REASONING IN THE PRESENCE OF INCONSISTENCY

Fangzhen Lin

Department of Computer Science
Hua Chiao University, Fujing, P.R.China

ABSTRACT

In this paper, we propose a logic which is nontrivial in the presence of inconsistency. The logic is based on the resolution principle and coincides with the classical logic when premises are consistent. Some results of interesting to Automated Theorem Proving are a sound and sometimes complete three-valued semantics for the resolution rule and a refutation process which is much in the spirit of the problem reduction format.

1. Introduction

There are at least two considerations in Computer Science and Artificial Intelligence force us to study nontrivial reasoning in the presence of inconsistency. In database system, we certainly do not wish our system to be wrecked by a single contradiction offered by the user and we often need to draw some conclusions about objects which are irrelevant to the contradiction because a contradiction is often difficult to be detected and corrected. In AI, there are many efforts to formalize common sense reasoning, for example, [McCarthy, 1980,1986], [Reiter, 1980]. A general rule for explaining common sense reasoning may fail sometimes. For example, according to closed world assumption (Reiter(1978)), a positive literal is not true if it is not a consequence of the facts in a database, so if we have a database expressed by the formula P(a)\/P(b), then we will run to contradiction by using the closed world assumption for we can infer -P(a)/\-P(b). Therefore reasoning in the presence of inconsistency seems necessary in common sense reasoning (the above contradiction caused by closed world assumption can be avoided by using circumscription [McCarthy, 1986], but as Davis showed in [Davis, 1980] that circumscription can also cause inconsistency).

Systems that are not wrecked by contradictions have been studied by Philosophers , Logicians and Computer Scientists for years [da Costa, 1974, Belnap, 1976, Priest, 1979 and Martins and Shapiro, 1986].

In this paper, we propose a logic satisfying the following three conditions: in the following, L is the new logic, G |+ A means A can be infered from G in L and G |- A means A can be infered from G according to classical logic, where A is a formula and G is a set of formulas.

(A) if G is a consistent set of formulas in the sense of first order logic, then for any formula A, G |+ A iff G |- A.

(B) the problem of deciding whether G |+ A is true for finite G is partial solvable, that is, the set {(G,A) | G |+ A} is recursively enumerable.

(C) for any finite G, there is a formula A such that G |+ A is not true.

The reasons for the three conditions are as follows. First, the condition (C) means that the deductive relation "|+" is nontrivial in every possible case. We regulate the condition (A) because we wish the new logic L to be a faithful extension of the first order logic. Finally, the requirment (B) is necessary for the logic L to be implemented by a computer program.

The systems in [Priest, 1979] and [Martins & Shapiro, 1986] satisfy the conditions (B) and (C) but not (A). As an example of the logic L that satisfies (A) and (C) but not (B), define G |+ A iff G' |- A for every maximal subset G' of G such that G' is consistent in the sense of the first order logic. It is easy to see that (A) and (C) hold, but (B) is false for the problem of deciding whether a finite set of formulas is consistent is undecidable. Note that the relation "|+" defined here is a direct

extension of a relation defined in [Rescher, 1964] in the propositional language to the first order one.

Our main point is that various contradiction tolerant systems can be constructed by restricting the classical rule of "reduction to absurdity". According to the rule, if we can infer a contradiction from A and -B, then we assert that B can be infered from A. From this it is easy to see that every thing can be infered from a contradiction. So we must restrict the rule as: A infers B whenever we can deduce a contradiction from A and -B by using some information from -B. The idea is certainly not a new one, for example, [Dunn, 1976]. What make our work new is just a new way of formalizing the statement: using some information from -B. In Section 2, we consider the resolution principle as a general rule of "reduction to absurdity" and define a deductive relation satisfying the conditions (A) to (C) above based on the resolution principle. In Section 3, we show some connections of the results obtained in Section 2 with Automated Theorem Proving. Section 4 contains some concluding remarks.

2. Propositional Resolution

In this section, we shall focus our attentions on the propositional logic, we consider how to use the propositional resolution to obtain a logic suitable for the reasoning in the presence of inconsistency. Our terminology are those in [Chang and Lee, 1973]. In particular, a deduction of a clause C from a set of clauses S is a sequence C1,...,Cn, where Cn is C and Ci is either in S or a resolvent of clauses preceding Ci. A deduction of 0 (the empty clause) from S is called a refutation. Proof by resolution principle is a complete rule of "reduction to absurdity" in the sense that for any formulas A and B, A |- B iff the union of S1 and S2 can be refuted by the resolution rule, where /\S1 and /\S2 are conjunctive normal forms of A and -B, respectively. Therefore, in order to prevent every thing from being infered from a contradiction, we have to restrict refutations of the union of S1 and S2, this motivates the following definitions.

