Index Expression Belief Networks for Information Disclosure

P.D. Bruza, L.C. van der Gaag

RUU-CS-92-21
June 1992

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

Index Expression Belief Networks for Information Disclosure

P.D. Bruza, L.C. van der Gaag

Technical Report RUU-CS-92-21
June 1992

Department of Computer Science
Utrecht University
P.O.Box 80.089
3508 TB Utrecht
The Netherlands

Index Expression Belief Networks for Information Disclosure

P.D. Bruza, L.C. van der Gaag

RUU-CS-92-21
June 1992

Utrecht University

Department of Computer Science

Padualaan 14, P.O. Box 80.089,
3508 TB Utrecht, The Netherlands,
Tel. : ... + 31 - 30 - 531454

ISSN: 0924-3275

Index Expression Belief Networks for Information Disclosure

P.D. Bruza *
Dept. of Information Systems
Faculty of Mathematics and Informatics
University of Nijmegen
Toernooiveld 1, 6525 ED Nijmegen
The Netherlands
e-mail: peterb@cs.kun.nl

L.C. van der Gaag
Dept. of Computer Science
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands
e-mail: linda@cs.ruu.nl

Abstract

It is widely accepted that to extend the effectiveness of information disclosure beyond the limitations of current empirically-based retrieval systems, some notion of document semantics has to be incorporated into the retrieval mechanism. A recent approach to bringing semantics into play is to found the retrieval mechanism on the notion of logical inference. In this paper, we build on this approach and describe a promising new mechanism for information disclosure, called the Refinement Machine. The Refinement Machine features the language of index expressions as a language for characterizing information objects and a deduction mechanism driven by rules of inference. Two types of inference rule are distinguished. The rules of strict inference follow the line of traditional logical deduction. As the characterizations of objects are incomplete and requests are typically partial descriptions of the information need, the rules of strict inference are supplemented with a rule of plausible inference. This rule of plausible inference is motivated by recent work in the area of plausible reasoning in knowledge-based systems and, in particular, is derived from the work on belief networks. Besides giving details of the Refinement Machine, this paper also presents some preliminary experimental results.

1 Introduction

We are currently experiencing the *Information Age*. Information is proliferating itself faster and faster with the consequence that organizations are becoming burdened with an information overload; filing cabinets full of dossiers and thousands of (incompatible) word processor files abound. It is becoming increasingly difficult for organizations to control these mounds of information, let alone make effective use of it. *Information disclosure* can simply be described as the quest to find all objects that are relevant to a searcher's information need.

The information disclosure problem is represented schematically in Figure 1. The problem begins with a person having an information need that they wish to fulfill. Henceforth, we will term this person the *searcher* and denote the information need by N. The information need is typically formulated in the form of a *request*, denoted by q, which is given to an automatic system, or a human intermediary such as a librarian. The intention is that the request q is an as good as possible description of the information need N. On the other hand, there is the information itself. This is modelled as a set $\mathcal{O}$ of *information objects*; the information objects

*This work has been partially supported by ESPRIT project APPED (2499).

