Fox, J., Parsons, S., Krause, P., and Elvang-Goransson, M. (1993) A generic framework for uncertain reasoning, in *Qualitative Reasoning and Decision Technologies*, N. Piera Carreté and M. G. Singh eds., CIMNE Press, Barcelona.

Ginsberg, M. L. (1987) *Readings in non-monotonic reasoning*, Morgan-Kaufmann, San Mateo, CA.

Guidi, J. N. and Roderick, T. H. (1993) Inference of order in genetic systems, *Proceedings of the First Conference on Intelligent Systems for Molecular Biology*, Bethesda, MD.

Harley, E. and Bonner, A. J. (1994) A flexible approach to genome map assembly, *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, Stanford.

Hearne, C., Cui, Z., Parsons, S., and Hajnal, S. (1994) Prototyping a genetics deductive database, *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, Stanford.

Krause, P., Ambler, S., and Fox, J. (1993) The development of "a logic of argumentation", in *Advanced Methods in Artificial Intelligence*, B. Bouchon Meunier, L. Valverde and R. Yager eds., Springer-Verlag, Berlin.

Krause, P., Ambler, S., Elvang-Goransson,M., and Fox, J. (1994) A logic of argumentation for reasoning under uncertainty, *Computational Intelligence* (to appear).

Kwan, S., Olken, F., and Rotem, D. (1993) Uncertain, incomplete and inconsistent data in scientific databases, *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems*, Catalina Island.

Motro, A. (1993) Sources of uncertainty in information systems, *Proceedings of the 2nd Workshop on Uncertainty Management and Information Systems*, Catalina Island.

Ng, R. T. and Subrahmanian, V. S. (1992) Probabilistic logic programming, *Information and Computation*, 101, 150-201.

Parsons, S. (1993) Qualitative approaches to reasoning under uncertainty, Ph. D. Thesis, Queen Mary and Westfield College, University of London (to be published by MIT Press).

Parsons, S. (1994a) Imperfect information and databases, Technical Report, Advanced Computation Laboratory, Imperial Cancer Research Fund.

Parsons, S. (1994b) Hybrid models of uncertainty for protein topology prediction, *Applied Artificial Intelligence*, (to appear).

Pitarelli, M. (1994) An algebra for probabilistic databases, *IEEE Transactions on Knowledge and Data Engineering*, 6, 293-303.

Prade H., and Testemale C. (1987) Representation of soft constraints and fuzzy attribute values by means of possibility distributions, in *Analysis of Fuzzy Information, Volume 2*, J. C. Bezdek ed., CRC Press, Boca Raton, Florida.

Reiter, R. (1978) On closed world databases, in *Logic and Databases*, H. Gallaire and J. Minker eds., Plenum Press, New York

Saffiotti, A. (1987) An AI view of the treatment of uncertainty, *The Knowledge Engineering Review*, 2, 75-97.

Saffiotti, A., Parsons, S., and Umkehrer, E. (1994) A case study in comparing uncertainty management techniques, *Microcomputers in Civil Engineering, Special Issue on Uncertainty in Expert Systems*, (to appear)

Shenoy, P. P., and Shafer, G. (1990) Axioms for probability and belief function propagation, in *Uncertainty in Artificial Intelligence 4*, R. D. Shachter, T. S. Levitt, L. N. Kanal and J. F. Lemmer eds., North-Holland, Amsterdam.

Smithson, M. (1989) *Ignorance and uncertainty: emerging paradigms*, Springer-Verlag, New York.

Williams, M. H., and Kong, Q. (1988) Incomplete information in a deductive database, *Data and Knowledge Engineering*, 3, 197-220.

Zadeh, L. A. (1965) Fuzzy sets, *Information and Control*, 8, 338-353.

Zadeh, L. A. (1978) Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, 1, 1-28.

sequential loss of speed. This might be of concern in systems with a large number of deductive rules, but could be offset by building the argumentation engine into the database's native inference mechanism. A related issue arises because, for argumentation to work correctly, it is necessary to build every possible argument for a given proposition in order to ensure that flattening gives the correct support. If this is not done the non-monotonic nature of the flattening function may over- or under-estimate the degree to which the proposition is warranted. Since many rules will generate may different arguments, this may cause problems in large systems. The third issue is that of using the different methods of handling imperfect information in a way that is more closely integrated than at present, a subject also addressed by Benferhat [1994], which we intend to persue further in the future. Fourthly, there is a question of how to combine numerical values with symbolic values such as + and ++. This may be addressed in a number of ways, both general and formalism dependent [Parsons 1993]. Finally, there is the issue of handling the other types of imperfect information, namely imprecision and vagueness, that we introduced at the beginning of the paper. At present we are unsure how argumentation can be of use here, but given the generality of the approach we are optimistic about its potential.