Definition 1. Suppose S is a set of clauses, C1,C2,...Ck is a deduction from S, for any Ci, cu(Ci), the set of clauses used in infering Ci, is defined inductively as follows:

(1) if Ci is in S, then cu(Ci) is {Ci}.

(2) if Ci is a resolvent of Cm and Cn, m,n < i, then cu(Ci) is the union of cu(Cm) and cu(Cn).

Note that in De.1, if Ci is both in S and a resolvent of Cm and Cn for some natural numbers m,n < i, or Ci is a resolvent of more than one pair of clauses preceding it, then by Definition 1, there are more than one way to compute cu(Ci). In order to avoid ambiguities, in the following, when we write down a deduction C1,C2,...,Ck from S, we shall attach a fixed cu(Ck) with it, so a deduction from S is in fact a deduction from S with a fixed way for computing cu(.). Therefore L1\/-L2, L2, L1 with cu(L1) = {L1\/-L2, L2} and L1\/-L2, L2, L1 with cu(L1) = {L1} are considered as two different deductions of L1 from S = {L1\/-L2, L2, L1}.

Definition 2. Suppose (S1,S2) is a pair of sets of clauses, a sequence C1, C2 ...,Ck is a refutation of (S1,S2) if it is a refutation of the union of S1 and S2 and there is a clause C which is both in S2 and cu(Ck).

It is conventional to transform a formula into a set of clauses. We can furthermore suppose the process of transformation is unique so that for any formula we can say the set of clauses corresponding to the formula. The function of De.2 is illustrated by the following definition

Definition 3. Suppose G is a set of formulas and F a formula, S1 and S2 are the sets of clauses corresponding to G and -F, respectively. F can be infered from G (by contradiction tolerant reasoning), written G |!- F, iff (S1,S2) can be refuted according to De.2.

For the convenience of express, in this paper, all propositions about "|!-" are stated in term of the refutationness of a pair of sets of clauses. The transformation is obvious. Two propositions come directly from the definitions.

Proposition 1. Suppose (S1,S2) is a pair of sets of clauses. If S1 is consistent then a sequence of clauses is a refutation of (S1,S2) iff it is a refutation of the union of S1 and S2.

Proposition 2. Suppose (S1,S2) is a pair of sets of clauses, if S1 and S2 have no common predicate and function symbols, then (S1,S2) can be refuted iff S2 can be refuted.

Proposition 1 and Proposition 2 correspond to the properties (A) and (C) in Sec.1, respectively. In order to see when (S1,S2) can be refuted in the case that S1 is inconsistent, we need some more definitions.

In the following, for any literal L, we write -L as the literal such that if L is the atom A, then -L is -A, and if L is the negation of the atom A, then -L is A. The following lemma about the resolution principle will play an important role in this paper.

Lemma 1. Suppose S is a set of clauses, C1,C2,...,Cn=0 is a refutation of S. For any C=L1\/...\/Lk in cu(Cn), there is a deduction of -Li from S for any i=1,2,...,k.

By the lemma, it is easy to see the following theorem is true.

Theorem 1. Suppose S1 and S2 are two sets of clauses. (S1,S2) can be refuted iff there is a clause C=L1\/...\/Lk in S2 such that for any i=1,...,k, there is a deduction of -L from the union of S1 and S2.

In order to further our study, we introduce a semantics such that the resolution rule is always sound, and sometimes complete in the semantics.

The semantics is a three-valued valuation, we define it for "-" and "\/" , other connectives are defined by definitions: A/\B = -(-A\/-B), A -> B = -A\/B. The three values are t (true), f (false) and p (?). There is no fixed meaning for "p", sometime it can be understood as true and sometime false.

-	t	f	p
	f	t	p

\/	t	f	p
t	t	t	t
f	t	f	f
p	t	f	p

The above truth tables are self-explanatory. A (three-valued) valuation v is a mapping from atoms to {t,f,p}. It is conventional to extend the domain of a valuation to the set of formulas. For any set S of formuals and formula F, F is a (three-valued) semantic consequence of S, written S |= F, iff for any valuation v, if for any member A of S, v(A) is not f, then v(F) is not f either.

Example 1. {L1, -L1\/L2} |= L2 is true, but L1 |= L1\/L2 is not true, where L1 and L2 are different literals.

Theorem 2. Suppose S is a set of clauses, C is a clause. If there is a deduction of C from S, then S |= C.

Theorem 2 shows that the resolution rule is sound within our (three-valued) semantics. The converse (completeness) of the theorem is also true if the clause C in the theorem is a literal.

Theorem 3. Suppose S is a set of clauses, L a literal. If S |= L, then there is a deduction of L from S.

In terms of "|-", Theorem 2 and Theorem 3 correspond to the following theorem.

Theorem 4. Suppose G is a set of formulas and C a clause. S is the set of clauses corresponding to G. We have

G |- -C iff S |= -C.

