# VERIFICATION ISSUES IN COMBINING INDEPENDENTLY CONSISTENT THEORIES

**Windy Gambetta**
Artificial Intelligence Laboratory,
School of Computer Science and Engineering,
University of New South Wales, PO Box 1, Kensington, NSW, Australia, 2033
windy@cs.unsw.oz.au

## Abstract

This paper addresses verification issues in combining knowledge bases or theories from different sources. The knowledge may be acquired using direct knowledge acquisition or inductive learning methods. Keeping the consistency of knowledge becomes an important issue when integrating two or more knowledge bases. Several independently consistent knowledge bases may give an inconsistent system when they are combined. The inconsistencies which could be syntactical or semantic are described in this paper and some suggestions to handle them are given. This process can be used to refine knowledge bases, in terms of validity and completeness, and to obtain a larger integrated system.

## 1. Introduction

Expert systems have been widely used in various areas. They have become the most successful area of Artificial Intelligence. Real systems with large knowledge bases[1] are not uncommon, so correctness is important. The careless design and implementation of knowledge acquired using various techniques can give problems ranging from minor to major [1]. Because of this, verification and validation of knowledge bases has become an active research area.

Many verification techniques have been developed to reveal anomalies but are only suitable for verifying small to medium knowledge bases, systems with several hundreds rules[2]. For larger knowledge bases, these techniques do not give a good performance because their time complexity is exponential especially when checking global consistency [2, 3]. As an example, COVER [3] needs 45 minutes to check of a knowledge base with 150 rules and 3.5 hours for 550 rules. One alternative is to use distributed verification [4] but the complexity appears to remain exponential.

This paper deals with the verification problem of combining knowledge bases where several knowledge bases which have been verified will not always give a consistent system when integrated. Conflicting views about some concepts may occur because the expertise may differ, or more commonly the knowledge bases are far from complete; the knowledge bases may perform well but may use concepts differently.

## 2. Related Research in Integrating Knowledge Bases

The integration process is not widely studied. Most knowledge bases are built individually rather than resulting from integration. They are based on either direct knowledge acquisition, inductive algorithms or case based reasoning; and perhaps using more than one expert. A few approaches to integration have been studied.

Brazdil and Torgo [5] conducted an experiment to develop an integrated knowledge base. They divided a set of cases into several subsets and the applied the inductive algorithms, ID3 and AQ11 to generate several individual theories. The theories were tested to measure the "quality" of each rule (i.e. the number of test cases successfully classified). Based on this quality, an integrated theory was developed that consists of the best rules from the sub knowledge bases. It is clear that this approach is only applicable if the knowledge bases to integrate have the same set of attributes and suitable test cases are available.

Ngwenyama and Bryson in [6] have a similar approach to Brazdil and Torgo. Despite using knowledge bases from cases, their knowledge which came from different experts still has to have the same attributes. Their propositional systems are written in tabular form.

Another paper concerning integrated knowledge bases is by Eggen et al [7]. A project called ACKnowledge which is developing a knowledge engineering workbench includes an integration method. The method is based on Murray's model

[1] Knowledge base and theory in this paper have the same meaning

[2] The knowledge bases dealt with in this paper are rule based.

[8] where one knowledge base is added rule by rule to another knowledge base so that a rule is only added if it does not conflict with the knowledge base being added to. If it is not the case, the user is asked to resolve the conflict. The problem in this method is how to choose the main knowledge base and how to order the rules. In a real situation, if we want to join two knowledge base it is very difficult to measure which one is more important. The method will work regardless, but the effort required to verfiy the system will depend on the choce of main knowledge base and the order in which rules are added.

Unlike the aforementioned proposals, Baral, Kraus, Minker, and Subrahmanian in [9, 10] combined theories in first order predicate calculus. This approach is theoretical and has some differences in how inconsistency problems are viewed from the knowledge based system view point.

Blackboard approaches [11] also can be viewed in terms of knowledge base integration. Instead of trying to develop one integrated system, systems work together to solve the problem. They work by changing common data using the blackboard.