Acknowledgement

The authors wish to thank Paul Krause for valuable discussions and other contributions to the paper.

References

Beneferhat S. (1994) Handling hard rules and default rules in possibilistic logic, *Proceedings of the International Conference on Information Processing and the Management of Uncertainty*, Paris.

Buckles, B. B. and Petry, F. E. (1987) Generalized database and information systems, in *Analysis of Fuzzy Information, Volume 2*, J. C. Bezdek ed., CRC Press, Boca Raton, Florida.

Clark, D. A. (1990) Numerical and symbolic approaches to uncertainty management in artificial intelligence, *Artificial Intelligence Review*, 4, 109-146.

Clark, K. L. (1978) Negation as failure, in *Logic and Databases*, H. Gallaire and J. Minker eds., Plenum Press, New York

Codd, E. F. (1979) Extending the database relational model to capture more meaning, *ACM Transactions on Database Systems*, 4, 397-434.

Cui, Z. (1994) Using interval logic for order assembly, *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology*, Stanford.

Das, S. K. (1992) *Deductive databases and logic programming*, Addison-Wesley, Wokingham.

Dubois, D., Lang, J., and Prade, H. (1991) Towards possibilistic logic programming, *Proceedings of the International Conference on Logic Programming*, Paris.

Elvang-Gøransson, M., Krause, P., and Fox, J. (1993) Dialectic reasoning with inconsistent information, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, Washington D. C.

Fox, J. (1986) Three arguments for extending the framework of probability, in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer eds., North-Holland, Amsterdam.

conclusion (thus having sign +) whereas all other rules and facts are confirming (having sign ++). This suggests that models of numerical uncertainty would give similar results. Indeed, they allow slightly more subtle models to be assembled. For instance, we can use numerical uncertainty measures to represent the accuracy of the results obtained by different laboratories, so allowing the quality of their work to be taken into account when arguing for the different maps. Thus, if we have a lot of confidence in the first laboratory, slightly less in the second laboratory, and little confidence in the third, we could build Δ8 where signs are lower bounds on necessity measures.

f1 : *before(a, b)* : 0.9	Δ8
f2 : *before(x, y)* : 0.9	
f3 : *before(y, x)* : 0.9	
f4 : *before(a, x)* : 0.85	
f5 : *before(x, b)* : 0.5	
r1 : *before(X, Z)* ← *before(X, Y)* ∧ *before(Y, Z)* : 1	
r2 : *before(Y, Z)* ← *before(X, Y)* ∧ *before(X, Z)* : 0.7	
r3 : *map(X, Y, Z, W)* ← *before(X, Y)* ∧ *before(Y, Z)* ∧ *before(*Z, W) : 1	

Note the difference in signs between the default and certain rules. We can now build the following arguments:

$$\Delta 8 \vdash_{ACR} (map(a, b, x, y), (\text{f1, f2, f4, r2, r3}), 0.7)$$

$$\Delta 8 \vdash_{ACR} (map(a, b, y, x), (\text{f1, f2, f3, f4, r1, r2, r3}), 0.7)$$

$$\Delta 8 \vdash_{ACR} (map(a, x, y, b), (\text{f2, f4, f5, r3}), 0.5)$$

As before, the argument for the first and third conclusions are preferred to that for the second because they are in a higher acceptability class, but now the first is preferred to the third because it has a higher necessity measure. Clearly this is because the default rule is judged to be more credible than the results of the third laboratory— if this judgement were to be reversed, so would the preference.

5. Discussion

The framework of argumentation appears to provide a principled framework for adopting an eclectic approach for managing imperfect data in databases. However, there are issues that we have not as yet addressed.

