The Knowledge Engineering Review, Vol. 00:0, 1–22. © 2009, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

A survey of formalisms for representing and reasoning with scientific knowledge

ANTHONY HUNTER[1] and WEIRU LIU[2]

[1]*Department of Computer Science, University College London, London, UK*
E-mail: a.hunter@cs.ucl.ac.uk
[2]*School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, UK*
E-mail: w.liu@qub.ac.uk

Abstract

With the rapid growth in the quantity and complexity of scientific knowledge available for scientists, and allied professionals, the problems associated with harnessing this knowledge are well-recognized. Some of these problems are a result of the uncertainties and inconsistencies that arise in this knowledge. Other problems arise from heterogeneous and informal formats for this knowledge. To address these problems, developments in the application of knowledge representation and reasoning technologies can allow scientific knowledge to be captured in logic-based formalisms. Using such formalisms, we can undertake reasoning with the uncertainty and inconsistency to allow automated techniques to be used for querying and combining of scientific knowledge. Furthermore, by harnessing background knowledge, the querying and combining tasks can be carried out more intelligently. In this paper, we review some of the significant proposals for formalisms for representing and reasoning with scientific knowledge.

1 Introduction

Publication of scientific knowledge in articles in peer-reviewed journals and conferences is a critically important way that scientific knowledge is communicated. Whatever the field, scientists will present their findings in a textual form supported by appropriate figures, tables, and numerical data, often conforming to a structured format using an informative title, abstract, introduction with background and links to literature, methods, results, discussion and conclusion (Harmon, 2007). The approach of scientific articles has proven to be a highly effective way to exchange information. It allows the scientist to abstract away from the detailed data that may have been used to write the paper, and it allows for those findings to be discussed by the authors.

As good as this process is, there are shortcomings that arise with the rapidly increasing volume and complexity of scientific papers. This growth in the scientific literature in part reflects the massive, perhaps exponential, growth in the amount of scientific data that is now being collected (Szalay and Gray, 2007). Furthermore, as readers of these articles, scientists are under increasing pressure to find and understand them more quickly. Information retrieval techniques and semantic web technologies offer substantial assistance in finding papers more quickly (e.g. Google Scholar, PubMed), but they do not offer ways of understanding them in any sense. For that, we need to consider harnessing formal representations of scientific knowledge, and using formal reasoning systems to query scientific knowledge, to combine scientific knowledge from heterogeneous sources, to generate arguments and counterarguments from scientific knowledge, and so on.

As a subfield of artificial intelligence, knowledge representation and reasoning (KR) offers a range of formalisms for capturing knowledge, and for reasoning with that knowledge. So our starting point

in this survey is that given a scientific paper, some useful knowledge can be extracted from that paper and represented in an appropriate knowledge representation formalism. Moreover, given a collection of papers on a topic, a knowledgebase of extracted knowledge can be constructed and used with automated reasoning technology.

To illustrate, consider the area of membrane proteins which is of increasing interest both for understanding neurophysiological phenomena and for understanding pharmacological mechanisms. There is a large number of papers written on aspects of membrane proteins and there is some commonality in the kinds of information represented in classes of papers in this area. For example, there is a class of papers that consider the effect of pharmacological agents on the electrophysiological properties of membrane proteins that are known as ion channels. These refer to the class of ion channel, the *in vivo* or *in vitro* model used for the experiments, the electrophysiological stimulation and monitoring techniques used, including variations on standard procedures, the class of pharmacological agent with concentration and delivery vehicle, and a characterization of the effect on the normal electrophysiological behaviour of this class of ion channel. Whilst this class of papers is a relatively small proportion of all biomedical research papers, there is a large and important number of papers in the class, and a knowledgebase could be constructed to capture some of the useful information in the collection of papers.

Scientific knowledgebases can be authored in a KR formalism based on appropriate information gleaned from a collection of scientific papers. Furthermore, information extraction technology, which is based on parsing techniques from computational linguistics can potentially be used to assist in this process. There is also substantial background knowledge, including ontological knowledge (e.g. synonyms, more specialised terms, translations between different units, translations between different classification schema, translations between different formats/granularities for time/dates), that can be incorporated in a scientific knowledgebase. Scientific facts can also be obtained for a scientific knowledgebase by querying online databases such as now used in fields such as bioinformatics, pharmacology, and toxicology.

In this survey, we consider how we can harness KR formalisms to capture scientific knowledge. We review some KR approaches that have been adapted for capturing scientific knowledge, we consider some application studies for them in bioinformatics, biochemistry, toxicology, and clinical science, and we consider some tasks including querying and combining scientific knowledge. We also consider issues of capturing meta-knowledge about scientific knowledge, and technologies for constructing scientific knowledgebases including natural language technologies and machine learning technologies. We conclude the survey by considering lessons learnt from this consideration of the state of the art.

2 Representing and reasoning with scientific knowledge

The KR field aims to capture fundamental concepts about the world in order to represent knowledge and to reason with it in a principled way: If we are to trust a computer with some processing of knowledge, perhaps to make a decision or undertake some task, then we want to be confident that the processing is well-behaved. Furthermore, the knowledge representation and reasoning field aims for formalisms that are transparent, and reasoning that is explainable and justifiable: Users need systems to provide knowledge in a clear, understandable, and unambiguous form, and the provenance for assumptions and justifications for inferences often needs to be provided.

Once knowledge from a collection of papers is represented as a set of formulae in a KR formalism, then it can be handled using automated reasoning technology. Some of the tasks that can be undertaken with a scientific knowledgebase Δ include: (Deductive querying) Querying Δ to see if scientific knowledge α is a deductive inference from Δ; (Abductive querying) Querying to see what scientific knowledge Γ can be abduced in order to obtain α as a deductive inference from $\Delta \cup \Gamma$; (Integrity checking) Checking the consistency of Δ with a set of integrity constraints Φ; (Inconsistency analysis) Analysing the degree and nature of conflicts arising in an inconsistent scientific knowledgebase Δ; (Knowledge aggregation) Aggregating the knowledge arising in a set of scientific knowledges $\Delta_1, ..., \Delta_n$ to produce a knowledgebase Δ' with fewer conflicts and/or less redundancy.

So given a scientific knowledgebase, there is a range of reasoning tasks that can be undertaken. This allows a scientist to explore and to refine knowledge obtained from the literature. Furthermore, it allows

a scientist to consider their own findings with respect to the knowledge from the literature: Suppose the scientist has a hypothesis, or some new experimental findings, the scientist can represent this using the KR formalism used for the scientific knowledgebase, and then see whether the new knowledge is consistent with the knowledgebase, or see whether it is confirmed by the knowledgebase. There appears to be much potential in developing tools to support a scientist in such endeavours, and automated reasoning technologies can potentially enable these tools to be efficient and effective.

An important need is that for aggregating scientific knowledge to decrease redundancy, to address incompleteness in individual pieces of knowledge, and importantly, to minimize the inconsistencies and uncertainties arising in the scientific knowledgebase. The quality of scientific knowledge can be significantly improved by aggregating scientific knowledge. So the purpose of aggregation is to give a better and a more complete summary and evaluation of the sources involved.

If we take an example of testing results of a new drug, different hospitals in different regions can give different summaries of the effectiveness of the drug on different groups of people (e.g., elderly, middle-aged, young people). Since the population used in each category of each experiment may vary, aggregating these results will give a better prediction of whether the drug is effective, to what degree, and to which group of people. Furthermore, in the process of aggregating, we are comparing the contents of the sources. So if two sources are on the same subject and they are mutually conflicting, i.e. the union of them is highly inconsistent, then they reveal that either one or both sources are not correct. This can then be a qualification assigned to the evaluative information in the scientific knowledgebase that indicates there is a problem with one or both sources. This can be very useful especially as additional meta-level knowledge to help scientists undertake further investigation.

Some KR formalisms have been applied to capturing scientific knowledge, in particular description logics, logic programming, argumentation systems, and uncertainty formalisms. Here we focus on these types of KR formalism in the following four subsections.

2.1 Description logics

The notion of ontology has had a long history in science. For example, in botany, taxonomies for plant classification are a form of ontology. Here the complexity may be in the shear number of entities that need to be considered rather than the need for sophisticated constructs. There are many species of plant, but the ontology is based on a relatively small number of groupings such as phylum, class, family, etc, with "type-of" used as a relation.

Once an ontology incorporates a large number of concepts and relationships, then it gives us the ability to standardize the terminology, thereby minimizing ambiguities and facilitating communication. This is particularly important in a distributed environment where there may be numerous users who need to feel confident about the terms and concepts being used. In an information system, they can be used to assist users to navigate around the content more easily, and to support more reliable and structured maintenance of the content. Recourse to an ontology can ameliorate the complexity inherent in content in many applications by providing a common framework for structuring the content.

For instance, the Gene Ontology (www.geneontology.org) is one of the major bioinformatics resources providing a controlled vocabulary to describe gene and gene product attributes in any organism. This is a major undertaking with over 20,000 terms. It aims to describe genes, and the proteins they encode, in terms of where the proteins are located (i.e. cellular component), what they do to other molecules (i.e. molecular function), and their overall aim (i.e. biological process). For example, the protein cytochrome c can be described by the cellular component terms *mitochondrial matrix* and *mitochondrial inner membrane*, the molecular function term *oxidoreductase activity*, and the biological process terms *oxidative phosphorylation* and *induction of cell death*.

The development of ontologies for use online is becoming a key tool in many areas of science, in particular in medical science (e.g. Unified Medical Language System, MeSH, and SNOMED CT). Furthermore, sharing of ontologies that have been systematically developed is further facilitating their use (see for example The Open Biomedical Ontologies Foundry (www.obofoundry.org)).

The opportunities for ontologies are further enhanced by capturing them in a description logic. The family of description logics has been developed to represent basic ontological knowledge in logic. Each description logic is a sublogic of classical logic in the sense that each description logic uses a subset of the language of classical logic. Different description logics offer differing expressibility, and different computational properties (for an extensive review see (Baader et al., 2003)). The core idea of description logics is to represent concepts (i.e. sets of individuals) by monadic predicates, and then to define relationships between these concepts (Nardi and Brachman, 2003). Some basic relationships include the subconcept relationship using the operator $\sqsubseteq$ (e.g. the concept `Men` is a subconcept of the concept `People`, which we can represent by the statement $\texttt{Men} \sqsubseteq \texttt{People}$); intersection using the operator $\sqcap$ (e.g. the concept `Men` can be defined as $\texttt{People} \sqcap \texttt{Males}$); union using the operator $\sqcup$ (e.g. the concept `People` can be defined as $\texttt{Men} \sqcup \texttt{Women}$); and complementation using the operator $\neg$ (e.g. the concept `Males` can be defined as $\neg\texttt{Females}$).

