

# Deductive Discovery and Composition of Resources

Richard Waldinger

Artificial Intelligence Center  
SRI International  
Menlo Park, CA 94025  
1 (650) 859-2216

waldinger@ai.sri.com

Jeff Shrager

Department of Plant Biology  
Carnegie Institution of Washington  
Stanford, CA 94305  
1 (650) 325-1251

jshrager@stanford.edu

## ABSTRACT

We consider the problem of answering a query, where the answer is not provided explicitly by any one resource, but has to be deduced from information provided by many resources; where the resources include both data and software; and where the resources are heterogeneous and not designed to work together. We adopt a deductive approach to this problem, in which the discovery of the appropriate resources and their composition is performed by a theorem prover. The techniques are domain-independent and are applied here to problems in molecular biology.

## Categories and Subject Descriptors

I.2.3 [Deduction and Theorem Proving]: Answer extraction

## General Terms

Design, Languages, Theory.

## Keywords

semantic integration, question answering, bioinformatics, answer extraction, procedural attachment, theorem proving.

## 1. RESOURCE DISCOVERY AND COMPOSITION

The difficulties of achieving interoperability of heterogeneous knowledge sources are well advertised. Much of the Semantic Web effort has been devoted to formulating annotation and rule languages whose expressive power has been restricted to facilitate efficient processing. Relatively little effort has yet been devoted to what might be regarded as the ultimate promise of the Semantic Web: the automatic discovery of resources appropriate to a given task and their composition into a new resource that solves that task.

We seek to develop techniques to support a researcher who, although ignorant of the available resources, might be guided to formulate a task description. From this our system will automatically discover the appropriate resources and compose them to perform the task. We assume neither that the resources know about each other nor that they have been designed to work together. And they may be Web services or other resources.

Throughout this paper we use a specific example to demonstrate that some of the restrictions that have been imposed in the name of efficient processing actually make the discovery and composition problem more difficult. While we shall talk here about a pure question-answering task, much of what we say applies equally well to the discovery and composition of general Web Services, which not only yield information but also make changes in the real world.

## 2. A DEDUCTIVE APPROACH

At the core of a deductive approach is a *subject domain theory*, which comprises an ontology that describes the meaning of the concepts in the domain, the capabilities of the available resources, and the background knowledge necessary to relate them. While the word “ontology” is sometimes taken to mean a description of vocabulary and taxonomy of the domain, we mean here a formal axiomatic theory that defines and relates the new concepts. We also use the word “axiom” to include what is often called a “rule” in the Semantic Web literature. We employ full first-order logic as the representation language of the subject domain theory.

The query, or question to be answered, is also expressed in the language of logic. In a deductive approach, the query is phrased as a conjecture, whose validity is to be proved by an automatic theorem prover in the subject domain theory. Once the proof is complete, answer-extraction techniques that have been introduced for program synthesis and deductive question answering [Green 1969, Manna and Waldinger 1980] are applied to the proof to yield an answer (or a program that computes an answer). If more than one proof is found, more than one answer may be extracted.

The axioms of the subject domain theory allow the query conjecture to be transformed and decomposed into subgoals. These axioms include *axiomatic advertisements*, which describe the capabilities of the available data and software resources; these include, but are not limited to, Web resources. When an axiom that advertises a resource plays a role in solving a subgoal, that resource may be involved in the composition that is extracted from the proof.

While some of the resources to be consulted are basic knowledge sources, others serve to translate the information produced by one resource into the form required by another. These translators have their own axiomatic advertisements, and their discovery and composition is performed by the same mechanism as for any resource.

While a resource may be invoked in executing a program that is composed after the proof is complete, we can also link a symbol in the theory to an external resource, by the mechanism of *procedural attachment*. If the linked symbol plays a role in the proof search, the attached resource is invoked as the proof is underway, so that it can provide needed information that is not represented explicitly in the theory. Also, when an efficient procedure exists for reasoning within a subtheory (such as arithmetic computation or temporal inference), the procedural-attachment mechanism allows such reasoning to be performed by that procedure, rather than axiomatically.