Van Someren et al [12] suggest integrating paradigms of knowledge acquisition. In their paper, paradigms to develop knowledge bases are mentioned and analysed. Based on the result, they argue that an integrated system can be developed and will be better than those developed using one of the paradigms.

## 3. Verification of Knowledge Bases

Verification of knowledge bases deal with the process of evaluating the internal consistency of the knowledge. Verification normally assumes that terms used in the system represent mutually exclusive concepts which have no other relation with other concepts except the explicit relations represented by rules. It is only concerned with syntactical consistency. It means that any term / symbol used to represent particular concept differs from any other term / symbol. In combining knowledge bases the issue of semantic consistency arises, i.e. the relationship among concepts other than those which are represented as rules.

### 3.1. Syntactical Consistency

Many references about anomalies in knowledge base [4, 13-17] identify different types of problem, but these can be reduced to four classes - redundancy, circularity, contradiction, and deficiency.

a. A rule is redundant if it can be derived using inference procedures from some other rules.

b. Circularity is defined as the problem of the existence of a chain of dependency of rules that can lead to an infinite loop.

c. Contradiction occurs when an illegal situation, a conflicting conclusion happens because of the firing of some rules. An example of conflicting conclusions is two complementary conclusions from one set of permissible input.

d. Deficiency means that the knowledge base gives no meaningful conclusion, and does not reach final hypothesis for a permissible input.

### 3.2. Semantic Consistency

The semantic view of knowledge base focusses on relationships which are probably not covered by rules. Some obvious relationships are :

a. Synonym

Two different terms are synonyms if they have the same meaning. For instance, *student* and *scholar*. Even here there may be a more complex relationship if *scholar* in one knowledge base refers to both post graduates and undergraduates while in the other *student* refers to undergraduates.

b. Antonym

Two different terms are antonyms of each other if they refer to conflicting concepts. For instance, from the Garvan ES-1 expert system [18], TSH_HIGH is an antonym of TSH_LOW. Rather than two boolean variables these concepts would be better represented by ordinal values for a single variable, TSH.

c. Positive Constraint

Two terms or more are in a positive constraint if they always both apply. For instance if A and B are in positive constraint, it means that every time A happens so does B.

d. Negative Constraint

A negative constraing is where a set of term can not happen together. If some of them happen then the rest have to be false. For instance the relationship between MALE and PREGNANT. If MALE is true then PREGNANT has to be false, and vice versa.

Semantic constraints representing semantic consistency could be declared in the knowledge base using rules or meta rules. The implicit assumption taken in designing the knowledge base is that the system is closed: constraints exist only if explicitly defined. The introduction of semantic consistency affects verification and the size of the verification prooblem space, e.g. synonymous and antonymous terms relate to redundancy and circularity.

## 4. Proposal for Verification in the Integration Process

The process of verification in integrating knowledge bases can be divided into a number of steps, such as

1. selecting parts of knowledge to merge,
2. integrating these parts,
3. checking for anomalies in the result and then
4. merging the remaining knowledge with the result of step 3.

In this paper, we assume a knowledge base is a set of rules in Horn clause form with facts/findings as conditions and hypotheses as conclusions. Some conclusions are final hypotheses if they never appear as conditions in a rule. It is clear that a knowledge base can be represented as a set of rules which map facts into final hypotheses, e.g. $H_i \leftarrow (f_i)$, $H_i$ is a final hypothesis, and $(f_j)$ is set of facts which conclude $H_i$. Colomb has shown that all propositional expert systems can be reduced to table form [19].

### 4.1. Selecting part of the knowledge to merge

There are a number of strategies which could be use to select which knowledge to merge.

a. Exhaustive Integration

All of knowledge is put together and considered in the verification process. In this strategy we assume that all part of the systems are affected by the integration process so that we have to check a whole integrated system. This simple strategy will take a long time because we have to check all of knowledge.

b. Feed-forward Integration

In this strategy one knowledge base is chosen as the background knowledge while rules from other knowledge bases are added one by one as input to the main knowledge. Although the consistency of the main knowledge base can be kept through the feed-forward integration process by checking the effect of the incoming rule on all parts of the main knowledge, the difficulty of the process will depend on the the choice of the main knowledge base and the order in which rules are added. These choices are not clear cut.

c. Common-term Integration

The minimisation of the knowledge to be considered in integrating and checking its internal consistency is the motivation behind this strategy. The integration process is focussed on those parts of the knowledge that may interact.