Further information can be captured in description logics by also using binary predicates called roles (e.g. the role `Parent` is the set of tuples (x, y) where x is the parent of y). Further concepts can be defined using quantification with roles and concepts. For a role R and a concept C, existential quantification, denoted $\exists R.C$, is the set of individuals x such that $(x, y) \in R$ and $y \in C$ (e.g. $\exists\texttt{Parent.Female}$ is the set of people who have a daughter), and universal quantification, denoted $\forall R.C$, is the set of individuals x such that for all y, if $(x, y) \in R$, then $y \in C$ (e.g. $\forall\texttt{Parent.Female}$ is the set of people who only have daughters).

Whilst there are numerous useful description logics, there are benefits from using a standard description logic. The W3C organization has taken this approach with the OWL standard (www.w3c.org). This standard incorporates a small family of description logics, and has resulted in a number of tools for constructing, managing, and reasoning with ontologies, in particular the open-source tool Protege for editing and analysing OWL ontologies (protege.stanford.edu), and the Pellet reasoning system for querying OWL onotologies (pellet.owldl.com). Furthermore, use of OWL has meant that tractable languages can be used for knowledge representation and reasoning with ontologies the size of SNOMED (which has nearly 400,000 concepts and 1 million terms). Description logics in OWL have been used to develop new ontologies and to analyze and improve existing ontologies. For example, using description logic tools and automated reasoning systems, the Gene Ontology can be checked in order to reduce the occurrence of biologically inconsistent combinations of terms (Bada et al., 2004). Similarly, description logics can be used for managing and querying ontologies for a range of biomedical applications see (Rector, 2003).

The development of the theory and tools for description logics has been impressive over the past few years with significant progress in extensions to description logics to allow encodation of diverse types of knowledge and to address challenging tasks such as merging multiple ontologies (to create a single consistent and coherent ontology) and aligning ontologies (to allow mapping between concepts and statements in one ontology to those in another ontology (e.g. (Noy and Musen, 2000; de Bruijn et al., 2004; Choi et al., 2006; Meyer et al., 2006)).

In conclusion, we see that description logics have been shown to be useful for many different kinds of scientific knowledge and indeed it would seem that any domain in science has ontological knowledge that could be usefully encoded and used in the form of a description logic.

2.2 Logic programming

Logic programming is a declarative approach to programming. A set of formulae (clauses) constitutes the program, and inference is the program execution. Logic programming (in particular Prolog (Bratko, 2000)) can be used for rapid prototyping and for building knowledgebase systems. It can also be used for inferential databases (which extend relational databases with rules) and for constraint satisfaction applications and configuration applications (where constraints can be presented as logical statements).

Logic programming languages such as Prolog offer a simple and clear way of implementing inferential extensions of relational information. As instances of ontological knowledge are effectively ground monadic or binary predicates, these ground predicates can be used within a Prolog program to obtain

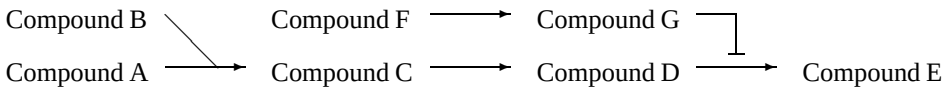

Figure 1 A graphical representation of a signal transduction pathway: Compounds A and B together are transformed into C, C is then transformed into D, and D is then transformed E. However, F is transformed into G and then G inhibits the transformation of D into E.

inferences. This can allow for context-sensitive inference of further facts concerning the instances in the ontological knowledge.

As an example of this, a mouse embryo anatomy has been encoded in Prolog (Burger et al., 2004). Using the "part-of" relation, much information about the mouse anatomy at a particular stage of development can be modeled. Furthermore, Prolog can be used to handle the mouse anatomy at different stages. Since the anatomy can change substantially through development, the anatomical nomenclature depends upon the context. For instance, at an earlier stage of embryo development *outflow tract* is a part of *primitive heart tube*, but at a later stage of embryo development, there is no *primitive heart tube*, and *outflow tract* is part of *heart*. Using Prolog rules, the nomenclature for mouse anatomy can be encoded so as to be conditional on the stage of development. Furthermore, links between mouse anatomy and gene data can be encoded so that the genes responsible for particular structures in the mouse anatomy can be inferred.

Logic programming languages also offer a simple and clear way of implementing inferential extensions of existing technology for managing and querying ontologies. For example, DR-Prolog is a system for defeasible reasoning with rules and ontologies on the Semantic Web (Antoniou and Bikakis, 2007). The system is syntactically compatible with RuleML, it features strict and defeasible rules with priorities, and it can reason with rules, RDF, RDF Schema, and (parts of) OWL ontologies. Further examples of Prolog systems that extend ontology systems include (Laera et al., 2004; Krtzsch et al., 2006).

Another application area for logic programming is in the representation of signal transduction pathways. Signal transduction occurs in a cell when it receives an appropriate stimulus and it involves a sequence of biochemical reactions each of which is instigated by an enzyme. The net result depends upon the type of the original stimulus and on the state of the cell. Types of biochemical compound involved include neurotransmitters and hormones. These pathways are often viewed in the form of graphs where each node is either a compound or an input or output of the pathway, and each arc denotes the transformation from one compound into another. More complex graphical notations have been used to incorporate further concepts such as the notion of inhibition: For example, some enzymes inhibit the process of one compound being transformed into another, as illustrated in Figure 1. Representing such a graphical structure is straightforward in Prolog: Arcs are represented by ground binary atoms and the axiom for transitivity is conditional on the non-existence of inhibition (using negation-as-failure).

In practice, a number of variants of this graphical notation have emerged that can enrich the logic programming representation. For instance, further types of node have been used including nodes that abbreviate graphs (i.e. a node can stand for a subgraph and so different granularities of the pathway can be captured), nodes that represent several compounds, and nodes that represent protein structures. Also, various types of arc have been introduced to provide details on the biochemistry of the transformation (e.g. bind, release, attach modifier, and detach modifier). Such graphical notations are amenable to logic-based representation as illustrated by a modelling of pathways involving TGF-β growth factors (Fukuda and Takagi, 2001).

A more detailed attempt at modelling biochemical pathways has been undertaken in a model of the control of metabolism in *E.Coli*. Again this involves modelling signal transduction, but it also involves modelling a wider range of phenomena in cell biochemistry. Given the complexity of the modelling task, the process model based on the BDI (Beliefs, Desires, and Intentions) approach is harnessed in a Prolog formalization (Jonker et al., 2002). This enables more details to be used about the states and the triggers from one state to another. In the following fragment for describing how a cell can import lactose, we see that the desire by the cell for growth, implies a desire by the cell for importing lactose and glucose into the

cell. This leads to intentions for performing the import. This is implied when there is no belief (denoted by neg as the second argument of the belief predicate) that there is glucose externally present and there is belief (denoted by pos as the second argument of the belief predicate) that there is lactose externally present. This intention then implies that the lactose import is performed, and this in turn leads to the world state being updated regarding the lactose now being internally present.

```
desire(grow).
desire(foodImport) :− desire(grow).
desire(lactoseImport) :− desire(foodImport).
intention(performLactoseImport) :− desire(lactoseImport),
                                   belief(lactoseExternallyPresent, pos),
                                   belief(glucoseExternallyPresent, neg).
performs(lactoseImport) :− intention(performLactoseImport).
worldstate(lactoseInternallyPresent, pos) :− performs(lactoseImport),
                                   worldstate(lactoseExternallyPresent, pos).
```

This model demonstrates the potential of a higher-level qualitative view of cell biochemistry compared to that obtained by detailed numerical calculations for the concentrations and timings of the substances in the processes (Jonker et al., 2002). The model involves over 700 possible settings for the outside conditions, including for availability of various concentrations of glucose, lactose, oxygen, phosphorus, etc. It was evaluated in terms of determining whether various parameters such as type of food used, and occurrence of anabolism, in the cell model were correctly identified given the settings for the outside conditions.

Formalizing notions such as "X triggers Y", "X causes Y", and "X inhibits Y" offers a more sophisticated approach to modelling cell biochemistry. In (Tran et al., 2005; Tran and Baral, 2009, 2007), the KR approach of action languages has been applied to modelling molecular interactions in cells. In this, the $\mathcal{A}_T^\infty$ action language (an extended version of the $\mathcal{A}_T^0$ action language) is used to give domain descriptions in a set of statements of the following form where f_i, g_i, h_i, f and g are fluents (i.e. propositions that change in truth value over time), and a, b, and c are propositions denoting actions.

(Causal rule)	a `causes` f `if` $f_1, ..., f_n$
(Trigger rule)	$g_1, ..., g_m$ `triggers` b
(Inhibition rule)	$h_1, ..., h_l$ `inhibits` c

These rules can be explained as follows: A causal rule says that f is true in the state succeeding a state where a occurs and all of f_1 to f_n are true; A trigger rule says that action b is to occur if it is not inhibited and if all the literals g_1 to g_m hold; and an inhibition rule says that action c cannot happen (i.e. it is inhibited) if all the literals h_1 to h_l hold. Then a sequence of states can be obtained where by each state is obtained from the previous state by applying these rules. Furthermore, this sequence of states can be obtained by a meta-level interpreter in Prolog (see for example (Kakas et al., 2000) developed for a related action language).

The $\mathcal{A}_T^\infty$ action language has been used to model the p53 protein which plays a role in the regulation of cell growth and cell death, and is associated with the suppression of tumours. Stimuli such as ultraviolet light (UV) and chemical carcinogens can induce DNA damage, which leads to genomic instability, which in turn triggers abnormal cell growth and thereby tumours. These stimuli also upregulate the gene expression of p53, and this results in a higher level of p53, and this p53 suppresses the abnormal cell growth, and hence prevents cancer. The following is a simple example of modelling the p53 protein.

high(UV), ¬unstable(cells) triggers damage(DNA)
damage(DNA) causes unstable(cells)
unstable(cells), ¬tumourous(cells) triggers proliferate(cells)
proliferate(cells) causes tumourous(cells)
high(UV), ¬high(p53) triggers upregulate(p53)
upregulate(p53) causes high(p53)
high(p53) inhibits proliferate(cells)

If only high(UV) is true at time 0, then actions damage(DNA) and upregulate(p53) will occur at time 0. Then unstable(cells) becomes true at time 1, which triggers proliferate(cells). However, the action fails to occur because it is inhibited by high(p53) which occurs at the same time.