### 3. A BIOINFORMATICS QUERY

Let us consider a particular example taken from a research study of cyanobacteria, which are bacteria that can perform photosynthesis, commonly (although misleadingly) known as blue-green algae. One of these strains of bacteria, *Prochlorococcus* sp. Med4 (called here “ProMed4”), is adapted to high light and lives in the upper part of the ocean; another, which we shall call Pro9313, is adapted to low light and lives in deeper waters. We are interested in which gene products (generally proteins) are involved in this adaptation. One way to address this is to ask which genes in ProMed4 have no ortholog in Pro9313—that is, which functions are found among the genes of one but not the other organism. (Orthologous genes have common ancestry and generally a common function.) This is not a certain solution because there could be several functions represented in one but not the other organism.

One can get a finer bead on the question by looking at the results of “microarray expression” experiments for those genes, asking which genes demonstrate a significant light response—for example, those whose level of production (called its “expression level”) is doubled in light stress experiments). Unfortunately, microarrays for the prochlorococci have been developed only recently, and such experimental work does not exist. There are, however, a number of studies on a related freshwater cyanobacterium, *Synechocystis* sp. 6803 (here called s6803). Indeed, much research specific to light acclimation has been conducted on one strain of this bacterium (e.g., [Hihara et al. 2001]). Going one step further, one may focus specifically on those genes that are annotated as photosynthesis-related (e.g., by the Gene Ontology [Gene Ontology Consortium 2000]).

In sum, we can formulate the question as follows: What photosynthesis-related genes in ProMed4 have no ortholog in Pro9313 but *do* have an ortholog in Syn680 that exhibits a light stress response (e.g., greater than 2x up-regulation ratio in light stress microarray experiments.)

Let us look at the logical form (interspersed with comments) of this light-acclimation query formulated in English:

```
exists((?gene, ?gene1)
  gene-in-organism(?gene, promed4)
    [?gene is in prochlorococcus marinus med4]
  & photosynthesis-related(?gene)
    [?gene is related to photosynthesis]
  & ortholog(?gene, ?gene1)
    [?gene has an ortholog ?gene1]
  & gene-in-organism(?gene1, s6803)
    [?gene1 is in synechocystis pcc6803]
  & not exists ((gene3)
    ortholog(?gene, gene3)
    & gene-in-organism(gene3, mit9313))
    [?gene has no ortholog ?gene3 in
    prochlorococcus marinus mit9313]
  & hihara-regulation-ratio(?gene1) > 2
    [the “Hihara” ratio of ?gene1 is greater than 2]).
```

Later we shall consider how such a query might be formulated by a biologist who is unfamiliar with either the language of logic or the vocabulary of the subject-domain theory. But first let us see how the query, once formulated, can be answered automatically.

### 4. A DEDUCTIVE SOLUTION

The above query is treated as a conjecture, whose validity in the subject domain theory is to be proved by a theorem prover. The conjecture posits the existence of two genes, ?gene1 and ?gene2, that satisfy a number of conditions. (Variables, prefixed by question marks, are symbols that may be replaced by other terms as necessary to allow the proof to go through.) In proving the existence of such pairs of genes, the theorem prover will be forced to find terms that describe them. These descriptions of genes that satisfy the conditions will then be extracted from the proof. The proof will then constitute an explanation of why the described genes satisfy the desired conditions.

The meanings of the symbols of the query, such as ortholog, are defined by axioms of the theory. The theorem prover transforms the query in terms of these axioms. Unlike a logic-programming system, the theorem prover is not constrained to process the query in the order given; rather it follows its own strategic controls. There are many branches to the search space; in this section we consider only a successful one.

Certain of the symbols of the query, such as gene-in-organism, have procedural attachments to external data sources. When the theorem prover encounters the expression

```
gene-in-organism(?gene, promed4),
```

a data source will be invoked that yields all the genes in the organism promed4. In different branches of the search space, the variable ?gene will be systematically replaced by one of these genes. The procedural-attachment mechanism allows the theorem prover to behave as if the axioms of the theory express the complete list of the genes of promed4, while in reality that knowledge is imported from the external data source only when it is needed.

Let us suppose that we are in a branch of the search space in which ?gene has been replaced by the gene pmed4.pmm0817, one of the thousands of genes of promed4. The two conditions of the query, now written

```
ortholog(pmed4.pmm0817, ?gene1)
& gene-in-organism(?gene1, s6803),
```

are transformed by application of the following axiom of the subject domain theory:

```
ortholog(?gene, ?gene1)
& gene-in-organism(?gene1, ?organism)
⇔
gene-has-ortholog-in-organism(?gene, ?gene1, ?organism).
```

This axiom defines the concept gene-has-ortholog-in-organism to mean that ?gene has an ortholog ?gene1 in the organism ?organism. The result of applying this axiom to the two conditions is a single condition

```
gene-has-ortholog-in-organism(pmed4.pmm0817, ?gene1, s6803).
```

The symbol `gene-has-ortholog-in-organism` also has a procedural attachment, which invokes software that searches for an ortholog of a given gene in a given organism. This will cause the variable `?gene1` to be replaced by an ortholog of `pmed4.pmm0817` in `s6803`, if one exists. In this case, `?gene1` is replaced by the orthologous gene `s6803.ssr2595`.