Common term integration, the method we propose in this paper, tries to minimise the integration task. In this regard it should be more consistent than feed forward or rule at a time integration where task minimisation depends on difficult choices of knowledge base and rule ordering. Common term integration minimises the task because it only considers the relevant parts of the knowledge bases. Only in the worst case does the common term integration method need to consider the whole knowledge base. This methods also ensures completeness because all of the relevant knowledge is checked together. The verification process will consider all the relations among the common terms at the same time.

There are two ways to find the common terms: syntactical and semantic. In the syntactical approach a term **X** is selected as a common term if the two knowledge bases contain **X** as either fact or final hypothesis. Here, the same term from any knowledge base is assumed to represent the same concept. On the other hand, in the semantic approach use of the same terms does not guarantee that they refer to the same concepts. Two different terms could represent the same concept or one term from different knowledge bases could point to different concepts. In this case, an oracle or user is needed to give information about the relation.

Using the common terms identified, we can find the intersecting knowledge which can be grouped into three classes:

a. Intersection between fact sets which identify the same concept,

b. Intersection between final hypotheses which identify the same concept,

c. Intersection between facts and final hypotheses where the conclusion derived from one knowledge base is input for the other.

### 4.2. Integrating the selected knowledge parts

The intersections in the knowledge are collected for integration. They are described in terms of relationship between final hypotheses and facts where each final hypothesis has a set of supporting facts. In addition to these relations, some semantic relations may also apply. All of the selected part are then combined using these relations to create further relationships between facts and hypotheses where appropriate. The terms with the same name but have different meanings should be renamed, while those with different names but have the same meaning obviously need to be treated as the same.

### 4.3. Checking for anomalies in the result

Having identified the intersection of the knowledge bases and drawn relations between them the next step is to check the consistency of this knowledge when combined. The main differences between checking anomalies in single knowledge bases and integrated systems is the semantic aspects. It is not sufficient to consider only syntactic aspects unless we are sure that the terms used are consistent in that each term is used uniquely to represent different concept.

Besides checking for anomalies, it is desirable to search for hidden relations in the integrated knowledge since an oracle is unlikely to know about both knowledge bases. Checking for anomalies we include refining the knowledge base using these relations.

a. Deficiency

The problem of deficiency is unlikely to occur in an integrated knowledge base if it is built from deficient-free knowledge bases. Assuming that the necessary relations have been found the integrated system will have no deficiency unless some careless changes are made when refining the system. To avoid this situation, any change made should not reduce the domain coverage of the system.

b. Redundancy

Two redundancy-free knowledge bases may give some redundant rules if they are combined. In a single knowledge base, redundancy detection can be performed only for rules with the same hypotheses but for an integrated knowledge base we also have to consider rules with different hypotheses because even though they are apparently different there may be a semantic link between them.

Some situations which would be classified as inconsistency in a single knowledge base verification are actually redundancy problems in an integrated system. As an example consider two rules $X \leftarrow (a_1, a_2, \ldots, a_k)$ and $Y \leftarrow (a_l, a_m, \ldots, a_p)$. If $(a_1, a_2, \ldots, a_k) = (a_l, a_m, \ldots, a_p)$, then in single system these rules can be classified as conflicting because X is not the same as Y, but if they are from two knowledge bases there would be redundancy if there was a synonym relation between the hypotheses or inconsistency if there are antonym relation. Redundancy would also be the correct analysis if there was a subsumption relation between the hypotheses. The oracle would have to be involved in resolving such problems.

c. Conflict

Like redundancy problems, checking conflicting rules involves semantic relations. Because of this, some rules can contain this anomaly if they depend on conflicting facts which break some constraints. Joining two inference paths from different knowledge bases reflecting different views may give such conflicts.