Whilst there are a number of action languages that have been developed, with applications such as cognitive robotics, the $\mathcal{A}_T^\infty$ action language has been developed specifically for modelling biochemical processes. For this, the $\mathcal{A}_T^\infty$ action language supports modelling of actions that are not necessarily immediate, i.e. actions where there is some latitude as to exactly when the action takes effect. It also supports the modelling of systems at different levels of granularity. For example, the above example of the p53 protein is very high-level, and so the process behind the expression of the protein is just in terms of the high-level notion of upregulate(p53), whereas a more detailed model would need to explicitly consider the upregulation of mRNA, which in turn causes raised levels of the appropriate mRNA, which then leads to the raised levels of p53. The framework for $\mathcal{A}_T^\infty$ incorporates features for consistent behaviour being maintained between the higher-level abstract models and the lower-level detailed models.

In related work (Baral et al., 2004), a framework for signalling networks has been based on the use of causal rules, trigger rules, and inhibition rules. The framework includes a scripting language for representing knowledge and queries, and an AnsProlog program (which is an answer set programming approach) for encoding the knowledge and for computing answers for queries. The framework supports three kinds of reasoning: Prediction which involves determining whether a particular property is eventually true in some state given some starting state; Explanation which involves determining the causes for a particular property to be true in some state; and Planning which involves determining what interventions are required to ensure that a particular property will be true in some state.

The use of an action language by (Baral et al., 2004) has been extended to provide a richer set of constructs, together with a translation mechanism from the action language description of a biological network into answer set programming, thereby allowing for queries to be handled by existing implementations for answer set programming (Dworschak et al., 2008). The approach has been evaluated by modelling the sulphur-starvation response pathway of *arabidopsis thaliana*, a small flowering plant that is popular for bioscience research.

A common issue in modelling biochemical pathways is that the available scientific knowledge is incomplete. One approach to dealing with incompleteness is to use abduction: Given a knowledgebase that is insufficient to entail some formula of interest, abduction is the logical process of "guessing" further premises that can be added to the knowledgebase in order to entail the formula of interest. So in a pathway model, missing steps in the model can be guessed based on what is known, such as demonstrated for the aromatic amino acid synthesis pathway (King et al., 2004).

In (Tamaddoni-Nezhad et al., 2004, 2007), abduction has been used for identifying previously unknown metabolic steps in the pathways for metabolism of a toxin (hydrazine). Knowledge about enzymes is represented by ground atoms of the form reactionnode(metabolite1, enzyme, metabolite2) where metabolite1 is the "input" to the enzyme and metabolite2 is the "output", and observables are represented by ground atoms of the form concentration(metabolite, level, time) which denotes that at the time point, the concentration of a particular metabolite is given by level. The abducible predicates are of the form inhibited(enzyme, metabolite1, metabolite2, time) denoting the metabolism from metabolite1 to metabolite2 by enzyme is inhibited at the time point. In addition, some Prolog rules such as the following (where down is a constant to denote that the concentration level is reduced, and E,

T, X, and Y are variables) and integrity constraints provide relationships between these facts:

$$\texttt{concentration}(\texttt{Y}, \texttt{down}, \texttt{T}) :- \texttt{reactionnode}(\texttt{X}, \texttt{E}, \texttt{Y}), \texttt{inhibited}(\texttt{E}, \texttt{X}, \texttt{Y}, \texttt{T}).$$

Using this knowledge, together with some simplifying assumptions (such as the primary effect of the toxin can be localized on the individual reactions in the pathway, the underlying network defined by the reactionnode predicate is correct and complete, all the reactions are *a priori* equally likely to be affected by the toxin, and inhibition in one reaction is sufficient to cause change in the concentration of the metabolites), offers a useful method for predicting inhibition.

Abductive inference has also been used for dealing with incomplete knowledge in the construction of genetic networks from mutant data (Zupan et al., 2001, 2003). Genetic networks are acyclic graphs where each terminal node denotes a biological process, each non-terminal node denotes a gene, and each arc is labelled as either inhibits or exciteps. Such a graph denotes how a biological process can affect other biological processes via a sequence of gene products corresponding to the sequence of genes. By using data from experiments where one or more genes have been knocked out or overexpressed, the role of each gene in the network can be determined.

Another interesting application area for Prolog has been for representing relationships between the structure of a molecule and its effects (Muggleton, 1999), including predicting the pharmacological activity of a molecule based on its structure (Finn et al., 1998), and predicting mutagenicity of a chemical compound (i.e. the capacity of a compound to cause permanent alteration to the genetic material in a cell) (King et al., 1996). The following Prolog rule, taken from (Finn et al., 1998), states that molecule A can be predicted to be an inhibitor of the angiotensin-converting enzyme (ACE) if molecule A binds to zinc at site B and molecule A contains a hydrogen acceptor C and the distance between B and C is 7.9 Angstroms and molecule A contains hydrogen acceptor D and the distance between B and D is 8.5 Angstroms and the distance between C and D is 2.1 Angstroms and molecule A contains a hydrogen acceptor E and the distance between B and E is 4.9 Angstroms and the distance between C and E is 3.1 Angstroms and the distance between D and E is 3.8 Angstroms.

$$\begin{aligned}\texttt{inhibits}(\texttt{A}, \texttt{ACE}) :- \texttt{binds}(\texttt{A}, \texttt{zinc}, \texttt{B}),\\ \texttt{contains}(\texttt{A}, \texttt{C}), \texttt{hydrogenacceptor}(\texttt{C}),\\ \texttt{dist}(\texttt{B}, \texttt{C}, 7.9),\\ \texttt{contains}(\texttt{A}, \texttt{D}), \texttt{hydrogenacceptor}(\texttt{D}),\\ \texttt{dist}(\texttt{B}, \texttt{D}, 8.5), \texttt{dist}(\texttt{C}, \texttt{D}, 2.1),\\ \texttt{contains}(\texttt{A}, \texttt{E}), \texttt{hydrogenacceptor}(\texttt{E}),\\ \texttt{dist}(\texttt{B}, \texttt{E}, 4.9), \texttt{dist}(\texttt{C}, \texttt{E}, 3.1), \texttt{dist}(\texttt{D}, \texttt{E}, 3.8).\end{aligned}$$

Constraint logic programming (CLP) has also been applied to predicting the activity of a protein from its structure (i.e. the protein structure-activity problem) (Dal Palu et al., 2004). CLP is based on the idea of representing constraints using a logic programming notation, and then a solution is a set of inferences of a logic program that satisfies the constraints. Using the CLP approach, constraints on the relationship between primary, secondary and tertiary structure can be encoded in logic, and this is then used by the CLP program for predicting secondary or tertiary structure from the primary structure.

In conclusion, we see that logic programming has been shown to be useful for diverse kinds of representing and reasoning with scientific knowledge including: (1) Ontological knowledge; (2) Biochemical pathway knowledge including signal transduction and metabolic pathways; and (3) Protein structure-activity knowledge. Furthermore, it seems that by using logic programming technologies (including deduction with negation-as-failure, abduction, constraint logic programming, answer set programming, and action languages), a far wider range of scientific knowledge could be captured.

2.3 Argumentation systems

Argumentation systems aim to reflect how human argumentation uses conflicting information to construct and analyze arguments. So argumentation systems involve identifying arguments and counterarguments

relevant to an issue (e.g. What are the pros and cons for the safety of mobile phones for children?). Argumentation systems may also involve weighing, comparing, or evaluating arguments (e.g. What sense can we make of the arguments concerning mobile phones for children?) and they may involve drawing conclusions (e.g. A parent answering the question "Are mobile phones safe for my children?"). In addition, argumentation systems may involve convincing an audience (e.g. A politician making the case that mobile phones should be banned for children because the risk of radiation damage is too great).

There are a number of frameworks for modelling argumentation. They incorporate a formal representation of individual arguments and techniques for comparing conflicting arguments (for reviews see (Chesñevar et al., 2000; Prakken and Vreeswijk, 2002; Bench-Capon and Dunne, 2007; Besnard and Hunter, 2008)). These proposals allow for the representation of arguments for and against some claim, and for attack relationships between arguments. In a number of key examples of argumentation systems, an argument is a pair $\langle \Phi, \alpha \rangle$ where the first item Φ is a consistent set of formulae that proves the second item α which is a formula. The notions of "consistent" and "proves" depend on a base logic that is assumed as part of the definition of the argumentation system. Different base logics give us different systems for argumentation. Options include classical logic or some form of defeasible logic. Some argumentation systems also impose further constraints on the definition. For instance, a common constraint is that for $\langle \Phi, \alpha \rangle$ to be an argument, it is necessary that there is no subset of Φ that proves α.

In logic-based argumentation, a counterargument is an argument that in some sense "attacks" another argument. A key form of counterargument is an undercut: One argument undercuts another argument when the claim of the first argument negates the premises of the second argument. Another form of counterargument is a rebut: One argument rebuts another argument when the claim of the first argument negates the claim of the second argument.

An example of logic-based argumentation system is the framework (Besnard and Hunter, 2001, 2005) based on classical propositional and first-order logic (with the classical consequence relation $\vdash$) and in which an argument (obtained from a knowledgebase Δ) is a pair $\langle \Phi, \alpha \rangle$ where Φ is a minimal subset of Δ such that $\Phi \vdash \alpha$ and $\Phi \nvdash \bot$. There are also a number of proposals for argumentation systems based on defeasible logics. The common feature of defeasible logics is the incorporation of a defeasible implication into the language. This form of implication is weaker than classical logic and is usually only subject to *modus ponens* and not to *modus tollens*. Indeed, the language is normally restricted to a propositional language of literals and rules of the form $\alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta$, where $\alpha_1, \ldots, \alpha_n, \beta$ are literals, and a modus ponens rule of the form

$$\frac{\alpha_1, \ldots, \alpha_n, \alpha_1 \wedge \ldots \wedge \alpha_n \rightarrow \beta}{\beta}$$

With a defeasible logic as base logic, arguments can be defined as chains of reasons leading to a conclusion with consideration of potential counterarguments at each step. With the explicit structure in the chains of reasoning, diverse notions of defeat can be conceptualized. A number of these defeasible systems (e.g. (Dung et al., 2006)) construct arguments logically, and then evaluate sets of them as an abstract system (Dung, 1995) (which provides principled criteria for determining which arguments are acceptable in a set of arguments and counterarguments). In this way, a defeasible system can "instantiate" an abstract system, or equivalently, the abstract system provides a "semantics" for the defeasible system.