Another (transformed) condition of the query,

```
not exists ((gene3)
            ortholog(pmed4.pmm0817, gene3)
            & gene-in-organism(gene3, mit9313)),
```

is transformed by the axiom that defines `gene-has-ortholog-in-organism`, and treated by a procedural attachment, just like the previous pair of conditions. Because the condition is negated, if `pmed4.pmm0817` were to have an ortholog in `mit9313`, this condition would not be satisfied, and this branch of the search space would fail. As it turns out, there is no such ortholog, and the condition is satisfied.

The (transformed) condition

```
higura-regulation-ratio(s6803.ssr2595) > 2,
```

treated by a procedural attachment that performs a computation based on the results of the light stress experiment, ensures that this ortholog has a significant light response.

The (transformed) condition

```
photosynthesis-related(pmed4.pmm0817)
```

is also treated by a procedural attachment, which checks the annotation for `pmed4.pmm0817` to ensure that it mentions the words “photo” or “light”; in fact, it does.

Once the proof is complete, the theorem prover extracts an answer to the query by examining what terms replace the variables `?gene` and `?gene1`, that is, `pmed4.pmm0817` and `s6803.ssr2595`. This is the only answer for this query. (In our implementation, it takes less than one minute to find.)

## 5. ISSUES RAISED

The above problem illustrates a number of general issues. First of all, it demonstrates how a combination of axiomatic reasoning, answer extraction, and procedural attachment gives a naïve user access to unfamiliar resources; otherwise, the user would need to discover them by hand and write a program to access them. In this case this would be quite challenging, considering that the query requires us to access large volumes of data from diverse sources, including genes in organisms, their orthologs, their annotations, and the results of light-stress experiments.

### 5.1 Limitations of OWL

Note that the logical form of the query, and the expressions necessary in its solution, would not be easily expressed in OWL. We use quantifiers, such as `exists`, and the negation, `not`, which do not exist in OWL. The negated existential quantifier is represented by the theorem prover with a Skolem function;

Skolem functions are not generally available in the Semantic Web rule languages.

The negation is true negation, not negation-as-failure. In other words, we need to know that no ortholog exists, not that we have merely failed to find one. There is a closed-world assumption applied to the procedural attachment: if the attachment finds no ortholog, we assert that no ortholog exists.

Also, note that our solution makes use of a ternary predicate symbol `gene-has-ortholog-in-organism`, which is not allowed in OWL. While the predicate symbol can be paraphrased in terms of two binary predicate symbols, `gene-in-organism` and `ortholog`, which have their own procedural attachments, the procedure for the ternary predicate symbol yields at most one ortholog, while the procedures for the two binary predicate symbols may yield thousands of genes in a given organism, and many orthologs, all of which must be sifted through.

Another point is the value of the equality relation and the use of function symbols, such as `higura-regulation-ratio`, in addition to predicate symbols. This point is better illustrated by other examples, however, e.g., ones requiring changes of units or coordinate systems.

Furthermore it is onerous to be forced to paraphrase a query to fit within the confines of a language with limited expressive power.

## 5.2 Efficiency Issues

Part of the motivation for the introduction of OWL is to achieve efficient reasoning through the limitation of expressive power. Because full first-order logic is undecidable, there is no hope of developing a fast general-purpose reasoning capability for it. Within a particular subject-domain theory, however, it is possible to develop strategic controls that allow a theorem prover to exhibit high performance that rivals that of special-purpose systems. Our solution to the light acclimation query is slower than a cleverly crafted program to answer the same query, but better than a naively encoded one. And it does require theory-specific domain engineering and strategic work to achieve good performance.