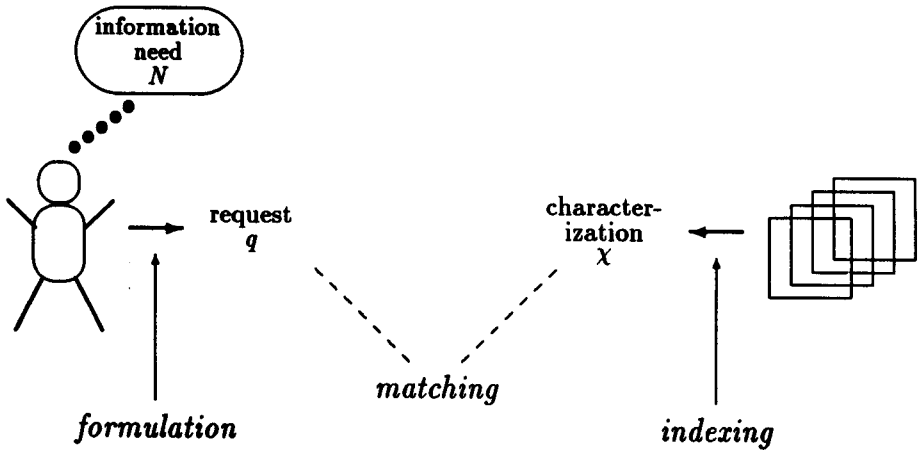

Figure 1: The Information Disclosure Problem

are also referred to as information carriers or documents. Each information object $O \in \mathcal{O}$ is characterized by a set of *descriptors* to facilitate its disclosure; these descriptors are drawn from a descriptor language $\mathcal{C}$. The characterization of an object O will be denoted by $\chi(O)$. The characterization is arrived at by a process called *indexing*. Now, information disclosure is typically driven by a process called *matching*. In this process, the request q of the searcher is matched with the characterizations $\chi(O)$ of the objects O in $\mathcal{O}$. If the matching algorithm deems an information object as being sufficiently similar to the request, then the object is assumed relevant and returned to the searcher, thus disclosing it.

Although only a limited number of concepts are involved in information disclosure, it nevertheless turns out to be a formidable problem due to the following reasons:

- *formulation* is not easy
- *indexing* produces incomplete object characterizations
- *matching* is based on nefarious assumptions

In this paper we describe a promising new framework for information disclosure which features the so-called *language of index expressions* as a language for characterizing information objects. Furthermore, a matching process is introduced which is based on deduction driven by an Information Disclosure Machine providing rules of strict and plausible inference. An important focus of the paper is on the notion of plausible deduction. This comes to the fore as characterizations of objects are incomplete and requests typically partial descriptions of the information need. Two approaches to plausible deduction are highlighted: a *context-free* approach in which specific rules are used with which to reason with uncertainty, and a more *context-sensitive* approach in which uncertainty is handled by a belief network. The second approach is chosen to be incorporated into the framework of the *Refinement Machine*, a particular concretization of an Information Disclosure Machine.

The paper is structured in the following manner. Section 2 provides some background to logic-based information disclosure; an important notion introduced in this section is that of the Information Disclosure Machine, which serves as the integrating framework for the ideas presented in this paper. Section 3 presents the language of index expressions as a

characterization language for information objects. In Section 4, the strict inference mechanism of the Refinement Machine is featured which is based on a logic defined over the language of index expressions. In Section 5 a set of rules of plausible inference is introduced based on context-free inference over index expressions. This inference mechanism is shown to be inadequate. Therefore, in Section 6 a more context-sensitive approach to plausible inference is presented based on the reasoning mechanism provided by a belief network built from a set of index expressions. This approach is subsequently incorporated into the Refinement Machine. Finally, in Section 7, the potential effectiveness of the Refinement Machine is explored in the light of some preliminary experimental results. The paper is rounded off with some conclusions and directions for further research.

2 Background and Preliminaries

The information disclosure system presented in this paper is motivated by the work of C.J. van Rijsbergen [Rij86a], [Rij86b], [Rij89]. Van Rijsbergen has recently developed a new information disclosure system in which the matching between a request and an object characterization is founded on the notion of logical inference. This contrasts with state-of-the-art disclosure systems, where the matching process is driven by empirical relations between query terms and documents. These empirically-based retrieval systems have the inherent limitation that they do not in any substantial way incorporate the meaning of an object. Much of the research in document retrieval over the last thirty years has been directed at maximizing disclosure effectiveness within this limitation. Some, however, believe that the limits of exploitation of empirically-based systems have been reached, [Rij86a]. Quite a considerable amount of recent research in document retrieval has therefore been directed at incorporating document semantics to arrive at more effective retrieval, see for example [SvR90], [Wea88], [BC89]. In this section we confine ourselves to a brief discussion of a logic-based approach to information disclosure only as it forms the basis of our disclosure system.

The crucial questions with regard to incorporating document semantics in information disclosure are as to how the meaning of an object should be represented, and given that the meaning is available, how can it be used to render effective disclosure. In the logic-based approach to information disclosure an object is assumed to have a formal semantics in the form of a set of so-called *axioms*. Each axiom describes or characterizes a part of the content of the object. In logic, a model is an interpretation in which all given axioms hold. In this sense, an information object O can be said to form a model of its associated axioms A, denoted as $O \models A$. Also in logic, a well-formed formula W can be deduced, or proved, from a set A of axioms by applying so-called rules of inference; the provability of W from A is often denoted as $A \vdash W$. A formal theory for information disclosure can now be constructed in the following way. The basis is a set of primitive descriptors which are used to describe the content of information objects; these primitive descriptors are the atomic formulae in information disclosure. The well-formed formulae are complex expressions involving primitive descriptors and can be used as more sophisticated characterizations of document content than the primitive descriptors. The language of complex descriptors is denoted by $\mathcal{C}$, and the axiomatization of objects is drawn from this language, that is, for each information object O we have $\chi(O) \subseteq \mathcal{C}$. In addition, it is assumed that the request q is a complex descriptor, so $q \in \mathcal{C}$. Furthermore, a set of rules of inference is assumed with which we attempt to derive the request q from a given object characterization $\chi(O)$. The