There are some implemented systems for argumentation for use with defeasible logic (e.g. DeLP (García and Simari, 2004), CASAPI (Gaertner and Toni, 2007), Argue tuProlog (Bryant et al., 2006), and Dungine (South et al., 2008)) and for classical logic (Efstathiou and Hunter, 2008). For a review, see (Bryant and Krause, 2008). Furthermore, there is increasing interest in standards (such AIF) for exchanging arguments between applications (Chesñevar et al., 2006) and for tool support for editing and analysing arguments (Rahwan and McBurney, 2007).

A domain that appears amenable to argumentative reasoning is toxicology. In this domain, there is much evidence from quite diverse studies that needs to be used together in order to determine whether a particular compound poses a toxicological risk, and there is significant uncertainty and inconsistency in the heterogeneous information available. There are various classifications of toxicological risk, such as the

following classification for risk of carcinogenicity by the US Environmental Protection Agency (Krause et al., 1998).

- **Known Human Carcinogen** iff there is *sufficient* evidence from human (epidemiological) studies.
- **Probable Human Carcinogen** iff there is *sufficient* animal evidence and evidence of human carcinogenicity, or *at least limited* evidence from human (epidemiological) studies.
- **Possible Human Carcinogen** iff there is *sufficient* animal evidence and but *inadequate* human evidence, or *limited* evidence from human (epidemiological) studies in the *absence of sufficient* animal evidence.
- **Not Classifiable** iff there is *inadequate* animal evidence and *inadequate* human evidence, but *sufficient* evidence of carcinogencity in experimental animals.
- **Non carcinogenic** to humans iff there is evidence for lack of carcinogenicity.

To investigate the possibility of using argumentative reasoning, the StAR Demonstrator was developed as a prototype computer-based assistant for the prediction of potential carcinogenic risk due to novel chemical compounds (Krause et al., 1993, 1995b, 1998). This system constructs arguments for and against ascribing various carcinogenic risk classifications on compounds. The system draws on heterogeneous data sources and gives the results using graphical and textual presentations. The argumentation is based on a form of defeasible reasoning. In addition, quantitative reasoning in the form of Dempster-Shafer theory is incorporated into the argumentation (Krause et al., 1995a).

The approach taken in the StAR Demonstrator was to build on an existing toxicology information system that identifies the key molecular substructures within a molecule, then with knowledge of the potential toxicity of the substructures, the toxicity of the molecule can be predicted. This prediction involves considering arguments and counterarguments based on the known toxicology studies involving those substructures. Since these toxicology studies involve diverse outcomes being measured (from mortality rate to changes in biochemical parameters in particular tissues) and use diverse *in vivo* or *in vitro* models (i.e. some involve human studies, some involve animal studies, and some involve laboratory-grown tissues). The argumentation system then identifies the pros and cons for a classification based on the classifications for the substructures available in the underlying information system.

In another analysis of argumentative reasoning for carcinogenics, a theoretical framework has been characterized for capturing modes of inference that are posited for dealing with a diverse range of heterogeneous sources for predicting risk (McBurney and Parsons, 2001). In order to infer that an effect of a compound in one context has the same effect in another depends on checking key questions (somewhat like the critical questions used for argument schemes (Walton, 2006)) such as “it is valid to consider the mechanism for the effect in one model exists in the other model”. Though these key questions have not been formalized in the proposal, it does raise some interesting issues for the further development of computational models of argument, in particular for how conflicting heterogeneous sources can be synthesized into a classification for risk of carcinogenics.

There has been some consideration of argumentation systems for representing and reasoning with expert knowledge from scientists. For instance, in a case study in using the BLAST bioinformatics tool, the expertise in determining whether two sequences match (such as amino acid sequences) can be partially captured in terms of a set of atomic arguments for and against the two sequences matching (Jefferys et al., 2006). For instance, an argument for the sequences matching may be that “over 80% of the sequences are the same” and an argument against may be that “the sequences are short (a chain of less than 20 amino acids)”. Such a set of such arguments and counterarguments can than be evaluated using Dung’s definition (Dung, 1995) for extensions to determine which arguments are accepted and which rejected, and hence to determine in terms of these arguments whether the two sequences match.

Another application domain that has proved promising for argumentation systems is for reasoning with knowledge in clinical pharmacology. In providing decision-support for clinicians, the CAPSULE system provides pros and cons for each drug for treating a problem in a particular patient (Fox and Das, 2000). The criteria for these arguments include whether the drug is contraindicated by other patient conditions, whether there are any interactions with other drugs the patient is taking, whether the drug is liked/disliked

by the patient, whether it follows local or national guidelines, whether the drug is expensive/inexpensive, etc. In addition to rules capturing these criteria, the system draws on a database of facts corresponding to the *British National Formulary* which tabulates information about each drug including the disorders it is intended for, and side-effects and contraindications. In trials, statistically significant improvements in decision making by clinicians were identified in terms of the quality of decision, the speed of decision-making, and of the cost of drugs chosen (Walton et al., 1997).

Another system to draw on knowledge of clinical pharmacology directly harnesses the results of published clinical trials. In this approach, called the Ontology-based Argumentation Framework (OAF), a logic-based argumentation formalism is coupled with a description logic ontology so that the predicates in the ontology correspond to terms in the medical literature, while arguments are generated by drawing defeasible inferences using defeasible rules (defined in terms of the outcomes of clinical trials) and facts represented in the onotology. In this work, it has been shown how these two formalisms can be tightly coupled by observing a few simple restrictions, and provides features not available in either formalism alone (Williams and Hunter, 2007).

The OAF approach was evaluated in a large case study on decision-making in treatment choice for breast cancer, where rules are developed from the results of published clinical trials. The focus of case study was on changes on *Overall Survival*, *Disease Free Survival*, or *Risk of other conditions* following from tamoxifen-based treatments for early stage breast cancer, and compared with a National Cancer Institute (NCI) Guideline (which provides authoritative advice based on the evidence gleaned from clinical trials). From the text in the guideline, there were 29 informal arguments on the focus, and from the 39 peer-reviewed clinical trials referenced in the bibliography on the focus, there were 68 defeasible rules that were extracted from the main statistically significant findings presented in the title or abstract. The guideline was used as "gold standard", and it was compared with the formal arguments generated by the OAF system. In this comparison, all the arguments concerning survival outcomes generated by the argument system were reflected in the guidelines. However, not all of the arguments in the guideline were produced by the argument system, for three reasons. The first is that the guideline fails to reference certain inferences that it makes (e.g. from increased disease specific survival to increased overall survival). The second is that the guideline is more relaxed about the patient characteristics than the trials explicitly justify (e.g. a trial on premenopausal women with early stage breast cancer is used as the reference for a statement about all women with early stage breast cancer). The third is that where a treatment has mixed effects (as most do) the argument system often fails to correctly identify them. The first two results are interesting, as they allow us to pinpoint exactly where the guideline fails to supply the evidence to support its statements, and thus we can use an argument system as a way of checking such guidelines. The third is important as it highlights a weakness in current formalizations of argumentation.

In conclusion, we see that argumentation systems have been shown to be useful for diverse kinds of representing and reasoning with scientific knowledge including: (1) Toxicology knowledge; (2) Knowledge about sequence analysis in genetics; and (3) Knowledge from clinical trials.

2.4 Uncertainty formalisms

An important feature of handling scientific knowledge is the need to represent uncertainty. Much leading-edge scientific information is subject to uncertainty in diverse ways, including empirical methods (such as the nature of populations and samples, estimates of experimental error, etc), statistical analyses (such as mean, standard deviation, probability statements, correlation, significance tests, etc), and subjective assessments drawn on the basis of the evidence.

This raises important and difficult questions in knowledge representation and reasoning. For empirical evidence used to support claims made in scientific papers, description of a dataset may include probability distributions over possible outcomes, belief distributions over a hypothesis set, and implicit use of possibility measures. Uncertainty can arise due to the accuracy problem of the method or equipment being used. Furthermore, given a dataset, different experiments may be conducted to analyze different aspects of the data. These descriptions may be inconsistent or conflict given the different ways that data are used,

or aggregated. Uncertainty in datasets can also arise from difficulties of categorizing data analysis results according to pre-defined classifications, where the boundaries between classes are not distinct.

A key requirement for representing and reasoning with uncertain scientific knowledge is that a "black box" approach to should be avoided. This is important for science applications where a scientist is not only interested in the answer to a question but also requires an explanation for the answer in terms of the relevant scientific knowledge.

Whilst statistical techniques are critically important, in general, for relating scientific observations to scientific hypotheses, and thereby to scientific knowledge, it is clear that statistical techniques do not directly provide means for reasoning with that knowledge. However, there is a wide range of uncertainty formalisms that could potentially be harnessed for representing and reasoning with scientific knowledge including those based on probability theory, possibility theory, Dempster-Shafer theory, fuzzy set theory, rough sets, and various kinds of logic. See (Krause and Clark, 1993; Parsons and Hunter, 1998) for a review of uncertainty formalisms.

Knowledge qualified by probabilistic information is particularly important in science, not least because much evidence for scientific theories is of a probabilistic or statistical form. One of the key drawbacks of using probabilistic information in reasoning is the need for a full probability distribution.

In order to circumvent the need for a full distribution, Bayesian networks (also called probabilistic networks or causal networks) are an approach to using conditional probabilities that use extra structural information relating the random variables to derive independence assumptions. These assumptions allow for a potentially substantial simplification of the conditional probabilities required (Pearl, 1988; Lauritzen and Siegelhalter, 1988; Heckerman and Wellman, 1995; Jensen, 1996). Probabilistic networks can be viewed as a set of nodes with directed arcs (arrows) providing connections between nodes. Furthermore, the resulting graph is restricted to being a connected acyclic graph. For example, for the random variables `smokes`, `disease`, and `symptom`, the following is a Bayesian network.

$$\texttt{smokes} \rightarrow \texttt{disease} \rightarrow \texttt{symptom}$$

From this network, we can obtain the following by assuming `symptoms` is independent of `smokes` conditional on knowing `disease` is true, which is much simpler than the full distribution for these three random variables.

$$\begin{aligned} &P(\texttt{smokes} \wedge \texttt{disease} \wedge \texttt{symptom}) \\ &\quad = P(\texttt{symptom} \mid \texttt{disease}) P(\texttt{disease} \mid \texttt{smokes}) P(\texttt{smokes}) \end{aligned}$$

There is a range of toolkits available for building and managing Bayesian networks (e.g. (Andersen et al., 1989)), and there are numerous applications including decision-support systems for diagnosis (e.g. (Andreassen et al., 1987)). In addition, variants (such as for qualitative probabilistic networks (Parsons, 1998)) are potentially valuable for capturing scientific knowledge.