One such mechanism is to use weights applied to symbols to decide which subformula to examine first. In the example, some symbols have procedural attachments while others do not. Furthermore, as mentioned above, some procedural attachments, such as `gene-in-organism`, can be very expensive, since an organism may have thousands of genes; others, such as `gene-has-ortholog-in-organism`, are relatively cheap, since a gene usually has only a small number of orthologs in a given organism. We have found that we can greatly improve the solution time by assigning weights to the symbols proportional to the expected number of solutions offered by the corresponding attached procedure.

## 5.3 Guided Query Formulation

We earlier mentioned the problem of how a user who is ignorant of the language of logic or the vocabulary of the subject domain can formulate a query in logical form. While we have experimented with natural-language queries, the user of the system may have no idea what vocabulary the system understands. Natural language provides the illusion that the system can understand everything. It is difficult for a system to engage the user in a natural-language dialogue to indicate what it does and does not understand.

However, while naïve users may not be able to formulate the appropriate logical query, we may be able to *guide* such users to formulate logical queries, even if they are ignorant of logical notation or the vocabulary of the subject domain theory. Our approach depends upon the use of a *sorted* theory, one in which each constant is assigned a sort, i.e., an indicator of a class to which it belongs; thus, *promed4* will be declared to be of sort *organism* (or *bacterium*, a subsort of *organism*).

Each function symbol will have a declaration of the sorts of arguments it requires and the sort of value it produces; each predicate symbol will also have a declaration of the sorts of arguments it expects.

Such declarations are valuable for a theorem prover, in that they restrict search, admit shorter proofs, allow some error detection, and permit more concise axioms and queries. But a sorted theory is of special value in that it allows guided query formulation.

Let us imagine that a user was trying to formulate the complex query discussed above. The user might select the term *promed4* from a menu of known organisms. Since *promed4* is of sort *organism*, the system would then offer a menu of relations and functions that accept terms of this sort as arguments. These include *gene-in-organism* and *gene-has-ortholog-in-organism*. An English paraphrase of the meanings of these relations is provided. (Alternatively, the user can type an approximation to the desired operator, and the system offers the closest matches that accept terms of sort *organism* as arguments.) The user selects the former. The guide produces the expression

```
gene-in-organism(?gene, promed4)
```

with the English paraphrase

```
?gene is in the organism prochlorococcus marinus med4.
```

Here the new variable, *?gene*, has been introduced as a placeholder. The system offers the user the choice between further instantiating this variable or wrapping an operator around the entire expression; the user selects the latter.

The entire expression is a formula, of sort *boolean*. The only operators that can take an argument of sort *boolean* are the logical operators, *&*, *or*, *not*, *↔*, *⇒*, *forall*, and *exists*. Since there are other conditions to satisfy, the user selects *&*. The system constructs the new expression

```
gene-in-organism(?gene, promed4)
& ?boolean.
```

The user is given the choice of instantiating the new variable *?boolean* or wrapping another operator around this formula. The user selects *?boolean*, to fill in the remaining conditions. The process continues until the user is satisfied with the result. At no point is the user required to know the specific symbols the theory employs. A complex query may be divided into several subqueries, each building on the results of the earlier ones. At any point the theorem prover can be summoned to produce answers for the query-so-far.

## 5.4 Subject Domain Theory Formation

Constructing a subject domain theory is a monumental and error-prone task. Fortunately, it need only be done once for each subject domain. Furthermore, we do not need to begin from scratch. We can import appropriate sections of subject domain

theory from such standards as Cycorp's OpenCyc[Cycorp 2002] and Teknowledge's SUMO ([Niles and Pease 2001], [Teknowledge 2004]) and more specific ontologies for the selected subject domain.

The axiomatization of molecular biology included in the Library of Ontologies of the Laboratory for Applied Ontology [<http://www.loa-cnr.it/>] has been under development for many years. Other ontologies that are being developed for biology include the Ontolingua Molecular Biology Theory ([www.loa-cnr.it/medicine/molecular-biology](http://www.loa-cnr.it/medicine/molecular-biology)) and the Open Biological Ontologies project ([obo.sourceforge.net/main.html](http://obo.sourceforge.net/main.html)); see also [Baclawski and Niu 2005]

But merging subject domain theories is not straightforward. Different theories may use different mechanism for describing the same concept, or use the same symbol with different meanings. One can build a subject domain theory by composing simpler theories. The notion of the *colimit*, obtained from the mathematical theory of categories [Barr and Wells 1999], has been found to be valuable for this purpose [Burstall and Goguen 1977]. Using the colimit, one can combine theories even if component theories use different vocabulary for the same concept, or the same vocabulary for different concepts.