first possibility is $\chi(O) \vdash q$, meaning that the request q can indeed be proved from the axiom set of object O. From $\chi(O) \vdash q$ we are sure that O is a model for the request q, or in less formal terms, that object O deals with, or *is about*, q. Therefore, O is relevant with respect to q and should be returned in response to the request q. If q cannot be deduced from $\chi(O)$, however, then no definitive statement can be made about O being relevant with respect to q; it only means that the relevance of O with respect to q cannot be proved from the axioms associated with O.

The above concepts form a so-called Information Disclosure Machine. Before defining the Information Disclosure Machine more formally, we introduce the notion of a Disclosure Structure.

Definition 2.1 *A* Disclosure Structure *is a triple* $D = \langle \mathcal{O}, \mathcal{C}, \chi \rangle$ *where*

- $\mathcal{O}$ *is a set of information objects*
- $\mathcal{C}$ *is a descriptor language*
- $\chi \subseteq \mathcal{O} \times \mathcal{C}$ *is an indexing relation*

After having defined a Disclosure Structure, the question remains as to how the relevance of an object O in response to a request q can be established. This is effectuated by the Information Disclosure Machine which is driven by a process of logic-based inference as outlined above.

Definition 2.2 *An* Information Disclosure Machine, *or* Disclosure Machine *for short, is a triple* $\Delta = \langle D, S, P \rangle$ *where*

- D *is a Disclosure Structure*
- S *is a set of rules of strict inference*
- P *is a set of rules of plausible inference*

We define the notion of derivation using the rules of inference of a Disclosure Machine.

Definition 2.3 *Let* $\Delta = \langle D, S, P \rangle$ *be a Disclosure Machine. Let* $\mathcal{C}$ *be the descriptor language of the Disclosure Structure* D. *For* $d \subseteq \mathcal{C}, x \in \mathcal{C}$ *and* $s \in S$, *we use* $d \vdash_s x$ *to denote that* x *can be deduced from* d *by applying the rule of strict inference* s. *Furthermore,* $d \vdash_\Delta x$ *denotes a sequence of zero or more deduction steps involving rules of strict inference from* Δ. *Analogously, we use* $d \mathrel{|\!\sim}_p x$ *to denote that* x *is plausibly deduced from* d *via the rule of plausible inference* $p \in P$; $d \mathrel{|\!\sim}_\Delta x$ *denotes a sequence of one or more deduction steps involving rules of strict inference and rules of plausible inference of* Δ *such that at least one step involves a rule of plausible inference.*

For reasons of brevity, in the sequel the subscript Δ will often be dropped from $\vdash_\Delta$ and $\mathrel{|\!\sim}_\Delta$.

We will take a closer look at the rules of inference of a Disclosure Machine. To begin with, *strict inference* based on object characterizations will be detailed. Earlier it was stated that from $\chi(O) \vdash q$ it is sure that $O \models q$. This statement is based on the assumptions that all object characterizations are valid and the rules of strict inference preserve relevance. Under these assumptions the Disclosure Machine is said to be *sound*:

$$\chi(O) \vdash q \;\Rightarrow\; O \models q$$

In other words, in a sound Disclosure Machine the objects from whose characterization the request q can be derived are relevant with respect to q. The converse of soundness is *completeness*. Completeness states that all valid propositions are deducible, or:

$$O \models q \;\Rightarrow\; \chi(O) \vdash q$$

A complete Disclosure Machine has the advantage that relevance can be established purely by strict inference. Unfortunately, a complete Disclosure Machine turns out to be very difficult to realize. This is due to the fact that the characterization of objects seems to be inherently incomplete. For this reason the power of the strict inference mechanism is limited, meaning that in general it is not often the case that an object can be proved relevant via strict inference only. Now note that if a request q cannot be (strictly) deduced from the axioms of an information object O, this does *not* necessarily mean that O is not relevant with respect to q. It only means that the axioms of O are too weak to establish the validity of q in O. In other words, it is important that the Disclosure Machine does not employ an implicit Closed World Assumption; this assumption would state that if a request q is not deducible via the rules of strict inference, then the object O is irrelevant with respect to q:

$$\neg(\chi(O) \vdash q) \;\Rightarrow\; \neg(O \models q)$$

It will be evident that applying only rules of strict inference results in an imbalance between relevance and derivability. To alleviate this imbalance *plausible inference* is used. The plausible inference mechanism strives to generate high probabilities of relevance for those relevant information objects that escaped the strict inference mechanism. If, for a given object O, the probability of relevance is high, then the Disclosure Machine might return the object after all. The plausible inference mechanism of a Disclosure Machine can be founded on the principle of *minimal axiomatic extension* which has its roots in the so-called logical uncertainty principle documented in [Rij86b]. The principle of minimal axiomatic extension states that the probability of an object being relevant to a request is inversely proportional to the minimal extension of the object description allowing to prove the request. It is important to note that either the characterization of the object must be extended with new axioms, or some axiom(s) of the description have to be *strengthened*; an inverse approach to description strengthening is query weakening [Nie86]. It is easy to see that by strengthening axioms, the deduction process becomes less certain because it involves suppositions that were not originally a part of the semantics of the object.

The Sections 4 and 6 feature a concretization of the Information Disclosure Machine, the so-called *Refinement Machine*. This Refinement Machine employs an inference mechanism defined over the language of index expressions. This language is the topic of the next section.

3 The Language of Index Expressions

One of the main assumptions underlying information disclosure research is that the better the characterization of an object, the better the possibilities to disclose it. The content descriptors of an information object must distinguish it from other objects to enable identifying it as relevant to a given request. However, the content descriptors must not only discriminate between objects but also be usable with respect to the searcher. For example, the disk address of an object is a perfect discriminator but is useless in regard to the formulation of requests. In this section, we introduce a descriptor language, called the language of index expressions, that satisfies these criteria.

3.1 Term and Term Phrase Descriptors

A descriptor can take several forms. The most elementary form of descriptor is a keyword, or *term*. Term descriptors have the advantage that there are a number of straightforward indexing methods to automatically derive them from information objects [Sal89]; the disadvantage of terms is that they sometimes lack discriminatory power as is particularly true in large object bases.

An extension to term descriptors is the *term phrase descriptor*, or *term phrase* for short. In general, a term phrase is a sequence of one or more terms; the most common term phrases consist of two terms. Term phrase descriptors have an increased specificity and thus an enhanced ability for discrimination. For example, the term phrase **computer programming** is more specific and thus more discriminating than each of the terms **computer** and **programming** separately. Even though straightforward indexing methods exist for generating term phrases [Sal89], these methods often suffer from the problem that either too many non-meaningful phrases are generated, or conversely a large percentage of the phrases are meaningful but the resulting object characterizations lack completeness.

3.2 Index Expression Descriptors

An extension to term phrase descriptors is the so-called *index expression descriptor*, or *index expression* for short. As a term phrase, an index expression consists of a number of terms; however, in an index expression the relationships between the separate terms are modelled explicitly by means of so-called *connectors*. The motivation behind this is that much of the content of an information object is embodied in term relationships [Far80]. A parallel can be drawn here with the conceptual model from the database world where relationship types between entities play an important role; the characterization of an object consisting simply of keywords would be like an entity relationship model without relationship types.

Definition 3.1 *Let T be a set of terms and C a set of connectors. We define the* language $\mathcal{L}(T, C)$ of index expressions *over T and C by the following syntax (in extended BNF format):*

$$\begin{aligned}
&\textsf{Expr} \rightarrow \epsilon \mid \textsf{Nexpr} \\
&\textsf{Nexpr} \rightarrow \textsf{Term}\ \{\textsf{Connector Nexpr}\}^* \\
&\textsf{Term} \rightarrow \textsf{t}\ ,\ \textsf{t} \in T \\
&\textsf{Connector} \rightarrow \textsf{c}\ ,\ \textsf{c} \in C
\end{aligned}$$

We use ϵ to denote the empty index expression. A term t basically corresponds to a noun, noun-qualifying adjective or noun phrase; a connector c denotes a relationship type between two terms and is basically restricted to the prepositions and the so-called *null connector* which is denoted by o. Figure 2 shows some of the allowable connectors and the relationship types they denote.