The first issue stems from the use of minimal logic rather than full first order predicate calculus as a basis for the deductive mechanism. This is a departure from the usual framework for deductive databases and is taken because of the desire for a sound semantic basis [Krause et al. 1993]. In practice it means that using argumentation in a deductive database requires a meta-interpreter, to assemble the arguments, on top of the basic deductive system, with a conand Roderick 1993]). We also know from experiments in another laboratory that *a* is closer to the top of the chromosome than *x*. We can represent all this information using the relation "before" (see Δ7), ignoring the signs for the moment. To this we can add some very simple rules for inferring new relations between loci, and for building ordered tuples of four loci, which are rough maps of the chromosome. The first expresses the transitivity of the "before" relation, while the second comes from the fact that known pairs of loci are usually some distance apart so that, for instance, *a* and *b* will be some distance from *x* and *y* so that if *a* is before *b* and *a* is before *x*, *b* is also before *x*. This, then, is a *default* rule for coping with incomplete information about the order of two pairs of loci.

f1 : *before(a, b).*	Δ7
f2 : *before(x, y).*	
f3 : *before(y, x).*	
f4 : *before(a, x).*	
r1 : $before(X, Z) \leftarrow before(X, Y) \wedge before(Y, Z)$.	
r2 : $before(Y, Z) \leftarrow before(X, Y) \wedge before(X, Z)$.	
r3 : $map(X, Y, Z, W) \leftarrow before(X, Y) \wedge before(Y, Z) \wedge before(Z, W)$.	

From Δ7 we have:

$$\Delta 7 \vdash_{ACR} (map(a, b, x, y), (\mathrm{f1}, \mathrm{f2}, \mathrm{f4}, \mathrm{r2}, \mathrm{r3}))$$

$$\Delta 7 \vdash_{ACR} (map(a, b, y, x), (\mathrm{f1}, \mathrm{f2}, \mathrm{f3}, \mathrm{f4}, \mathrm{r1}, \mathrm{r2}, \mathrm{r3}))$$

Clearly these disagree with one another. Using the definitions in 3.2 we can see that the argument for *map(a, b, x, y)* is in class A2 because it can be *rebutted* by the argument for *map(a, b, y, x)*, but this latter argument is only in A1 since it is not consistent. Thus argumentation enables us to handle the inconsistency in our initial information. Now consider what would happen if we later learnt that *y* was before *b* as a result of another experiment in a third laboratory. Adding this fact f5 to Δ7 would give us a new argument for a new map:

$$\Delta 7 \vdash_{ACR} (map(a, x, y, b), (\mathrm{f2}, \mathrm{f4}, \mathrm{f5}, \mathrm{r3}))$$

which is also in A2 since it can be rebutted. To resolve the conflict between the arguments for *map(a, b, x, y)* and *map(a, x, y, b)* we need further information. Elvang-Gøransson et al. [1993] have suggested that the problem may be resolved by the idea of having a preference order over subsets of the database, when arguments from more preferred subsets are accepted over those from less preferred subsets. In this case it is natural to prefer the subset of the database that does not include the default rule r2 since conclusions made using defaults are less certain to be valid than others, and this ensures that the preferred map is *map(a, x, y, b)*.

It is worth noting that this solution would also be obtained if the default rule were identified as *supporting* its

f1 : *penguin(opus)* : (0.8)	$\Delta 6$
f2 : *dead(opus)* : (0.5)	
r1 : $\neg$*flies(X)* $\leftarrow$ *penguin(X)* : (1)	
r2 : $\neg$*flies(X)* $\leftarrow$ *dead(X)* : (1)	

$$\Delta 6 \vdash_{ACR} (\neg \mathit{flies(opus)}, (f1, r1), [0.8, 1])$$

$$\Delta 6 \vdash_{ACR} (\neg \mathit{flies(opus)}, (f2, r2), [0.5, 1])$$

which may be flattened using interval arithmetic to get a rather unhelpful probability of [0, 1] for Opus being able to fly. Under Krause et al's scheme, we obtain:

$$\Delta 6 \vdash_{ACR} (\neg \mathit{flies(opus)}, (f1, r1), 0.8)$$

$$\Delta 6 \vdash_{ACR} (\neg \mathit{flies(opus)}, (f2, r2), 0.5)$$

which may be flattened to get a probability of 0.8 + 0.5 – 0.4 = 0.9 for Opus being unable to fly. Better bounds than those given by Nilsson may also be obtained using the method suggested by Ng and Subrahmanian [1992]. However, we believe that the global approach is more in keeping with the spirit of argumentation.

