J. Hayes & R. Kowalski (1969) *Semantic trees in automatic theorem proving.* Machine Intel. 4, Edinburgh Univ. Press, pp.87-101.

J.R. Hobbs & S.J. Rosenschein (1977) *Making computational sense of Montague's intensional logic.* AI 9 pp.287-307.

Y.J. Jiang (1986) *A formalism for representing qualified knowledge and its implementation for large knowledge bases* Ph.D thesis, Comp. Sci. Dept., Univ. of Manchester, 1986.

A. Kobsa (1985) *Using situations and Russellian attitudes for representing beliefs and wants.* IJCAI 85, Vol.1.

K. Konolige (1985) *Belief and incompleteness.* in Formal theories of the commonsense world, Ed. J.R. Hobbes, R.C.Moore. Ablex Pub. Corp. 1985. pp.359-403.

K. Konolige (1985a) *A computational theory of belief introspection.* IJCAI 85 Vol. 1.

R. Kowalski (1983) *Logic Programming..* IFIP 83, pp.133-145.

S. Kripke (1975) *Outline of a theory of truth.* J. Philosophy 72, pp.690-716.

H.J. Levelsque (1984) *A logic of implicit and explicit belief.* Proc. National Conf. on Artificial Intelligence. pp.198-202.

H.J. Levelsque (1984a) *The logic of incomplete knowledge bases.* On Conceptual Modelling, eds M.Brodie, J.Mylopouos&J.Schmidt, Springer-Verlag, 1984, pp.165-187.

P. London (1978) *Dependency networks as representation for modelling general problem solver.* Ph.D. Dissertation, Tech. Report, Dept. of Comp. Science, Univ. of Maryland, 1978.

J. McCarthy (1979) *First order theories of individual concepts and propositions.* Machine Intel. 9, Eds, J.Hayes, D.Michie, L. Mikulich, Ellis Horwood.

J.McCarthy (1980) *Circumscription-A form of non-monotonic reasoning.* AI 13, pp.27-40, 1980.

R.C. Moore & G. Hendrix (1979) *Computational models of belief and the semantics of belief sentences.* SRI Technical Note 187, Menlo Park CA.

R. C. Moore (1985) *A formal theory of knowledge and action.* in Formal theories of the commonsense world, Ed. J.R. Hobbes, R.C.Moore. Ablex Pub. Corp. 1984.

R.C. Moore (1985a) *Semantic considerations on non-monotonic logic.* AI. Vol. 25, pp.75-94.

D. Perlis (1985) *Languages with self-reference I: foundations..* Artificial Intel. 25, pp.301-322, 1985.

W.V. Quine (1956) *Quantifiers and propositional attitudes..* Journal of Phil. 53.

R.Reiter (1980) *A logic for default reasoning* AI 13, pp.81-132,1980.

J.A. Robinson (1965) *A machine oriented logic based on the resolution principle.* in JACM, Vol 12, pp.23-41.

A. Tarski (1952) *The semantic conception of truth.* Semantics and Philosophy. (L. Linsky ed) Univ. of Illinois.

D.H. Warren (1981) *Higher order extensions to Prolog- are they needed?.* in Machine Intel. 10, Ellis Horwood.