The colimit takes as its input a diagram of several theories that are linked by *morphisms*, or mappings between symbols. When two symbols in different theories are linked by a morphism, they are to be identified as the same in the composed theory; otherwise, even if they are syntactically identical, they are to be kept distinct in the composed theory. New axioms are then introduced into this colimit theory to express the relationships between the symbols in one theory and the symbols in the others. (As a simple example, if one theory measures distances in miles, and another measures distances in kilometers, we may introduce an axiom that says how many kilometers are in a mile. Similarly for rectangular and polar coordinates.) Caution must be observed: the faulty formation of colimits can cause inconsistency. Also one must be careful that efficiency of inference is preserved in the new theory.

When new content must be introduced, the same mechanism we used in guided query formulation can also be used to provide guided axiom formulation. In this way, a subject domain expert who is ignorant of the language of logic or of the vocabulary of the existing subject-domain theory can be guided to add new axioms to the theory.

## 5.5 Integration of OWL Resources and Web Services

Special attention is paid to the integration of ontologies and services that are expressed in OWL or provided with OWL annotation. This is possible because first-order logic is stronger than OWL; OWL has a first-order semantics, and even constructs that seem to be higher-order, such as the cardinality restrictions, can be paraphrased in first-order logic. A first-order theorem prover, provided with appropriate strategic controls, can act as an interpreter for rule languages such as RuleML ([www.ruleml.org](http://www.ruleml.org)) and SWRL ([www.daml.org/2003/11/swrl](http://www.daml.org/2003/11/swrl)).

When resource is registered as a Web Service, its Web Service description can then be expressed as a first-order axiomatic advertisement for the service. A procedural attachment can be introduced to link the Web service with the symbol in the theory that denotes it. In this way, the Web service can be invoked as the

proof is in progress. This is particularly appropriate for an informational service, which yields information but produces no changes in the world.

Alternatively, we can extract from the proof an answer term that includes invocations of the appropriate service, so that the service is only invoked after the proof is complete. This is appropriate for Web Services with side effects, since we do not want to invoke the service until we it has been proven that it takes part in a complete solution to the problem at hand.

## 6. Prototypical Implementation

While the approach we are outlining is independent of any particular implementation, we have been experimenting with a prototype in the biological domain, called BioDeducta, built largely from existing components. The theorem prover is SRI's first-order reasoning system SNARK [Stickel et al. 2002]. The subject domain theory is formulated in the language of SNARK—a first-order logic with LISP syntax. Biological data and software resources are drawn from the BioBike environment [Massar et al. 2005], a biological data repository and biology-based programming environment at Stanford University. We intend to draw data also from the BioWarehouse [Lee et al. 2005], which loads data from a variety of sources into a single database scheme. A new guided query-formulation environment is under development. Composition of theories will be performed by Specware, a category-theory-based formal software development environment of the Kestrel Institute; We say a bit about each of these components.

### 6.1 SNARK

SNARK [Stickel et al. 2000] is an automatic theorem prover implemented at SRI. It contains many of the features we need for the composition of resources. It has a mechanism for extracting answers to queries from proofs. It has a procedural-attachment mechanism that allows us to consult external resources while a proof is in progress. It operates in a sorted logic, which is necessary for guided query formulation. It has a built-in version of the Allen temporal interval calculus [Allen 1983] and of the Regional Connection Calculus [Randell et al. 1992] for reasoning about spatial regions. It is written in Common Lisp.

### 6.2 BioBike

BioBike [Massar et al. 2005] is an online biological knowledge base, a repository of biological data sources, and an associated LISP-based programming language. BioBike provides access to a number of data sources, including the Gene Ontology (GO) [Gene Ontology Consortium 2000], the Kegg Database ([www.genome.jp/kegg/](http://www.genome.jp/kegg/)), the BioCyc Database for Cyanobacteria, and PubMed; as well as software tools such as BLAST and Clustal. SNARK has been incorporated into BioBike; procedural attachment links symbols of the subject domain theory to procedures in BioBike.

## 6.3 Specware

Specware [Smith 2006] is a software development environment created by the Kestrel Institute. A category-theory-based framework, it implements essential theory-manipulation operations, including the colimit notion we require for the composition of multiple ontologies and theories. It provides an interface that allows us to access and generate SNARK theories and proofs. We are planning to use the theory composition facilities of Specware to fuse a number of existing general purpose and biological theories and ontologies.