However, for scientific knowledge, there appears to be a need combining probabilistic reasoning with logical reasoning. There have been a number of proposals for combining probability theory and logic resulting in a range of probabilistic logics (e.g. (Nilsson, 1986; Bacchus, 1990; Halpern, 1990). To illustrate some of the issues arising in probabilistic logics, consider the following two statements. The first is a probabilistic belief about a specified individual, whereas the second is a statistical assertion concerning a set of individuals.

- The probability that John (a particular person) does not have prostate cancer is greater than 0.9.
- The probability that a randomly chosen male does not have prostate cancer is greater than 0.9.

These two kinds of statement require different approaches to their semantics and proof theory. The first statement can be captured by a possible worlds approach where each world is a classical model representing a possibility and there is a probability distribution over the worlds. If the proportion of worlds in the possible worlds interpretation, where $\texttt{notProstateCancer}(\texttt{John})$ is true, is greater than 0.9, then $\texttt{P}(\texttt{notProstateCancer}(\texttt{John})) > 0.9$ is true. In contrast, the second statement is a chance set-up. It is

the result of doing some experiment or trial, and so it is a statistical statement. For this, the possible worlds approach is not adequate, since in each possible world there may be a patient who does have prostate cancer. A better alternative is to use a single classical world plus a probability distribution over patients, and then extend classical predicate logic with the new syntax for formulae,

$$P_{\mathtt{X}}(\mathtt{notProstateCancer}(\mathtt{X}) \mid \mathtt{Male}(\mathtt{X})) > 0.9$$

where X is a randomly chosen element of the domain, and the formula is a statement that is either true or false in this extended classical model. This kind of formula can then be used in more complicated classical formulae using the usual logical symbols.

A key shortcoming of extending classical logic in order to handle probabilistic or statistical information, either by using a possible worlds approach or by adding a probability distribution to each model, is the computational complexity that it involves. An alternative is to extend a more computationally viable approach, namely logic programming, with a representation of probabilistic information, giving rise to the idea of probabilistic logic programming.

In probabilistic logic programming, each logic programming clause is annotated with probabilistic information, such as a probability interval, and this information is then used to annotate the inferences of the program with probabilistic information. For example, in (Lukasiewicz, 1998), each program clause is annotated with subintervals of $[0, 1]$ that describes the range for the conditional probability of the head of a clause given its body. In the following clause, the conditional probability of not having prostate cancer for a randomly chosen individual given that the individual is male is in the interval $[0.9, 1]$.

$$(\mathtt{notProstateCancer}(\mathtt{X}) \mid \mathtt{Male}(\mathtt{X}))\ [0.9, 1]$$

As an illustration of a potential application of using this formalism with scientific knowledge, consider the following clause that captures part of the meta-analysis published by the Lancet Earlier Breast Cancer Trialists Collaborative Group (EBCTCG) for local therapy comparisons [1]. This clause (a conditional event) says that the probability of a woman's 10-year mortality, under the condition that she has had breast cancer, she is ER-poor, and she has had breast conservation surgery (BCS) plus radiotherapy (RT) treatment, is 22.3%.

$$(\mathtt{Mortality}(\mathtt{X}, \mathtt{Y10}) \mid \mathtt{BreastCancer}(\mathtt{X}) \wedge \mathtt{ER}(\mathtt{X}, \mathtt{poor}) \wedge \mathtt{BCS}(\mathtt{X}, \mathtt{Y1}) \wedge \mathtt{RT}(\mathtt{X}, \mathtt{Y1}))[0.223, 0.223]$$

The usefulness of probabilistic logic programs to represent imprecise probabilistic knowledge and harness this knowledge to answer queries can further be demonstrated by an example from biochemistry on the human enzyme galactokinase, which uses galactose as a substrate. Galactose has the molecular formula $C_6H_{12}O_6$, but other compounds have the same or similar formula. Since not all possible substrates for the enzyme have been tested, the information regarding this enzyme and its substrates is incomplete, then the question is: can we predict which will be the substrates for the human enzyme galactokinase based on incomplete and imperfect information? Many factors lead to the information being imperfect including different research laboratories using different criteria for scoring a compound as a substrate and some information is based on galactokinases from other species, so we cannot be certain that substrate specificity is conserved for humans.

Each galactose molecule is arranged as a hexagonal ring (e.g., the α-D-Galactose molecule in Figure 2). There are six carbon atoms in a galactose molecule and one oxygen atom. These six carbon atoms are numbered from 1 to 6. The oxygen atom is not numbered. The other atoms can be regarded as coming off these carbon atoms. The first four of the carbon atoms each has an OH molecule attached to it, and the fifth one has the sixth carbon atom attached to it from outside the ring, forming a CH_2OH group. The OH group can either be "up" or "down" (i.e. they are chiral). The combination of ups and downs gives a specific form of the molecule (in effect, each form of the molecule is a different compound), and the actual combination can significantly affect the biochemical behavior of the molecule. Therefore, for the OH groups attached to these atoms, we need to know if they are *up*, *down* or *absent*. The sixth carbon is

[1] http://www.ctsu.ox.ac.uk/projects/ebctcg

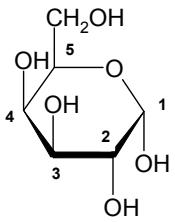

Figure 2 The α-D-Galactose molecule.

not chiral, and so the OH is neither up nor down. Hence, the OH for the sixth carbon is marked as either *present* or *absent*. Current experimental results published in the literature provide a set of conditional events with probabilities suggesting how likely a particular structure is a substrate for the enzyme. Based on the probabilistic knowledge in the probabilistic logic program, we can predict the probability for any combination of these six carbons.

For instance, we have the following probabilistic conditional event in the probabilistic logic program containing all the probabilistic knowledge, where $\mathtt{sub}(\mathtt{X})$ denotes that $\mathtt{X}$ is a substrate, $\mathtt{c2}(\mathtt{X},\mathtt{u})$ denotes the OH group of carbon number 2 is $\mathtt{up}$ etc. This probabilistic knowledge can then be used to query any possible structure like $\mathtt{c2}(\mathtt{X},\mathtt{u}) \wedge \mathtt{c3}(\mathtt{X},\mathtt{d}) \wedge \mathtt{c4}(\mathtt{X},\mathtt{u}) \wedge \mathtt{c6}(\mathtt{x},\mathtt{p})$ and obtain the conditional probability that it is a substrate for the enzyme.

$$(\mathtt{sub}(\mathtt{X}) \mid \mathtt{c2}(\mathtt{X},\mathtt{u}) \wedge \mathtt{c3}(\mathtt{X},\mathtt{d}) \wedge \mathtt{c4}(\mathtt{X},\mathtt{d}) \wedge \mathtt{c6}(\mathtt{x},\mathtt{p}))[0.6, 0.9]$$

Probabilistic logic programming has been used to represent and reason with probabilistic knowledge in many real-world applications, (e.g., (Fuhr, 2000; Raedt et al., 2007; Baral and Hunsaker, 2007)). Furthermore, it seems a very promising approach for representing and reasoning with a wide variety of uncertain scientific knowledge.

In conclusion, scientific knowledge is subject to diverse kinds of uncertainty. Bayesian networks have already been shown to be important in many areas for specific kinds of reasoning tasks (such as diagnosis) and probabilistic logic programming promises to be useful in many applications for richer knowledge representation and reasoning. Nonetheless, there appears to remain many further opportunities for developing and applying uncertainty formalisms for representing and reasoning with knowledge specific domains in science.

3 Systems for combining scientific knowledge

So far in this review, we have considered various ways that scientific knowledge can be represented, and reasoned with, in a declarative formalism. However, repositories of scientific knowledge in a formalism that can be subject to computational querying and analysis are difficult and expensive to obtain. In part this is because the complexity of pulling together knowledge for encodation in a structured format from multiple sources rises rapidly with the size of the knowledgebase, and the automated techniques for supporting this activity are under-developed. There is therefore a pressing need to develop technology for combining scientific knowledge where the information or knowledge comes from multiple sources.

In order to combine heterogeneous scientific knowledge, we need to take account of the contents of each piece of knowledge. Different kinds of content, and different types of conflict (whether qualitative or quantitative) need to be combined in different ways. This is because a source may provide information that *conflicts* with, rather than *complements* that provided by another source; that is, instead of providing additional information about different events or phenomena, a source may provide different information about the same events or phenomena.

Scientific knowledge can be combined in a range of ways. We can keep separate knowledgebases for each source, and use technology for answering queries in the different knowledgebases. This approach is a form of information integration or a form of distributed system. Alternatively, we can aggregate the

knowledge to form a single knowledgebase. So the input would be a tuple of knowledgebases, and the output would be a knowledgebase. An important form of aggregation is knowledge (or belief) merging which results in a consistent knowledgebase as output. The need for combining scientific knowledge is not just about using resources from multiple sources to answer queries, but it may be an end in itself. In other words, a knowledgebase aggregated from multiple sources may provide a useful summary or review of some field. We consider this range of approaches in the rest of this section.

There is much interest in information integration in areas such as semantic web technology. However, the emphasis in information integration is at a different level to that of knowledge aggregation. Information integration is about pulling diverse bits of data (e.g. individual names, terms, numbers, etc) together in response to queries. For example, the Gene Ontology (GO) data source has been used in the TAMBIS (Transparent Access to Multiple Bioinformatics Information Sources) project to provide a single, uniform query interface to diverse bioinformatics sources (Baker et al., 1998; Stevens et al., 2001a,b).

In contrast to integration, knowledge aggregation is about forming richer theories, in a logic, on a focused topic, and as such is a more difficult problem than information integration. The most developed approach to knowledge aggregation is that of knowledge (or belief) merging. Logic-based knowledge merging takes a tuple of knowledgebases as input and produces a consistent knowledgebase as the output that preserves as much as possible of the input knowledge (Baral et al., 1991; Konieczny et al., 2004). So knowledge merging looses information in order to provide an acceptable consistent view of the conflicting input knowledge. The approaches to knowledge merging are either syntactic or semantic.

In syntactic approaches to knowledge merging, the maximally consistent subsets of the union of the input knowledgebases are considered as candidates for the output (Baral et al., 1991). In contrast, in the semantic approaches to knowledge merging, the models of the input knowledgebases and the models of candidate output knowledgebases are considered so that an acceptable compromise amongst the input knowledgebases is found (Konieczny et al., 2004). For example, the potential output knowledgebase that has models that are nearest to the models of the majority of the input knowledgebases is selected. Semantic approaches have a number of desirable properties that the syntactic approaches lack. However, the semantic approaches are restricted to propositional logic. There are also algorithms, and a prototype implementations based on BDD technology, for the semantic approaches (Gorogiannis and Hunter, 2008a).