In contrast to terms and term phrases, index expressions have associated a structure. The way the structure of an index expression is derived will be discussed briefly in Section 3.4. For now it suffices to note that the structure of an index expression is tree-like. In the sequel, we will often indicate the structure of an index expression explicitly and represent the expression by its associated tree. For example, the index expression **attitudes of (students of (universities)) to (war in (vietnam))** is depicted in Figure 3.

The more expressive nature of index expressions over terms and term phrases will be evident from the following observation. Let $\mathcal{L}(T, C)$ denote the language of index expressions

Connector	*Relationship Type*	*Examples*
of	possession, action-object	castle of queen, pollination of crops
by	action-agent	voting by students
in, on, etc.	position	trees in garden
to, on, for, in	directed association	attitudes to courses, research on voting
with, o, and	association	assistance with problems, fruit o trees
as	equivalence	humans as searchers

Figure 2: Connector Table

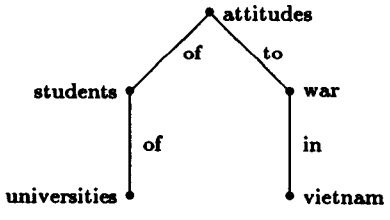

Figure 3: An Example Index Expression

over a set of terms T and a set of connectors C. Then, the term phrases are described by the language $\mathcal{L}(T, \{\circ\})$ and the terms by $\mathcal{L}(T, \emptyset)$. Now note that

$$\mathcal{L}(T, C) \supset \mathcal{L}(T, \{\circ\}) \supset \mathcal{L}(T, \emptyset)$$

Also note that $\mathcal{L}(\{t\}, \{c\})$ is an infinite language whereas $\mathcal{L}(\{t\}, \emptyset) = \{t\}$ is finite.

3.3 Power Index Expressions

Building on the notion of an index expression, we define the notion of a *power index expression.* This notion bears a strong resemblance to the power set concept: the power index expression of an index expression is the set of all its index subexpressions. We introduce the notion of an index subexpression of a given index expression informally in terms of its graphical representation: an index subexpression of a given index expression is an index expression represented by a subgraph of the representation of the given index expression. For example, the index expressions **war in vietnam** and **attitudes of students to war** are index subexpressions of the index expression **attitudes of students of universities to war in vietnam** depicted in Figure 3, whereas **students in vietnam** is not. In the sequel, we will use $\trianglelefteq$ to denote the is-subexpression-of relation over a language of index expressions, that is, we take $i \trianglelefteq j$ to denote that i is an index subexpression of j in the sense described above. Note that the relation $\trianglelefteq$ is reflexive, antisymmetric and transitive; in fact, $(\mathcal{L}(T, C), \trianglelefteq)$ is a poset.

We now define the power index expression of a given index expression more formally.

Definition 3.2 *Let i be an index expression in a language $\mathcal{L}(T, C)$. The* power index expression *of i, denoted by $\mathcal{P}(i)$, is the set*

$$\mathcal{P}(i) \quad = \quad \{j \mid j \trianglelefteq i\}$$

where $\trianglelefteq$ is the is-subexpression-of relation as above.

Note that for any index expression i in a language $\mathcal{L}(T, C)$ we have that $\mathcal{P}(i) \subseteq \mathcal{L}(T, C)$.

Like the power set of a given set, the power index expression of a given index expression forms a lattice where the underlying ordering relation is $\trianglelefteq$. The top of this lattice is the index expression itself and the bottom is the empty index expression ϵ. The Hasse diagram of the power index expression of the index expression represented in Figure 3 is depicted in Figure 4.

As we are going to exploit power index expressions for information disclosure, it is useful to have an indication of the size of the power index expression of a given index expression. For this purpose, an upper bound and a lower bound on the number of elements in the power index expression of an arbitrary index expression are given. Let i_n be an index expression comprising n terms. We use $s(i_n)$ to denote the size of the power index expression of i_n. Then, the upper bound on $s(i_n)$ is given by $2^{n-1} + n$, and the lower bound by $\frac{n(n+1)}{2} + 1$. These bounds are attained for so-called umbrella expressions and for path expressions respectively, as depicted in Figure 5. Unfortunately, the upper bound on the size of the power index expression of an index expression is exponential in the number of terms. However, in practice umbrella expressions are rare. Our experiences with index expressions thus far, have shown that a polynomial size of the power index expression is more common. For the proofs of the bounds mentioned above and a discussion of other useful properties of index expressions, the reader is referred to [Bru92].