Note that the result of the theorem is not true if we replace S |= -C by G |= -C, that is, the process of transforming a formula to its conjunctive normal form is not truth preserving according to our three-valued semantics. In fact, the problem is the distribution laws, it is easy to see that |= A\/(B/\C) <-> (A\/B) /\(A\/C) is not true.

In a sense, our three-valued semantics is a weakening of the conventional two-valued semantics. For any sets of formulas G and formula F, it is easy to see that if G |= F, then G |- F, but the converse is not true. It is of interesting to note that the three-valued semantics can be furthermore weakened . If we just change the truth table for "\/" above so that the truth-value of A\/B is p not f when A is f and B is p or A is p and B is f, then we get a semantics which is exactly the one in [Priest, 1979]. It can be proved that for any set G of formulas and formula F, if F is a semantic consequence of G according to the new semantics (with t and p designated), then G |- F, but the converse is not true.

Now, let's see how to extend the above results to the first order level. Suppose S1 and S2 are sets of clauses. Closed(S1,S2) is the pair (ClosedS1,ClosedS2), where ClosedSi = {C | C = C1(t1,...,tn), where C1 is in Si and t1,...,tn are terms in the Herbrand domain of the union of S1 and S2}, i=1,2. For any sets S1 and S2 of clauses, (S1,S2) can be refuted iff Closed(S1,S2) can be refuted according to De.2. So for any formulas A and B, A

|- B iff (S1,S2) can be refuted, where S1 and S2 are the sets of clauses corresponding to A and -B, respectively. The three conditions in Sec.1 are still true when the relation "|+" there is replaced by "|-" here. Condition (A) is easy. Note that for condition (C) to be true, we must assume that our language be infinite, for if the language is finite, for example, there is only one predicate P(x), then it is easy to see that for any formula B, (x)(P(x)/\-P(x)) |- B is true. For the condition (B), note that Closed(S1,S2) can be refuted iff there are two finite sets S1' and S2' such that Si' is included in Closed(Si) and (S1',S2') can be refuted, i=1,2.

Finally, before we concluding this section, we would like to pointed out that relevant logics of similar spirits as the one developed in this section can be obtained by other formalisms than resolution. For example, as one of the reviewers has pointed out that the set of support theorem-proving strategy (included in MESON format, see [Loveland and Stickel, 1973]) is a convenient formalism. The other formalism we have used is 'coupled tableaux' system [Lin, 1987]. It is certainly of interesting to establish some connections among varies relevant logics which satisfy the conditions (A) to (C) above and are based on diffrent formalisms. But this is still an open problem.

3. Some Applications In Automated Theorem Proving

It is of interesting to notice that the results obtained in Sec.2 motivate a refutation process which is in the spirit of the problem reduction format and its extension: MESON format [Loveland and Stickel, 1973].

Theorem 5. Suppose S is a set of clauses, L is a literal and S1 is the subset of S such that -L does not occur in any member of S1, then there is a deduction of L from S iff there is a deduction of L from S1.

Note that Theorem 5 corresponds to the repeated goals deletion rule [Loveland and Reddy, 1981]. In fact, we consider it as the most general form of the repeated goals deletion rule in clausal form. A refutation process motivated by Theorem 5 is as follows:

(1) S can be refuted iff there is a clause C = L1\/...\/Lk in S such that for any i=1,...,k, there is a deduction of -Li from S.

(2) For any literal L, there is a deduction of L from S iff there is a clause C = L1\/...\/Lk in S1 such that for any i=1,...,k, there is a deduction of -Li from the union of S1 and S2, where S1 = {C | L\/C in S and -L not in C} and S2 = {C | C in S and neither L nor -L in C}.

(3) For any literal L, if L is in S, then there exists a deduction of L from S.

Example. S = {-P\/-Q\/R, P\/R, Q\/R, -R}

This is Example 6.1 in [Chang and Lee, 1973]. Chang and Lee used this example to show the necessarity of introducing mechanisms for reducing the useless clauses generated by the general resolution rule. Let's refute S by using the process described above:

S can be refuted if there is a deduction of R from S if there is a deduction of -Q from {-P\/-Q, P, Q}, if there is a deduction of -P from {-P}, but by (3) above, there is indeed a deduction of -P from {-P}, so S can be refuted.

Again note that the results we have obtained in this section can be easily extended to the first-order level. Let's see an example

Example. S = {(1),(2),(3),(4),(5),(6),(7)}, where (1) = -E(x)\/V(x)\/S(x,f(x)), (2) = -E(x)\/V(x)\/C(f(x)), (3) = P(a), (4)=E(a)

(5) = -S(a,y)\/P(y), (6) = -P(x)\/-V(x), (7) = -P(x)\/-C(x).