## 7. Antecedents

BioDeducta is the most recent of a series of efforts in applying automated deduction to the composition of knowledge resources.

### 7.1 Amphion

NASA's Amphion, a software composition system ([Lowry et al. 1994, Stickel et al. 1994]), automatically produces software for planetary astronomers by composing components provided by a subroutine library, including routines to access ephemeris tables and the data from interplanetary missions. Amphion accepts user queries formulated with the help of a graphical query-formulation guide, which produces a diagrammatic representation of the query. The diagram is then translated into a logical conjecture, which is passed to SNARK for proof. A composition of calls to subroutines in the library is then extracted from the proof. The subroutines are themselves heterogeneous; they often use different spatial coordinate systems, time measurement schemes---the finite speed of light is significant---and other representations, but other subroutines are invoked to translate from one representation to another. Software constructed by Amphion has been used to plan photography and analyze data from the Cassini mission to Saturn.

### 7.2 GeoLogica and QUARK

SRI's GeoLogica and QUARK [Waldinger et al. 2004] are both experimental systems that use deductive methods to compose heterogeneous data and software components, much in the way we discuss here. While GeoLogica responds to queries posed by an Earth systems scientist, QUARK serves as an assistant to an intelligence analyst. Both systems accept queries in English; they are translated into logical form by a natural-language parser. The logical form is presented as a conjecture to SNARK, and an answer to the query is extracted from the proof. SNARK's procedural-attachment mechanism is used to access external components, such as the Alexandria Digital Library Gazetteer. Sources annotated with OWL, such as the CIA World Factbook [CIA 2003], are accessed through the Semantic Web browser ASCS of Teknowledge ([projects.teknowledge.com/DAML](http://projects.teknowledge.com/DAML)). Both systems rely on SNARK's fast temporal and spatial reasoning capabilities.

## 8. OTHER RELATED WORK

We shall not attempt to survey work in general program synthesis or planning, much of which is relevant to the problem of the composition of resources. We shall mention only a couple of

projects that are directly tied to the deductive composition of Web services.

The Infomaster system [Genesereth et al. 1997] uses deductive methods to integrate multiple Web sources for such applications as searching rental advertisements of product catalogs. It uses deductive techniques—logic programming rather than full theorem proving—for coordination. Special attention is paid to the efficiency of the computation by which the answer to a request is discovered.

The Ariadne system [Knoblock et al. 2000] also uses deductive methods to integrate Web services. It incorporates special features that makes it easier to introduce procedural attachments for new services. It has been applied to geographical question answering and to entertainment guidance (e.g., planning an evening including dinner and a movie).

## 9. CONCLUDING SUMMARY

Automated deduction is appropriate technology for the composition of resources, including local sources and Web services. While full first-order logic has tended to be discounted for its formal undecidability, it has been shown to be viable when supplied with subject-specific strategic controls and it may compete favorably with more restricted, less expressive languages.

## 10. ACKNOWLEDGEMENTS

We would like to thank Mark Stickel for help with the use of the SNARK theorem prover, J. P. Massar for help with the BioBike server, Yannick Pouliot for advice on problem domains and on the use of the BioWarehouse, and Carolyn Talcott and Merrill Knapp for discussions on data sources and formalization of the subject domain. The light acclimation example is based on a problem posed by Jeff Elhai. This research has been supported in part by the National Science Foundation, under the Science and Engineering Information Integration and Informatics (SEIII) Program.

## 11. REFERENCES

[1] [Allen 1983] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11), Nov. 1983.

[2] [Baclawski and Niu 2005] K. Baclawski and T. Niu. *Ontologies for Bioinformatics*. MIT Press. (October, 2005)

[3] [Barr and Wells 1999] Barr, M. and Wells, C. *Category Theory for Computing Science*. Les Publications CRM, Montreal, third edition, 1999.

[4] [Burstall and Goguen 1977] Burstall, R. and Goguen, J. Putting Theories Together to Make Specifications. *Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI)*, Cambridge, Massachusetts, pp. 1045-1058, 1977

[5] [CIA 2003] CIA. *The World Factbook*, 2003. [www.cia.gov/cia/publications/factbook/](http://www.cia.gov/cia/publications/factbook/)

[6] [Cycorp 2002] Cycorp, *The Ontological Engineers' Handbook*. <http://www.cyc.com/doc/handbook/oe/oe-handbook-toc-opencyc.html>

[7] [Gene Ontology Consortium 2000] The Gene Ontology Consortium (2000) Gene Ontology: tool for the unification of biology. *Nature Genet.* 25: 25-29.