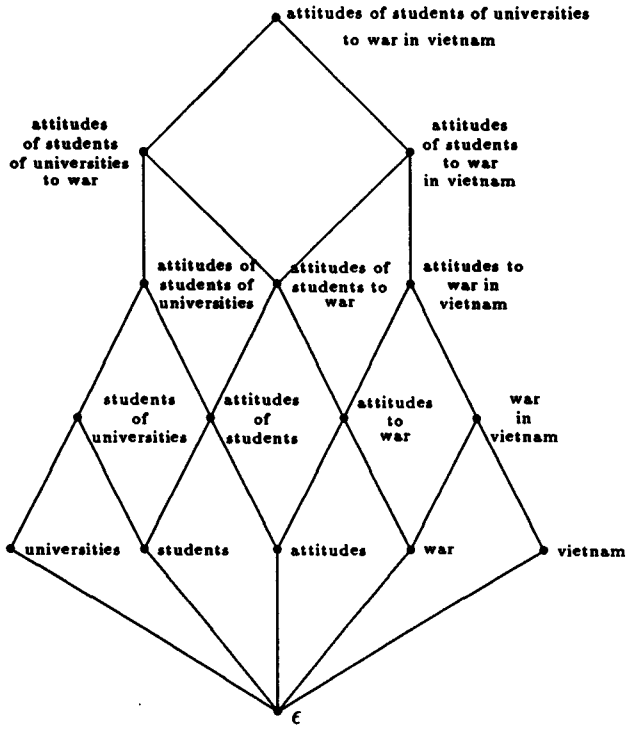

Figure 4: An Example Power Index Expression

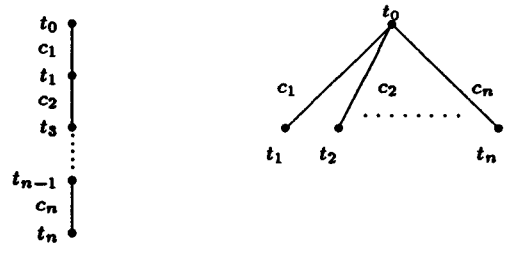

Figure 5: Path and Umbrella Expressions

Thus far we have looked at the power index expression of a single index expression only. For a set of objects, however, a core set of index expressions is generated each of which gives rise to a power index expression. These power index expressions may have some non-trivial overlap. For example, the power index expressions of the index expressions **effective o information o retrieval** and **people in need of information** share the index expressions **information** and ϵ. Now, by forming the union of all power index expressions for a set of objects, that is, by taking

$$\bigcup_{i \in \mathcal{I}} \mathcal{P}(i)$$

where $\mathcal{I}$ is the core set of index expressions generated, a lattice-like structure is rendered. For the index expressions mentioned above the structure shown in Figure 6 is yielded. Note that this structure is not a lattice since for example no join exists for the two index expressions **effective o information o retrieval** and **people in need of information**. The lattice-like structure of a union of power index expressions will be termed a *lithoid* because the associated diagram resembles a crystalline structure.

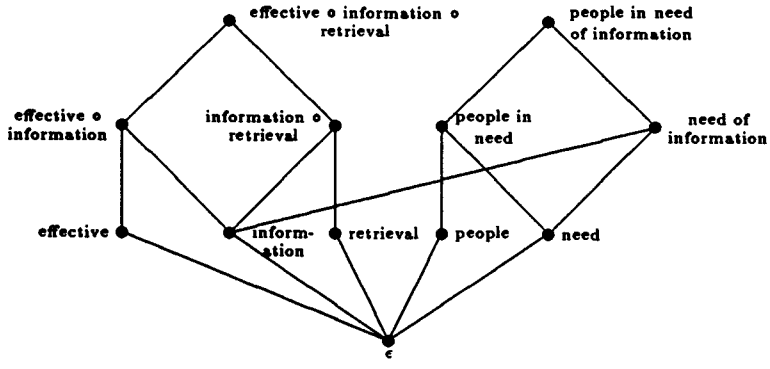

Figure 6: An Example Lithoid

One way of exploiting the lithoid for the purpose of information disclosure is as follows. If we take every vertex in the lattice as a potential focus of the searcher, then the surrounding vertices are enlargements or refinements of the context represented by the focus. The searcher can browse through the lithoid by refining or enlarging the current focus until a focus is found that fits the information need. Searching in this way is coined *query by navigation*, [BvdW90], [Bru90].