Following Dubois et al. [1991] we may also use possibility theory. Here the signs in a database are the lower bounds on the necessity measures of the propositions, the combination operation is *minimum*, and the flattening operation is *maximum*. Thus if we take the values in $\Delta 6$ to be lower bounds on necessity measures we can infer that:

$$\Delta 6 \vdash_{ACR} (\neg \mathit{flies(opus)}, (f1, r1), 0.8)$$

$$\Delta 6 \vdash_{ACR} (\neg \mathit{flies(opus)}, (f2, r2), 0.5)$$

which may be flattened to get the conclusion that there is a lower bound of 0.8 on the necessity of Opus being unable to fly.

4. An example

We are interested in the problem of modelling information concerning the order of pieces of genetic material along a chromosome, and the use of deductive techniques to do so [Hearne et al. 1994, Cui 1994], a task that involves the use of very imperfect information [Guidi and Roderick 1993], [Harley and Bonner 1994]. In this section we show how argumentation might be used to model what information is known, and to deduce new information about orders in a way that handles some of the problems of the imperfection.

We are concerned with the order of four pieces of genetic material (known as loci in biological parlance) along a chromosome. Experiments from one laboratory tell us that *a* is closer to the top of the chromosome than *b*, while other results point to both *x* preceding *y* and *y* preceding *x* (such inconsistency is typical of genetic data [Guidi

More acceptable arguments are preferred to less acceptable ones. Thus given the database Δ5:

f1 : *bird(tweety)* : (++)	Δ5
f2 : ¬*bird(tweety)* : (++)	
r1 : *flies(X)* ← *bird(X)* : (++)	

We can see that the argument:

$$\Delta 5 \vdash_{ACR} (\mathit{flies(tweety)}, (\mathrm{f1}, \mathrm{r1}), ++).$$

is in A3 since it cannot be rebutted. However, it is not in A4 because it is open to being undercut by the argument that:

$$\Delta 5 \vdash_{ACR} (\neg \mathit{bird(tweety)}, (\mathrm{f2}), ++).$$

while this latter argument is in A2 since it can be rebutted by the argument for *bird(tweety)*. Thus the most acceptable conclusion that can be deduced from Δ5 is that Tweety flies.

3.3 Uncertain information

In the examples we have considered so far we have restricted ourselves to representing only facts that are true (and thus have sign ++), and rules that are either true and so confirm their conclusions (and have sign ++), or add support to their conclusions (and so have sign +). In general, however, the facts that we hold about the real world are not always true; it is possible to represent this by using different dictionaries of signs. In particular we can use any number in the interval [0, 1] as a sign, and by judicious choice of the combination and flattening operations used to calculate the sign of any deduced fact, we can apply various established numerical models for handling uncertainty.

For instance, if we want to make use of a probabilistic method to quantify uncertainty we have several different ways in which we can compute the probability of a proposition from the probabilities of the steps in the argument. We may use a local scheme such as that proposed by Ng and Subrahmanian [1992], or Nilsson [1988], which require the use of an interval probability because of the possible dependencies between the steps. Under Nilsson's scheme, for example, resolving a rule R with probability $p(R)$ with a fact F, probability $p(F)$, to get a fact G, requires a combination function that gives $p(G) = [p(R) + p(F) - 1, p(R)]$, and flattening two probabilistic measures $p(G)$ and $p'(G)$ from two different arguments gives $[0, p(G) + p'(G) - 2p(G)p'(G)]$. Alternatively, Krause et al. [1994] provide a global approach which takes the structure of the arguments into account. Here the probability of an argument for a proposition is the product of the probabilities of all the steps in the argument, while flattening arguments consists of summing the probabilites of the different arguments and then subtracting the product of the probabilities of the steps in all the arguments. Thus, if we have the probabilistically quantified database Δ6 in which we are not sure of Opus' status as a penguin, or even whether or not he is alive, then Nilsson's scheme allows us to infer that:

From this database we can deduce two conflicting arguments:

$$\Delta 4 \vdash_{ACR} (\textit{flies(tweety)}, (\mathrm{f1}, \mathrm{r1}), +).$$

$$\Delta 4 \vdash_{ACR} (\neg \textit{flies(tweety)} (\mathrm{f2}, \mathrm{r2}), ++).$$

Since both arguments refer to the same proposition, we can apply a “flattening function” to provide an overall opinion about Tweety’s ability to fly. One function [Fox et al. 1993] simply allows a confirming argument to outweigh any number of supporting arguments; applying this gives the conclusion that Tweety does not fly, reversing the initial conclusion. Thus argumentation permits non-monotonic reasoning [Ginsberg 1987], and so captures the kind of ability to handle incomplete information that is provided by the closed world assumption [Reiter 1978] and negation as failure [Clark 1978] commonly used in logic databases. Note, however, that the number of arguments is monotonically increasing as new facts are added to the database so that it is the flattening function that makes the conclusions drawn from the database non-monotonic.