Recently, a generalization of the semantic approaches has been proposed that is based on a notion of dilation, a way of weakening classes of models, and thereby allows merging of knowledgebases of first-order predicate logic (Gorogiannis and Hunter, 2008b). In addition, this generalization maintains many of theoretical advantages of the semantic approaches, for knowledge merging.

Unfortunately, the knowledge merging approaches suppress conflict, and if not enough is known about the conflict, it may result in arbitrary loss of knowledge. For example, suppose one clinical trial states that treatment T_1 is better than treatment T_2, denoted $T_1 > T_2$, with another clinical trial states that $T_2 > T_1$, and there is an integrity constraints that says $\neg((T_1 > T_2) \wedge (T_2 > T_1))$. This is inconsistent, but merging would just result in a disjunction (i.e. $((T_1 > T_2) \vee (T_2 > T_1))$ which is a tautology. In contrast, it would be better for medical professionals to know that there is a conflict concerning these treatments.

Ontology alignment and ontology merging is an increasingly important issue for information integration (Noy and Musen, 2000; de Bruijn et al., 2004; Choi et al., 2006; Meyer et al., 2006). Some of these offer techniques for combining information in the form of description logic knowledge from multiple sources to be combined into a single description logic ontology. Even with the restriction to description logic, this is a challenging problem. Unfortunately, these techniques do not offer solutions for knowledge that is not in the form of description logic, and they do not support analysis of conflicting perspectives. Though argumentation has been applied to ontology alignment, with the argumentation used to choose an alignment rather than to qualify reasoning with the ontology (Laera et al., 2007).

Other logic-based approaches to combining knowledge include the KRAFT system and the use of Belnap's four-valued logic (Belnap, 1977). The KRAFT system uses constraints to check whether information from heterogeneous sources can be merged (Preece et al., 1999; Hui and Gray, 2000). If knowledge satisfies the constraints, then the knowledge can be used. Failure to satisfy a constraint can

be viewed as an inconsistency, but there are no actions on inconsistency. In contrast, Belnap's four-valued logic uses the values "true", "false", "unknown" and "inconsistent" to label logical combinations of information (see for example (Loyer et al., 2000)), and so conflicting information can be combined by assigning the "inconsistent" truth value accordingly. However, this approach does not provide actions in case of inconsistency.

Merging information is also an important topic in database systems. A number of proposals have been made for approaches based on schema integration (e.g. (Poulovassilis and McBrien, 1998)), the use of global schema (e.g. (Grahne and Mendelzon, 1999)), conceptual modelling for information integration based on description logics (Calvanese et al., 1998a,b; Franconi and Sattler, 1999; Bergamaschi et al., 2001), and querying and repairs for inconsistent databases (Bertossi and Chomicki, 2003). Heterogeneous and federated database systems are relevant, but they do not identify and act on inconsistency in a context-sensitive way (Cholvy and Moral, 2001), though there is increasing interest in bringing domain knowledge into the process (e.g. (Cholvy, 1998; Calvanese et al., 2008)). However, all these approaches are obviously restricted to relational data which in general are insufficiently expressive to capture scientific knowledge.

An approach for combining semi-structured data is that of fusion rules for certain kinds of XML document called structured reports (essentially a restriction of text entries to being individual words or simple phrases, such as names and domain-specific terminology, dates, numbers and units). The antecedent of each fusion rule specifies the types of input for which it will fire, and the consequent specifies an action to be undertaken in constructing the output report (Hunter and Summerton, 2006). This provides a context-dependent form of combining, and so a set of fusion rules needs to be developed for an application. The approach includes mechanisms for representing and aggregating heterogeneous uncertainty based on probability values, degrees of beliefs, or necessity measures (Hunter and Liu, 2006). As with the approaches for relational databases, this approach is restricted to knowledge that can be captured, where in this case, knowledge in the form of semi-structured data).

Whilst we see that there are potential solutions to merging some kinds of qualitative and quantitative knowledge, there is also the need to develop techniques for dealing with statistical knowledge. For instance, results from clinical trials are usually summarized in the form of sampling distributions. When full information (mean, SEM) about these distributions is given, performing meta-analyses is straightforward. However, when some of the sampling distributions only have mean values, a challenging issue is to decide how to use such distributions in meta-analysis. Currently, the most common approaches are either ignoring such trials or for each trial with a missing SEM, finding a similar trial and taking its SEM value as the missing SEM. Both approaches have drawbacks. As an alternative, (Ma et al., 2007, 2008) present two new methods for estimating any missing SEMs from a set of sampling distributions with full information, together with a merging method handling clinical trials with partial information to simulate meta-analysis.

More generally, there is a need for further techniques for aggregating a wider range of uncertain and inconsistent knowledge as arising in scientific domains. For instance, there is the need for inconsistencies that arise between the sources to not be arbitrarily suppressed, but rather the aggregated knowledge should reflect some of the key conflicts (such as points of view) between the sources if there is not a strong reason to resolve the conflict in a particular way. Furthermore, there is the need to develop better techniques for harnessing the uncertainty in the knowledge and about the knowledge, and using this for better integration and aggregation that reflects the uncertainty in the sources.

4 Constructing scientific knowledgebases

Whilst the emphasis of this review is on representing and reasoning technology for scientific knowledge, there is the important issue of how scientific knowledgebases can be generated. Here we briefly consider harnessing natural language technologies and machine learning technologies.

4.1 Harnessing natural language technologies

Information extraction (IE) technology (or synonymously text mining technology) aims to "read" text and pick out the bits of information that are needed. IE systems tend to be developed for focused applications

where there is some regularity in the information being presented in the text. For example, in papers on clinical trials, there is some regularity in the information being presented: Such a paper is quite likely to include the patient class of the trial, treatment classes to which the patients were assigned, and the comparative outcomes of treatments. Hence, with an information extraction system for an application, there is the idea of a template that specifies the information that is sought by the system.

Key tasks in IE include term recognition (which involves identifying the key names, actions, etc, used the text), co-reference identification (which involves associating different occurrences of the same name or action), and relationship identification (which involves identifying the key relationships with their arguments in the text). For example, consider the following abstract [2], from which we may extract the drug names, *latanoprost*, *bimatoprost*, and *travoprost*, the patient group *newly diagnosed open-angle glaucoma patients*, the study period *8 week*, the study type *randomized, parallel group, masked evaluator study*, the treatment protocol *once daily administration*, and outcome measured *IOP reduction*, and the conclusion *no statistically significant difference*. These are some of the terms that need to be recognized. As an illustration of co-reference identification, in the conclusions section, *these drugs* needs to be associated with the drugs *latanoprost*, *bimatoprost*, and *travoprost*, listed on the first line. Finally, relationship identification involves associating each pair of drugs in the relationship of *no statistical difference* holding.

> PURPOSE: To evaluate the efficacy of latanoprost, bimatoprost, and travoprost given in the evening over the 24-hour curve in newly diagnosed open-angle glaucoma patients. METHODS: This 8-week, randomized, parallel group, masked evaluator study compared the efficacy of once daily administration ..... CONCLUSIONS: ..... the IOP reduction of these drugs is indistinguishable within statistical parameters.

It is worth adding that the MeSH headings for this paper under PubMed include the listing of the drugs involved, the general class of study, and some features of the patient class. This information can be harnessed by an IE application for cross-checking and thereby improving system correctness. However, in general, information extraction will, in addition, require specialized lexicons and rules or heuristics for dealing with variants of entries in the lexicons, and for the richer information in the text that is not captured in the MeSH headings.

A number of viable information extraction systems have been developed (Cowie and Lehnert, 1996). For example, the GATE System provides an implemented architecture for managing textual data storage and exchange, visualization of textual data structures, and plug-in modularity of text processing components (Cunningham et al., 2002). The text processing components include LaSIE which performs information extraction tasks including named entity recognition, coreference resolution, template element filling, and scenario template filling. Furthermore, a number of natural language parsers have been developed that can be incorporated in information extraction systems (for a comparison for biological applications see (Clegg and Shepherd, 2007)).

Statistical techniques are also increasingly common in information extraction systems (Manning and Schutze, 2000). For instance, tagging words in a sentence according to the part of speech. This helps in resolving many ambiguities that arise in parsing since many words having multiple roles. An important advantage of statistical techniques is the increased robustness of the resulting system and the vast amounts of data available that can be harnessed for building systems. Analyzing large corpora is recognized as critically important given that what might be believed is a well known word may occur only once in say ten thousand articles.

Information extraction is recognized as an important technology for fields such as bioinfomatics and medical information since it is not viable to have scientific articles published in a completely codified format. In (Hahn et al., 2007), it is argued that scientific terminologies are too large and complex for them to be used uniformly by everyone, and they are incomplete, not least because scientific discovery often involves identifying and describing new concepts. Furthermore, the meaning of an article is much more than just a set of terms. With the grammar and underlying semantics of the text, much more sophisticated

[2] From N Yildirim, A Sahin, and S Gultekin (2008) The effect of latanoprost, bimatoprost, and travoprost on circadian variation of intraocular pressure in patients with open-angle glaucoma, J. Glaucoma 17(1):36-9.

knowledge is represented in a scientific article. And since free text in articles is unavoidable, information extraction technology is necessary for drawing out important aspects of the knowledge in articles.

An important application area for IE is in the extraction of information about molecular pathways (e.g. (Friedman et al., 2001; Oda et al., 2008)). Here, the aim is to identify entities such as substrates and enzymes, and related terms such as conditions for reactions, and to identify relationships such as "enzyme E metabolizes X into Y". Even within what may seem like a focused domain, there are often numerous ways to write an entity, and there are many synonyms, more general, and more specialized names for relationships.

As with most applications of information extraction, the process is not error-free, but the empirical results are promising, and suggest that it will offer a viable technology for generating scientific knowledgebases in focused domains.

4.2 Harnessing machine learning technologies

Machine learning technologies offer a wide range of approaches to learning knowledge from data. Furthermore, there have been numerous applications of these approaches to applications in science. One approach that is particularly relevant to the theme of this review is that of inductive logic programming (ILP) because the output knowledge is in the form of a logic program: The input to the learning is a set of examples represented by facts in a logic program (E), and background knowledge represented by a set of clauses in a logic program (B), and the ILP system produces a hypothesis (H) which explains E in terms of B (Muggleton, 1991). In other words, the ILP system generalizes the set of examples E to give H such that the union of H and B implies each example in E.