For a set of rules conflicts can be classified in three ways. Firstly, a set of rules with the same final hypotheses conflict if some facts for the hypotheses conflict. Secondly, a set of rules with conflicting hypotheses conflict if the fact set of any rule is subset of those of other rules in that set. Thirdly, contradiction may occur when sets of rules have the same fact set but with different hypotheses. As mentioned before, these hypotheses may represent the same concept. If it is the case then no conflict exists but redundancy.

d. Circularity

For each common term each fact in the fact set is tested by substituting it with the fact set from which the fact to be replaced can be concluded. If the result contains terms which includes its header then a circular path is detected. As an example, $A \leftarrow (B, C, D)$ and $B \leftarrow (A, D, X)$,will give a combined rule $A \leftarrow (A, D, X, C)$. This can happen if some facts from one knowledge base become final hypotheses for the other knowledge base, and vice versa.

### 4.4. Merging the remaining knowledge with the integrated part

The last step is merging the integrated knowledge with all of the non-selected knowledge. To avoid conflicts which may occur because of terms used as intermediate hypotheses, names have to be kept unique. This can be performed simply by including in the name some identification of the knowledge base it come. For instance, an intermediate hypothesis A from the knowledge base 1 may be renamed into A1.

## 5. Discussion

In this paper some aspects of verification in integrating knowledge are addressed. The verification process has to be extended if we are dealing with an integrated knowledge base which is produced from several smaller knowledge bases which are independently consistent. The syntactic and semantic anomalies are the result of different views of the domain taken by the knowledge engineer (and expert) when designing the system. As reported by Shaw [20] different experts may not agree with each other as they may use different terminology for the same concept or the same terminology for different concepts,. Even one single expert may create this problem when dealing with the same concepts at different times. Because of this problem, the verification of integrated knowledge bases can not be fully automatic. Semantic relationships among terms used must be found to enable anomalies in the system to be discovered.

These relations can be obtained manually or possibly semi-automatically. In a fully manual method the oracle or user will be asked to supply these relations in terms rules or meta rules. The verification process will use this information to check the consistency. On the other hand, in computer-aided mode the identification of these relations could be guided by an inductive logic program which will try to find the relations and ask the user to verify the relation it suggests. By this interaction, anomalies can be removed and some hidden concepts can be uncovered [21, 22]. Inductive logic programming would appear to be feasible for this task particularly if the type of relations to be discovered are all fully identified as background knowledge. However, at this stage the utility of ILP for this task is an hypothesis.

In addition to joining knowledge bases, the integration process can also be used for verifying a large knowledge base. The system is divided into partitions using some clustering method so that related rules are grouped. The partitions created are considered as knowledge bases, verified independently and then the integration process applied. A good partition algorithm will then minimise the knowledge parts to be considered in integration because each partition will be relatively independent. This is a divide-and-conquer approach where a problem is divided into sub problems which are easier to handle. The partitions will be easier to verify than the whole knowledge base because of their size. This conjecture remains to be proven.

Although it looks promising, the idea of integration should be tested using some actual knowledge bases. The next step in our research is to implement and test this idea against real knowledge bases. A crucial issue will be the use of Inductive logic programming techniques to enhance the process of uncovering relations among concepts. Another consideration to take into account is defining some characteristics of systems to integrate.

## 6. Acknowledgment

I would like to thank my supervisor, Mr. Paul Compton, for his helpful comments.

## References

1. Geismann, J.R. and R.D. Schultz, Verification and Validation of Expert Systems, *AI Expert*, 1988, 3(2): p. 26 - 33.

2. Ginsberg, A., A New Approach to Checking Knowledge Bases for Inconsistency and Redundancy. In *Proceedings of The Third Annual Expert Systems in Government Conference*, 1987, Washington DC.

3. Preece, A.D., R. Shinghal, and A. Batarekh, Verifying Expert Systems : A Logical Framework and a Practical Tool, *Expert Systems With Applications*, 1992, 5: p. 421 - 436.