This is Example 5.22 in [Chang and Lee, 1973]. A refutation process for S when the rules (1) to (3) above are suitably extended to the first-order level looks like: (in the following, for any formula F(x), F(x)|{x=t1,...,tk} will mean that F(t1),...,F(tk) have been used in the resolution process and need not being used again).

S can be refuted if there is a deduction of -P(a) from S, if there is a deduction of V(a) from {(1),(2),(4),(5),(6)|{x=a}, (7)|{x=a}, -C(a)}, if there are deductions of E(a) and -C(f(a)) from S1={(1)|{x=a}, (2)|{x=a}, (4), (5), (6)|{x=a}, (7)|{x=a}, -C(a), -E(a)\/S(a,f(a))}, if there is a deduction of -C(f(a)) from S1, if there is a deduction of P(f(a)) from {(1)|{x=a}, (2)|{x=a}, (4), (5), (6)|{x=a}, (7)|{x=a,f(a)}, -C(a),

$-E(a) \vee S(a,f(a))\}$ if there is a deduction of $S(a,f(a))$ from $\{(1)|\{x=a\}, (2)|\{x=a\}, (4), (5)|\{x=f(a)\}, (6)|\{x=a\}, (7)|\{x=a,f(a)\}, -C(a), -E(a) \vee S(a,f(a))\}$, if there is a deduction of $E(a)$ from $\{(1)|\{x=a\}, (2)|\{x=a\}, (4), (5)|\{x=f(a)\}, (6)|\{x=a\}, (7)|\{x=a,f(a)\}, -C(a)\}$, but (4) = $E(a)$, so S can be refuted.

4. Concluding Remarks

Intuitively, as Halpern said in Halpern(1986), reasoning in the presence of inconsistency is an issue which need to be considered eventually in the design of knowledge bases for it is always possible to receive contradictory information from users. In practice, we think, few reasoning systems can infer everything from a contradiction, for example, in most Prolog implementations, a program P (which is a set of Horn clauses) answers a question ?- L (L is a literal) with "yes" iff the union of P and {-L} can be refuted by using linear input resolution with -L as the top clause iff (P,-L) can be refuted iff $P \vdash L$, according to our definitions. Therefore, the logic proposed in this paper may be considered as a formalization of the logic used by some practical reasoning systems. Conversely, we hope the results obtained in this paper would be useful in the design of practical reasoning systems.

Acknowledgement

I am grateful to Joe Halpern, Donald W.Loveland, Graham Priest and two reviewers for helpful comments on an earlier draft of this paper.

References

Belnap N.D., 1977, "A useful four-valued logic". in Modern Uses of Multiple-valued Logics (eds. G. Epstein and J.M.Dunn) Reidel,1977

Chang,C.L. and Lee,R.C.T. (1973) Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1973

da Cosda, N. (1974), "On the theory of inconsistent formal systems", Notre Dame Journal of Formal Logic XV(1974), 497-509

Dunn, J.M.(1976), "Intuitive Semantics for First-Degree Entailments and 'Coupled Trees'", Philosophical Studies 29(1976) 149-168

Davis,M.(1980), "Notes On the Mathematics of Non-Monotonic Reasoning", Artificial Intelligence 13(1980)

Halpern, J.Y. (1986), "Reasoning about Knowledge: an overview", in proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge, (ed. J.Y. Halpern) Morgan Kaufmann, 1986.

Lin, Fangzhen, (1987), "Tableau Systems for Some Paraconsistent Logics", delived to the Journal of Philosophical Logic.

Loveland, D.W. and Stickel, M.E. (1973), "A hole in general tree, some guidance from resolution theory", in IJCAI—1973.

Loveland, D.W. and Reddy, C.R. (1981), "Deleting repeated goals in the problem reduction format", J.ACM, V.28(1981), 646-661.

Martins, J.P. and Shapiro, S.C. (1986), "Theoretical foundations for belief revision", in proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge, (ed. J.Y. Halpern) Morgan Kaufmann, 1986

McCarthy, J. (1980), "Circumscription — A form of non-monotonic reasoning", Artificial Intelligence 13(1980)

McCarthy, J.(1986), "Applications of circumscription to formalizing common-sense knowledge", Artificial Intelligence 28(1986) 89-116

Mitchell, J.C. and O'Donnell, M.J. (1986), "Realizability semantics for error tolerant logics", in Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge, (ed. J.Y. Halpern) Morgan Kaufmann, 1986.

Priest, G. (1979), "The logic of paradox", J. of Philosophical Logic 8(1979) 219-241.

Reiter, R. (1978), "On closed world data bases", in Logic and Data Bases (eds. H. Gallaire and J. Minker) Plenum Press, 1978.

Reiter, R. (1980), "A logic for default reasoning", Artificial Intelligence 13(1980)

Rescher,N. (1964), Hypothetical Reasoning, North-Holland, 1964.