ILP systems have been successfully applied to various problems in learning scientific knowledge (Muggleton, 1999) including predicting the pharmacological activity of a molecule based on its structure, predicting mutagenicity of a chemical compound (i.e. the capacity of a compound to cause permanent alteration of the genetic material in a cell), predicting carcinogenicity of a chemical compound, and predicting protein secondary structure (i.e. the main 3D substructures of the protein) from the primary structure (i.e. the sequence of amino acid residues in the protein).

Given the wide ranging role that logic programming can take on in representing and reasoning with scientific knowledge, as discussed in Section 2.2, it would appear that ILP offers an important approach to generating scientific knowledgebases for applications based on logic programming.

5 Discussion

In this review, we have considered five areas of research (namely description logics, logic programming, argumentation systems, uncertainty formalisms, systems for combining knowledge) in the field of knowledge representation and reasoning, and shown how in each of them there are some promising applications for analyzing scientific knowledge. From this review, we draw the following conclusions concerning the use of KR formalisms for scientific knowledge:

1. Description logics appear to offer a valuable and viable approach to capturing ontological knowledge in any area of science.
2. Logic programming formalisms, including action languages, have been the basis of specialized calculi developed for scientific domains (such as in biochemistry) where there are substantial bodies of knowledge in a regular form. Furthermore, it seems that there are many further scientific domains where further specialized calculi could be developed in a similar way.
3. Argumentation systems are a promising approach to reasoning with conflicting or inconsistent scientific knowledge. They have been shown to be useful in scientific domains (including toxicology and clinical science), and it seems likely that there are many further scientific domains where the incomplete and inconsistent nature of the knowledge means that the identification and evaluation of arguments and counterarguments will be useful.
4. Uncertainty formalisms are a promising approach to reasoning with uncertain scientific knowledge. Some formalisms such as Bayesian networks have been shown to be useful in diverse scientific

domains for diagnostic and classification tasks. However, there appear to be many further proposals for representing and reasoning with uncertainty that could be harnessed for dealing with scientific knowledge (such as probabilistic logic programming).

5. Systems for combining knowledge from multiple sources are recognized as an important need in general. Furthermore, developing such systems for dealing with scientific knowledge seems to be a good domain in which to develop theoretical solutions and tools for combining knowledge in general.

From these observations, it is clear that more research is needed in developing general principles and techniques for representation, inference, and aggregation for scientific knowledge, and more consideration needs to be given to identifying scientific domains to which this technology can be applied. Furthermore, there remain many important and interesting research questions in developing calculi for capturing specific kinds of scientific knowledge, and tools for querying, and combining, that knowledge.

Much of this review has been focused on the object-level scientific knowledge. But an important and under-developed area is that of techniques for harnessing meta-knowledge. Whenever we have knowledge, we have meta-knowledge (which is knowledge about that knowledge). A key requirement of reasoning with scientific knowledge is to provide meta-knowledge on the provenance and quality of the scientific knowledge. Scientists do not just want answers, they want justifications for those answers. They want to see where the original information comes from, how it was formalized, and what conflicts and uncertainties were flagged. This meta-knowledge makes the results of reasoning auditable.

Normally a scientific article contains diverse kinds of meta-level information. For instance, scientific papers often highlight the goal and contribution of the paper, and explain how other papers fail to address that goal or provide that contribution. A scientific paper is likely to include basic assumptions that are regarded as uncontroversial. In addition, a paper may present arguments, or opinions, or beliefs, (as premises for the work in the paper or as a result of the findings in the paper) and explain how they are similar or different to other arguments, or opinions, or beliefs, in other papers. In an approach for text summarization (Teufel and Moens, 2000, 2002), a methodology has been presented for tagging fragments of a paper according to the rhetorical status and relatedness of each of these fragments based on the phraseology used in these fragments. This means that the goal and contribution of the paper, and comparisons with other papers can be identified and tagged. Moreover, it indicates a step towards formalizing the rhetorical status and relatedness of a paper in a meta-level logic, and as such raises interesting issues for argumentation systems.

Drawing out meta-level information from scientific papers has also been considered in graphical notations such as Claimfinder (Buckingham Shum, 2007) which support scientists in analyzing free text arguments obtained from scientific papers by allowing for the relationships to be flagged between extracts from papers such as *supports* and *contradicts*. Again, this approach seems to suggest a formalization for reasoning about arguments from scientific papers in a meta-level logic, which raises further interesting research questions.

References

S. Andersen, K. Olesen, F. V. Jensen, and F. Jensen. Hugin – a shell for building bayesian belief universes for expert systems. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 1080–1085. Morgan Kaufmann, 1989.

S. Andreassen, M. Woldbye, B. Falck, and S. Andersen. MUNIN – a causal probabilistic network for interpretation of electromyographic findings. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI'87)*, pages 366–372, 1987.

G. Antoniou and A. Bikakis. DR-prolog: A system for defeasible reasoning with rules and ontologies on the semantic web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):233–246, 2007.

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

F. Bacchus. *Representing and Reasoning with Probabilisitic Knowledge: A Logical Approach to Probabilities*. MIT Press, 1990.

M. Bada, D. Turi, R. McEntire, and R. Stevens. Using reasoning to guide annotation with gene ontology terms in goat. *SIGMOD Record*, 33(2):27–32, 2004.

P. G. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. Tambis: Transparent access to multiple bioinformatics information sources. an overview. In *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology (ISMB'98)*, pages 25–34. AAAI Press, 1998.

C. Baral and M. Hunsaker. Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and non-naive conditioning. In *Proc. of 20th International Joint Conference on Artificial Intelligence (IJCAI07)*, pages 243–249, 2007.

C. Baral, S. Kraus, and J. Minker. Combining multiple knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 3:208–220, 1991.

C. Baral, K. Chancellor, N. Tran, N. Tran, A. Joy, and M. Berens. A knowledge based approach for representing and reasoning about signaling networks. *Bioinformatics*, 20:i15–i20, 2004. Supplement 1.

N. Belnap. A useful four-valued logic. In G. Epstein, editor, *Modern Uses of Multiple-valued Logic*, pages 8–37. Reidel, 1977.

T. Bench-Capon and P. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171:619–641, 2007.

S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration of heterogeneous information sources. *Data and Knowledge Engineering*, 36:215–249, 2001.

L. Bertossi and J. Chomicki. Query answering in inconsistent databases. In G. S. J. Chomicki and R. van der Meyden, editors, *Logics for Emerging Applications of Databases*. Springer, 2003.

P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128: 203–235, 2001.

P. Besnard and A. Hunter. Practical first-order argumentation. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI 2005)*, pages 590–595. MIT Press, 2005.

P. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.

I. Bratko. *Prolog Programming for Artificial Intelligence*. Addison Wesley, 2000.

D. Bryant and P. Krause. A review of current defeasible reasoning implementations. *Knowledge Engineering Review*, 23(2):1–34, 2008.

D. Bryant, P. Krause, and G. Vreeswijk. Argue tuprolog: A lightweight argumentation engine for agent applications. In *Computational Models of Argument (Comma'06)*, page 2732. IOS Press, 2006.

S. Buckingham Shum. Modelling naturalistic argumentation in research literatures: Representation and interaction design issues. *Intelligent Systems*, 22(1):17–47, 2007. Special Issue on Computational Modelling of Naturalistic Argumentation.

A. Burger, D. Davidson, and R. Baldock. Formalization of mouse embryo anatomy. *Bioinformatics*, 20 (2):259–267, 2004.

D. Calvanese, G. D. Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Source integration in data warehousing. In *Proceedings of the 9th International Workshop on Database and Expert Systems (DEXA'98)*, pages 192–197. IEEE Computer Society Press, 1998a.

D. Calvanese, G. D. Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *Proceedings of the 6th Conference on the Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13. Morgan Kaufmann, 1998b.

D. Calvanese, G. G. G, D. Lembo, M. Lenzerin, and R. Rosati. Inconsistency tolerance in P2P data integration: An epistemic logic approach. *Information Systems*, 33(4-5):360–384, 2008.

C. Chesñevar, A. Maguitman, and R. Loui. Logical models of argument. *ACM Computing Surveys*, 32: 337–383, 2000.

C. Chesñevar, J. McGinnis, S. Modgil, I. Rahwan, C. Reed, G. Simari, M. South, G. Vreeswijk, and S. Willmott. Towards an argument interchange format. *The Knowledge Engineering Review*, 21(4): 293–316, 2006.

N. Choi, I. Song, and H. Han. A survey on ontology mapping. *Sigmod Record*, 35(3):34–41, 2006.

L. Cholvy. Reasoning with data provided by federated databases. *Journal of Intelligent Information Systems*, 10:49–80, 1998.

L. Cholvy and S. Moral. Merging databases: Problems and examples. *International Journal of Intelligent Systems*, 10:1193–1221, 2001.

A. Clegg and A. Shepherd. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics 8:24*, 8:24, 2007.

J. Cowie and W. Lehnert. Information extraction. *Communications of ACM*, 39:81–91, 1996.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

A. Dal Palu, A. Dovier, and F. Fogolari. Constraint logic programming approach to protein structure prediction. *BMC Bioinformatics*, 5:186, 2004.

J. de Bruijn, F. Martin-Recuerda, D. Manov, and M. Ehrig. State-of-the-art survey on ontology merging and aligning v1. Technical report, SEKT: Semantically Enabled Knowledge Technologies, 2004. EU-IST Integrated Project (IP) IST-2003-506826.

P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

P. Dung, R. Kowalski, and F. Toni. Dialectical proof procedures for assumption-based admissible argumentation. *Artificial Intelligence*, 170:114–159, 2006.

S. Dworschak, S. Grell, V. Nikiforova, T. Schaub, and J. Selbig. Modeling biological networks by action languages via answer set programming. *Constraints*, 13(1-2):21–65, 2008.

V. Efstathiou and A. Hunter. Algorithms for effective argumentation in classical propositional logic: A connection graph approach. In *Foundations of Information and Knowledge Systems, 5th International Symposium (FoIKS'08)*, volume 4932 of *LNCS*, pages 272–290, 2008.

P. Finn, S. Muggleton, D. Page, and A. Srinivasan. Pharmacophore discovery using the inductive logic programming system progol. *Machine Learning*, 30:241–271, 1998.

J. Fox and S. Das. *Safe and Sound: Artificial Intelligence in Hazardous Applications*. MIT Press, 2000.

E. Franconi and U. Sattler. A data warehouse conceptual data model for multidimensional aggregation. In S. Gatziu, M. Jeusfeld, M. Staudt, and Y. Vassiliou, editors, *Proceedings of the Workshop in Design and Management of Data Warehouses*, 1999.

C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 (Suppl. 1): S74–S82, 2001.