4. Zlatareva, N.P, Distributed Verification : A New Formal Approach for Verifying Knowledge-Based Systems. In *Proceedings of The World Congress on Expert Systems*, 1991, Orlando, Florida, USA: Pergamon Press.

5. Brazdil, P.B. and L. Torgo, Knowledge Acquisition via Knowledge Integration. In *Current Trends in Knowledge Acquisition*, B. Wielinga, *et al.*, editor, 1990, IOS: Amsterdam. p. 90 - 104.

6. Ngwenyama, O.K. and N. Bryson, A Formal Method For Analysing and Integrating The Rule Sets of Multiple Experts, *Information Systems*, 1992, 17(1): p. 1 - 16.

7. Eggen, J., A.M. Lundteigen, and M. Mehus, Integration of Knowledge from Different Knowledge Acquisition Tools. In *Current Trends in Knowledge Acquisition*, B. Wielinga, *et al.*, editor, 1990, IOS: Amsterdam, p. 123 - 142.

8. Murray, K.S. and B.W. Porter, Developing a tool for knowledge integration: Initial Results. In *Proceedings of The Third Knowledge Acquisition for Knowledge-based Systems Workshop*, 1988, Banff, Canada.

9. Baral, C., S. Kraus, and J. Minker, Combining Multiple Knowledge Bases, *IEEE Transactions on Knowledge And Data Engineering*, 1991, **3**(2): p. 208 - 220.

10. Baral, C., *et al.*, Combining Knowledge Bases Consisting of First-Order Theories, *Computational Intelligence*, 1992, **8**(1): p. 45 - 71.

11. Nii, H.P, Blackboard Systems: The Blackboard Model of Problem Solving and The Evolution of Blackboard Architectures (Part One), *The AI Magazine*, 1986, Summer 1986: p. 38 - 53.

12. Van Someren, M.W., L.L. Zheng, and P. Wilfried, Cases, models or compiled knowledge? - a comparative analysis and proposed integration. In *Current Trends in Knowledge Acquisition*, B. Wielinga, *et al.*, editor. 1990, IOS: Amsterdam. p. 339 - 355.

13. Yu, X. and G. Biswas, CHECKER: An Efficient Algorithm for Knowledge Base Verification. In *Proceedings of The Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems,* IEA/AIE-90, 1990.

14. Preece, A.D., R. Shinghal, and A. Batarekh, Principles and Practice In Verifying Rule-based Systems, *The Knowledge Engineering Review*, 1992, **7**(2): p. 115 - 141.

15. Chang, C.L., J.B. Combs, and R.A. Stachowitz, A Report on The Expert Systems Validation Associate (EVA), *Expert Systems With Applications*, 1990, **1**(1): p. 217 - 230.

16. Harrison, P.R. and P.A. Ratcliffe, Towards Standards for the validation of Expert Systems, *Expert Systems With Applications*, 1991, **2**: p. 251 - 258.

17. Nazareth, D.L, Issues in The verification of Knowledge In Rule-based Systems, *International Journal of Man-Machine Studies*, 1989, **30**: p. 255 - 271.

18. Horn, K.A., *et al.*,An Expert System For The Interpretation of Thyroid In A Clinical Laboratory, *The Australian Computer Journal*, 1985, **17**(1): p. 7 - 11.

19. Colomb, R.M. and C.Y. Chung, Very Fast Decision Table Execution of A propositional Expert System. In *Proceedings of AAAI-90 The Eight National Conference on Artificial Intelligence*, 1990, AAAI Press.

20. Shaw, M.L.G, Validation in A Knowledge Acquisition Systems with Multiple Experts. In *Proceedings of The International Conference on Fifth Generation Computer Systems*, 1988, Tokyo: ICOT.

21. Muggleton, S, Duce, An Oracle Based Approach to Constructive Induction. In *Proceedings of International Joint Conference on Artificial Intelligence*. 1987, Milan, Italy: Morgan Kaufmann.

22. Muggleton, S. ,editor, Inductive Logic Programming, 1992, Academic Press: London.