[8] [Genesereth et al. 1997] Genesereth, M.R., A.M. Keller, and O.M. Duschka. Infomaster: An information integration system. In *Proceedings of the 1997 ACM SICMOD Conference*, ACM Press, 1997.

[9] [Green 1969] Green, C.C. Application of Theorem Proving to Problem Solving. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Washington, DC, May 7-9, 1969.

[10] [Hihara et al. 2001] Hihara Y, Kamei A, Kanehisa M, Kaplan A, Ikeuchi M. (2001) DNA microarray analysis of cyanobacterial gene expression during acclimation to high light. *Plant Cell*, 13(4):793-806.

[11] [Knoblock et al. 2001] Knoblock, C. A., S. Minton, J.L. Ambite, N. Ashish, I. Muslea, A.G. Philpot, and S. Tejada. The Ariadne approach to Web-based information integration. *International Journal on Cooperative Information Systems (IJCIS) 10 (1-2) Special Issue on Intelligent Information Agents: Theory and Applications*, pp. 145-169, 2001. <http://citeseer.ist.psu.edu/cache/papers/cs/4385/http:zSzzSzwwww.isi.edu:zSzsimszSznaveenzSzaai98.pdf/knoblock98modelling.pdf/>

[12] [Lee et al. 2005] Lee, T., Y. Pouliot, V. Wagner, P. Gupta, D. W. J. Stringer-Calvert, J. D. Tenenbaum, and P.D. Karp. BioWarehouse: A bioinformatics database warehouse toolkit. *Bioinformatics*, to appear 2005.

[13] [Lowry et al. 1994] Lowry, M. R., Philpot, A., Pressburger, T., and Underwood, I. AMPHION: Automatic Programming for Scientific Subroutine Libraries. *ISMIS 1994*: 326-335.

[14] [McGuinness and van Harmelen 2004] McGuinness, D. and van Harmelen, F., eds. *OWL Web Ontology Language Overview*. Technical report. W3C, February 2004. <http://www.w3.org/TR/owl-features/>

[15] [Manna and Waldinger 1980] Manna, Z., and Waldinger, R., A deductive approach to program synthesis. *ACM Transactions on Programming, Languages, and Systems*, 2:90-121, 1980.

[16] [Massar et al. 2005] J. P. Massar, Michael Travers, Jeff Elhai, and Jeff Shrager (2005) BioLingua: a programmable knowledge environment for biologists. *Bioinformatics* 21(2):199-207.

[17] [Niles and Pease 2001] Niles, I., and A. Pease. Towards a Standard Upper Ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Chris Welty and Barry Smith, eds, Ogunquit, Maine, October 17-19, 2001.

[18] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. *Proc. KR-92*, 1992. Morgan Kaufmann

- [19][Smith 2006] Douglas R. Smith, Composition by Colimit and Formal Software Development, Technical report, Kestrel Institute, 2006.
- [20][Stickel et al. 2002] Stickel, M., Waldinger, R., Chaudhri, V.: A Guide to SNARK. Technical report. SRI International, Artificial Intelligence Center (2002) [www.ai.sri.com/snark/tutorial/tutorial.html](http://www.ai.sri.com/snark/tutorial/tutorial.html)
- [21][Stickel et al. 1994] Stickel, M., Waldinger, R., Lowry, M., Pressburger, T. and Underwood, I. Deductive Composition of Astronomical Software from Subroutine Libraries, in 12th Conference on Automated Deduction, Nancy, France, June 28-July 1, 1994. In Automated Deduction, A. Bundy, ed., Springer-Verlag Lecture Notes in Computer Science, Vol. 814.
- [22][Teknowledge 2004] Teknowledge. SUMO Ontology. Technical report. [ontology.teknowledge.com/](http://ontology.teknowledge.com/) 2004.
- [23][Waldinger et al. 2004] Waldinger, R., D. Appelt, J. Fry, D. Israel, P. Jarvis, D. Martin, S. Riehemann, M. Stickel, M. Tyson, J. Hobbs, J., and J. Dungan. Deductive Question Answering from Multiple Resources. In New Directions in Question Answering, AAAI, 2004.