3.2 Inconsistent information

Database Δ4 demonstrates one way of handling inconsistent information by viewing the argument for one of a pair of inconsistent conclusions as more credible than the other. A more sophisticated approach is to use the intuitive idea that arguments may be “defeated” by other arguments, examining the structure of the arguments to find conflicting alternatives, and to use the conflicts to determine which conclusions are the most acceptable.

The examination of argument structure is based upon the notions of rebuttal and undercutting. Argument *a1* for proposition *p rebuts* proposition *q*, which is supported by argument *a2*, if *p* directly contradicts *q* (in other words *p* is logically equivalent to ¬*q*). Similarly, *p undercuts q* if *p* directly contradicts *r* which is one of the steps in *a2*. We also distinguish consistent arguments, which draw facts from consistent sub-bases of the whole database, and logical arguments which are based on the axioms of the logic that underlies the system of argumentation rather than the information in the database. Having made these distinctions we can identify the following classes of arguments for a database Δ, which are listed in increasing order of acceptability [Elvang-Gøransson et al. 1993]:

A1: the set of all arguments that may be made from Δ.

A2: the set of all *logically consistent* arguments that may be made from Δ (so that ⊥ cannot be derived from the steps in the argument).

A3: the set of all arguments that may be made from Δ for propositions *for which there are no rebutting arguments.*

A4: the set of all arguments that may be made from Δ for propositions *for which there are no undercutting arguments.*

A5: the set of all *tautological* arguments that may be made from Δ.

which indicate that there is support for both Tweety and Opus flying, while it is also confirmed that Opus does not fly.

f1 : *penguin(opus)*: (++)	$\Delta 2$
f2 : *bird(tweety)*: (++)	
r1 : *flies(X)* ← *bird(X)*: (+)	
r2 : *bird(X)* ← *penguin(X)*: (++)	
r3 : ¬ *flies(X)* ← *penguin(X)*: (++)	

3. A general framework for managing imperfect information

The system of argumentation described in Section 2 may be used to handle a number of different types of imperfect information by choosing different signs, or methods of flattening the various arguments for and against propositions.

3.1 Incomplete information

In general, there are two ways to handle incomplete information, either by explicitly representing the fact that information is missing [Codd 1979], or by providing a means of making assumptions about the missing values [Reiter 1978], which may necessitate revising conclusions later (since assumptions might be contradicted by new information). In applying argumentation we take the latter approach, as illustrated by database $\Delta 3$:

f1 : *bird(tweety)* : (++)	$\Delta 3$
r1 : *flies(X)* ← *bird(X)* : (+)	
r2 : ¬*flies(X)* ← *dead(X)* : (++)	

Note that rule r1 makes assumptions about birds flying which is not always valid, but often is. From $\Delta 3$, we can construct the following argument:

$$\Delta 3 \vdash_{\mathrm{ACR}} (\textit{flies(tweety)}, (\mathrm{f1}, \mathrm{r1}), +).$$

which indicates that there is support for Tweety flying. Now, suppose we learn some new information— that Tweety is dead— so that we have an enlarged database $\Delta 4$.

f1 : *bird(tweety)* : (++)	$\Delta 4$
f2 : *dead(tweety)* : (++)	
r1 : *flies(X)* ← *bird(X)* : (+)	
r2 : ¬*flies(X)* ← *dead(X)* : (++)	

labels to arguments which denote the confidence that the arguments warrant in their conclusions. This form of argumentation may be summarised by the following schema:

$$\text{database} \vdash_{ACR} (\text{Sentence, Grounds, Sign})$$

where $\vdash_{ACR}$ is a consequence relation for a logic of argumentation, which sanctions inferences made using the rules in Figure 2 (which are adapted from those in [Fox et al. 1993]), along with the identity $\neg St \equiv \bot \leftarrow St$ ($\bot$ is logical contradiction). Infomally, Grounds (G) are the facts and rules used to infer Sentence (St) and Sign (Sg) is a number or a symbol drawn from a dictionary of possible numbers or symbols which indicate the confidence warranted in the conslusion. The use of a number of different dictionaries of signs is one of the marks of generality of argumentation since most formalisms for handling imperfect data are restricted to a single dictionary. The rules in Figure 2 are independent of the dictionary used— different dictionaries imply different functions *comb* for combining signs during the construction of arguments. Typically we will have a number of different arguments for a given sentence, and so we *flatten* these to give a single measure which may or may not be expressed using the same dictionary.