N. Fuhr. Probabilistic datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society for Information Science*, 51(2):95–110, 2000.

K. Fukuda and T. Takagi. Knowledge representation of signal transduction pathways. *Bioinformatics*, 17 (9):829–837, 2001.

D. Gaertner and F. Toni. Computing arguments and attacks in assumption-based argumentation. *IEEE Intelligent Systems*, 22(6):24–33, November/December 2007.

A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4:95–138, 2004.

N. Gorogiannis and A. Hunter. Implementing semantic merging operators using binary decision diagrams. *International Journal of Approximate Reasoning*, 49(1):234–251, 2008a.

N. Gorogiannis and A. Hunter. Merging first-order knowledge using dilation operators. In *Foundations of Information and Knowledge Systems (FOIKS'08)*, volume 4932 of *LNCS*, pages 132–150. Springer, 2008b.

G. Grahne and A. Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proceedings of the 7th International Conference on Database Theory (ICDT'99)*, volume 1540 of *LNCS*, pages 332–347. Springer, 1999.

U. Hahn, J. Wermter, R. Blasczyk, and P. Horn. Text mining: Powering the database revolution. *Nature*, 448:130, 2007.

J. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.

J. Harmon. *The Scientific Literature: A Guided Tour*. University of Chicago Press, 2007.

D. Heckerman and M. Wellman. Bayesian networks. *Communications of the ACM*, 38:27–30, 1995.

K. Hui and P. Gray. Developing finite domain constraints – a data model approach. In *Proceedings of Computation Logic 2000 Conference*, volume 1861 of *LNCS*, pages 448–462. Springer, 2000.

A. Hunter and W. Liu. Fusion rules for merging uncertain information. *Information Fusion*, 7(1):97–134, 2006.

A. Hunter and R. Summerton. A knowledge-based approach to merging information. *Knowledge-Based Systems*, 19(8):647–674, 2006.

B. Jefferys, L. Kelley, M. Sergot, J. Fox, and M. Sternberg. Capturing expert knowledgewith argumentation: A case study in bioinformatics. *Bioinformatics*, 22:924–933, 2006.

F. Jensen. *Introduction to Bayesian networks*. UCL Press, 1996.

C. Jonker, J. Snoep, J. Treur, H. Westerhoff, and W. Winjgaards. Putting intentions into cell biochemistry: An artificial intelligence perspective. *Journal of Theoretical Biology*, 214:105–134, 2002.

A. Kakas, R. Miller, and F. Toni. E-RES: a system for reasoning about actions, events and observations. In *Proccedings of the International Workshop on Non-monotonic Reasoning (NMR'00)*, 2000.

R. King, S. Muggleton, A. Srinivasan, and M. Sternberg. Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences*, 93:438–442, 1996.

R. King, K. Whelan, F. Jones, P. Reiser, C. Bryant, S. Muggleton, D. Kell, and S. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.

S. Konieczny, J. Lang, and P. Marquis. DA^2 merging operator. *Artificial Intelligence*, 157:49–79, 2004.

P. Krause and D. Clark. *Representing Uncertain Knowledge: An Artificial Intelligence Approach*. Intellect, 1993.

P. Krause, J. Fox, and P. Judson. An argumentation-based approach to risk assessment. *IMA Journal of Mathmematics in Business and Industry*, 5:249–263, 1993.

P. Krause, S. Ambler, M. Elvang-Gorannson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995a.

P. Krause, S. Ambler, M. Elvang-Gorannson, and J. Fox. Is there a role for qualitative risk assessment. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995b.

P. Krause, J. Fox, P. Judson, and M. Patel. Qualitative risk assessment fulfils a need. In *Applications of Uncertainty Formalisms*, volume 1455 of *LNCS*, pages 138–156. Springer, 1998.

M. Krtzsch, P. Hitzler, D. Vrandecic, and M. Sintek. How to reason with OWL in a logic programming system. In *Proceedings of the Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML2006)*, pages 17–26. IEEE Computer Society, 2006.

L. Laera, V. Tamma, T. Bench-Capon, and G. Semeraro. Sweetprolog: A system to integrate ontologies and rules. In *Rules and Rule Mark-up Languages for the Semantic Web*, volume 3323 of *LNCS*, pages 188–193. Springer, 2004.

L. Laera, I. Blacoe, V. Tamma, T. Payne, J. Euzenat, and T. Bench-Capon. Argumentation over ontology correspondences in mas. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2007)*, 2007.

S. Lauritzen and D. Siegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society B*, 50:157–224, 1988.

Y. Loyer, N. Spyratos, and D. Stamate. Integration of information in four-valued logics under non-uniform assumptions. In *Proceedings of 30th IEEE International Symposium on Multiple-Valued Logic (ISMVL2000)*. IEEE Press, 2000.

T. Lukasiewicz. Probabilistic logic programming. In *Proceedings of the European Conference on Artificial Intelligence (ECAI'98)*, pages 388–392. John Wiley, 1998.

J. Ma, W. Liu, and A. Hunter. Incomplete statistical information fusion and its application to clinical trials data. In *Scalable Uncertainty Management (SUM'07)*, volume 4772 of *LNCS*, pages 89–103. Springer, 2007.

J. Ma, W. Liu, A. Hunter, and W. Zhang. Performing meta-analysis with incomplete statistical information in clinical trials. *BMC Medical Research Methodology*, 8:56, 2008.

C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2000.

P. McBurney and S. Parsons. Dialectical argumentation for reasoning about chemical carcinogenicity. *Logic Journal of the IGPL*, 9(2):191–203, 2001.

T. Meyer, K. Lee, and R. Booth. Knowledge integration for description logics. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'06)*, pages 645–650, 2006.

S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

S. Muggleton. Scientific knowledge discovery using inductive logic programming. *Communications of the ACM*, 42:42–46, 1999.

D. Nardi and R. Brachman. An introduction to description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation and Applications*, chapter 1. Cambridge University Press, 2003.

N. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–87, 1986.

N. Noy and M. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *Proc. AAAI'00*, pages 450–455, 2000.

K. Oda, J. Kim, T. Ohta, D. Okanohara, T. Matsuzaki, Y. Tateisi, and J. Tsujii. New challenges for text mining: Mapping between text and manually curated pathways. *BMC Bioinformatics*, 9:1–14, 2008. (Supplement 3):S5.

S. Parsons. *Qualitative Approaches to Reasoning Under Uncertainty*. MIT Press, 1998.

S. Parsons and A. Hunter. A review of uncertainty handling formalisms. In *Applications of Uncertainty Formalisms*, volume 1455 of *LNCS*, pages 8–37. Springer, 1998.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of plausible inference*. Morgan Kaufmann, 1988.

A. Poulovassilis and P. McBrien. A general formal framework for schema transformation. *Data and Knowledge Engineering*, 28:47–71, 1998.

H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In D. Gabbay, editor, *Handbook of Philosophical Logic*, pages 219–318. Kluwer, 2002.

A. Preece, K. Hui, A. Gray, P. Marti, T. Bench-Capon, D. Jeans, and Z. Cui. The KRAFT architecture for knowledge fusion and transformation. In *Expert Systems*. Springer, 1999.

L. D. Raedt, A. Kimmig, and H. Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 2462–2467, 2007.

I. Rahwan and P. McBurney. Argumentation technology. *IEEE Intelligent Systems*, 22(6):21–23, November/December 2007.

A. Rector. Medical informatics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The Description Logics Handbook*, pages 406–426. Cambridge University Press, 2003.

M. South, G. Vreeswijk, and J. Fox. Dungine: A Java Dung reasoner. In *Computational Models of Argument (COMMA'08)*, pages 360–368, 2008.

R. Stevens, C. Goble, I. Horrocks, and S. Bechhofer. OILing the way to machine understandable bioinformatics resources. *Special issue: IEEE Information Technology in Biomedicine*, 6(2):129–134, 2001a.

R. Stevens, C. Goble, I. Horrocks, and S. Bechhofer. Building a bioinformatics ontology using OIL. *Special issue: IEEE Information Technology in Biomedicine*, 6(2):135–141, 2001b.

A. Szalay and J. Gray. 2020 Computing: Science in an exponential world. *Nature*, 440:413–414, 2007.

A. Tamaddoni-Nezhad, R. Chaleil, A. Kakas, and S. Muggleton. Abduction and induction for modelling inhibition in metabolic networks. In P. Flach, A. Kakas, and O. Ray, editors, *Proceedings of the International Workshop on the Integration of Abduction and Induction in Artificial Intelligence*, 2004.

A. Tamaddoni-Nezhad, R. Chaleil, A. Kakas, M. Sternberg, J. Nicholson, and S. Muggleton. Modeling the effects of toxins in metabolic networks. *IEEE Engineering in Medicine and Biology*, 26:37–46, 2007.

S. Teufel and M. Moens. What's yours and what's mine: Determining intellectual attribution in scientific text. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000.

S. Teufel and M. Moens. Summarising scientific articles: Experiments with relevance and rhetorical status. *Computational Linguistics*, 28:409–445, 2002.

N. Tran and C. Baral. Hypothesizing about signaling networks. *Journal of Applied Logic*, 2009. (doi:10.1016/j.jal.2008.10.001doi:10.1016/j.jal.2008.10.001).

N. Tran and C. Baral. Reasoning about non-immediate triggers in biochemical networks. *Annals of Mathematics and Artificial Intelligence*, 51(2-4):267–293, 2007.

N. Tran, C. Baral, and C. Shankland. Issues in reasoning about interaction networks in cells: Necessity of event ordering knowledge. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'05)*, pages 676–681. MIT Press, 2005.

D. Walton. *Fundamentals of Critical Argumentation*. Cambridge University Press, 2006.

R. Walton, C. Gierl, P. Yudkin, H. Mistry, M. Vessey, and J. Fox. Evaluation of computer support for prescribing (CAPSULE). *British Medical Journal*, 315:791–795, 1997.

M. Williams and A. Hunter. Harnessing ontologies for argument-based decision-making in breast cancer. In *Proceedings of the International Conference on Tools with AI (ICTAI'07)*, pages 254–261. IEEE Press, 2007.

B. Zupan, I. Bratko, J. Demsar, J. Beck, A. Kuspa, and G. Shaulsky. Abductive inference of genetic networks. In *Artificial Intelligence in Medicine*, volume 2101 of *LNCS*, pages 304–313. Springer, 2001.

B. Zupan, I. Bratko, J. Demsar, P. Juvan, J. Halter, A. Kuspa, and G. Shaulsky. Genepath: A system for automated construction of genetic networks from mutant data. *Bioinformatics*, 19(3):383–389, 2003.