$$(\leftarrow E)\quad \frac{\Delta \vdash_{ACR} (St' \leftarrow St, G, Sg) \qquad \Delta \vdash_{ACR} (St, G', Sg')}{\Delta \vdash_{ACR} (St', G \cup G', \textit{comb}(Sg, Sg'))} \qquad (\wedge E1)\quad \frac{\Delta \vdash_{ACR} (St \wedge St', G, Sg)}{\Delta \vdash_{ACR} (St, G, Sg)}$$

$$(\wedge I)\quad \frac{\Delta \vdash_{ACR} (St, G, Sg) \qquad \Delta \vdash_{ACR} (St', G', Sg')}{\Delta \vdash_{ACR} (St \wedge St', G \cup G', \textit{comb}(Sg, Sg'))} \qquad (\wedge E2)\quad \frac{\Delta \vdash_{ACR} (St \wedge St', G, Sg)}{\Delta \vdash_{ACR} (St', G, Sg)}$$

$$(\leftarrow I)\quad \frac{\Delta \cup (St,G,Sg) \vdash_{ACR} (St', G', Sg')}{\Delta \vdash_{ACR} (St' \leftarrow St, G \cup G', \textit{comb}(Sg, Sg'))} \qquad \text{Axiom}\quad \frac{(l{:}St{:}Sg) \in \Delta}{\Delta \vdash_{ACR} (St, \{l\}, Sg)}$$

Figure 2. The rules which define the construction of arguments.

One simple dictionary includes just two signs: ++, meaning confirmed, and +, meaning supported, where confirmation is stronger than support, and combining the two signs gives the weaker. A sentence is confirmed by an argument if the argument constitutes a logical proof of the sentence, whilst support means that the argument warrants an increase in belief in the sentence. Thus, given the database Δ2 we can construct the following arguments:

$$\Delta 2 \vdash_{ACR} (\textit{flies(tweety)}, (f2, r1), +).$$

$$\Delta 2 \vdash_{ACR} (\textit{flies(opus)}, (f1, r2, r1), +).$$

$$\Delta 2 \vdash_{ACR} (\neg\textit{flies(opus)}, (f1, r3), ++).$$

This profusion of different techniques makes the situation in handling imperfect information in databases very similar to that in Artificial Intelligence a few years ago. At that time there were a number of schools of thought, each with its own techniques, which were applied to every form of imperfection irrespective of the type it had been devised to handle. The debate that ensued prompted a number of researchers [Fox 1986], [Saffiotti 1987], [Clark 1990], [Parsons 1993], to suggest that an eclectic approach, in which different techniques are used together, might be profitable. This viewpoint has gained ground in recent years, especially with the development of general frameworks in which different models may be studied [Fox et al. 1993] [Shenoy and Shafer 1990] and has been vindicated by several empirical results [Saffiotti et al. 1994], [Parsons 1994b].

The eclectic position has much to offer the database world [Parsons 1994a]. The the aim of this paper is to demonstrate this by presenting a general framework for handling imperfect information in deductive databases. In previous work we and our colleagues have developed an inference system called "argumentation" for reasoning with imperfect information [Krause et al 1994]. Here we show how argumentation may be used in deductive databases to make inferences in the presence of incompleteness, inconsistency and uncertainty, illustrating the use of our method on an example problem drawn from the domain of molecular biology. We do not address the complementary problem of how to choose the correct method of handling imperfect information in a given situation— this is addressed elsewhere [Saffiotti et al. 1994] [Parsons 1994]. In the next section we give a brief introduction to the method, before demonstrating in Section 3. how it may be used to model different forms of imperfect information in the context of a deductive database. A fuller treatment may be found in [Krause et al. 1994].

2. A system of argumentation

In a standard deductive database [Das 1992], an *argument* is a sequence of inferences leading to a conclusion. If the argument is correct, then the conclusion is true. Thus from the simple database $\Delta 1$:

f1 : *bird(tweety).*	$\Delta 1$
r1 : *flies(X)* $\leftarrow$ *bird(X).*	

the argument $\Delta 1 \vdash$ *flies(tweety)* is correct because *flies(tweety)* follows from $\Delta 1$ given the usual logical axioms and rules of inference. Thus a correct argument simply yields a conclusion which in this case could be paraphrased "*flies(tweety)* is true in the context of f1 and r1". In our system of argumentation this traditional form of reasoning is extended to allow arguments to indicate support and doubt in propositions as well as proving them, by assigning

Incomplete information has missing values, either because the value cannot be measured, is not available, or has not been obtained. Imprecise information is information that is not known with sufficient precision so that what should be single-valued attributes take disjunctions of single values. Vague information arises from the “fuzziness” of natural language when terms such as “young” are used, and uncertain information is the rest of subjective opinion in establishing whether or not something is true. Finally, inconsistent information is the result of two or more pieces of information that conflict with one another.

Name	Age	Certainty
Jack Dulouz		
Irwin Garden	25, 26 or 27	1
Bull Hubbard	Old	1
Japhy Ryder	28	0.9
Cody Pomeroy	27	1
Cody Pomeroy	92	1

Figure 1: A relational table with imperfect data

To illustrate, consider the relational table in Figure 1. The first tuple in the table is incomplete because Jack’s age is not known[1]. The second tuple is imprecise, because Irwin’s age is not known with complete precision; it is only known that he is between 25 and 27. The third tuple is vague because we only know that Bull is old, and the concept ‘‘old’’ covers a vague range of ages. The fourth tuple is uncertain because whoever entered Japhy’s age was only 90% certain that his age is 28, and the fifth and sixth tuples are inconsistent with one another because we have two contradictory, yet certain, pieces of information, one stating that Cody is 27 and one stating that he is 92.

Different types of imperfect information have typically been handled using different techniques. Thus incomplete information is often represented with explicit null values [Codd 1979]. Imprecise and vague information may be handled using fuzzy set theory [Zadeh 1965], as Buckles and Petry [1987] suggest. Uncertain information may be managed using probability theory, e.g. [Pittarelli 1994], though methods based upon possibility theory [Zadeh 1978] have also been applied [Prade and Testemale 1987]. Inconsistent information has been largely ignored. Some of these approaches have been extended to handle the problem of deriving new information in deductive databases [Dubois, Lang and Prade 1991], [Ng and Subrahmanian 1992], [Williams and Kong 1988].

1. Note that by leaving the relevant row of the table blank, we have finessed the problem of the certainty of this information, a point which is rather subtle. If we have no information about Jack’s age, are we certain that this is the case, or are we just ignorant of the whole issue? If the latter is the case, what certainty should be attached to our lack of knowledge?

Proceedings of the Workshop on Uncertainty in Databases and Deductive Systems, 1994,
(WUDDS ‘94).

A general approach to managing imperfect information in deductive databases

Simon Parsons & John Fox
Advanced Computation Laboratory
Imperial Cancer Research Fund
P. O. Box 123
Lincoln’s Inn Fields
London WC2A 3PX
United Kingdom

{sp,jf}@acl.lif.icnet.uk

Abstract

Many different models for handling imperfect information have been proposed in the context both of databases and of artificial intelligence. In general these are either symbolic methods for handling incomplete information or numerical models for handling imprecision, vagueness or uncertainty, and all typically handle a different aspect of the problem. In this paper we suggest that in order to properly model imperfect information in deductive database we should adopt an eclectic position that attempts, in a principled way, to use whatever model best fits the given data. To do this we suggest a general framework based upon previous work on a system of argumentation.

1. Introduction

To paraphrase Motro [1993], the ability to represent and manage imperfect information in information systems is important because: “Imperfect information permeates our understanding of the real world. The purpose of information systems is to model the real world. Hence information systems must be able to deal with imperfect information”. Imperfection may arise from many sources [Kwan et al. 1993], and take many forms [Smithson 1989], and the precise categorisation of the different forms has been long debated in the literature. Composing the different suggestions Parsons [1994a] categorises imperfect information as being incomplete, imprecise, vague, uncertain or inconsistent.