3.4 Automatic Generation of Index Expressions

T.C. Craven [Cra86] mentions that the title of an information object and the titles of its sections, subsections and figures often have a form very similar to that of index expressions when stop words such as **the** and **a** are removed. This suggests that automatic generation of index expressions can be based on these headings. Unfortunately, headings will not always be content-revealing as sometimes they are used for structural organization; consider for example, the heading *Introduction*. In this paper, we do not discuss this issue in further detail; we assume that a core set of index expressions has been identified. As to deriving the tree structure of a given index expression, we confine ourselves to summarizing the main

algorithm from [BBB91]. After the removal of stop words from a title the remaining tokens are successively processed in building a representation of the corresponding index expression. In the case of a connector token, its priority is used to decide whether the current tree is to be deepened, or broadened. The underlying idea of this approach is the observation that some connectors bind terms more strongly than others; those that bind stronger lead to deepening in the structure. For more details the reader is referred to [Bru92]. It will be evident that once a core set of index expressions has been identified and the index expressions have been associated a structure, the lithoid can be constructed automatically.

4 The Refinement Machine: Rules of Strict Inference

In the previous section, the language of index expressions was introduced as a language for expressing object characterizations. Building on this language, the so-called *Refinement Machine* will be defined. The Refinement Machine owes its name to the way minimal axiomatic extension is realized: axioms are in the form of index expressions which are strengthened by *refinement*. In this section, the strict inference mechanism of the Refinement Machine is elucidated; its plausible inference mechanism will be discussed in Section 6.

4.1 Rules of Strict Inference

The strict inference mechanism of the Refinement Machine is based on a single rule of inference called *modus continens* which will be discussed in detail. Besides this rule, two closely related rules, called *modus generans* and *modus substituens*, will be introduced; although these rules are no part of the strict inference mechanism of the Refinement Machine at present, they constitute a point of departure for further research.

The rule of strict inference called *modus continens* may be looked upon as deduction by way of containment. To illustrate the basic idea, we consider the following example. Suppose a given information object has the index expression **pollution of rivers** as an axiom. From this index expression we can see that the object is about **pollution**, because the information conveyed by **pollution** is also inherent in **pollution of rivers**; a similar observation holds for **rivers**. *Modus continens* is formally defined as follows.

Definition 4.1 *Let i and j be index expressions in a language $\mathcal{L}(T,C)$ and let $\trianglelefteq$ be the is-subexpression-of relation over $\mathcal{L}(T,C)$. Then, if j is an index subexpression of i, j can be derived from i, or:*

$$j \trianglelefteq i \;\Rightarrow\; i \vdash_{MC} j$$

This rule of inference is called modus continens *and is denoted by MC.*

Note that there is an analogy here with *modus ponens.*

The intuition behind *modus generans* is deduction by way of generalization. The basis of this rule of inference are generalizations captured in the form of an ISA-relation. For example, given the generalization **salmon** ISA **fish**, *modus generans* affirms that any information object that deals with **salmon** also implicitly deals with **fish**.

Definition 4.2 *Let $\mathcal{L}(T,C)$ be a language of index expressions and let $i,j \in \mathcal{L}(T,C)$. Let* ISA $\subseteq T \times T$. *If i* ISA *j, then j can be inferred from i, or:*

$$i \text{ ISA } j \;\Rightarrow\; i \vdash_{MG} j$$

This rule of inference is called modus generans *and is denoted by MG.*

The ISA-relation is quite common in frame-based knowledge-representation languages [LvdG91]. Note that it brings domain knowledge into play within the Refinement Machine. Unfortunately, the ISA-relation cannot typically be derived automatically. We observe that care must be exercised when using *modus generans* due to homonyms. For example, the generalization **crane** ISA **bird** can only be exploited if the context is avian and not building construction.

The third rule of inference, called *modus substituens*, drives deduction by way of substitution. Recall from a previous example that **pollution** is deducible from **pollution of rivers** by *modus continens*. Now *modus substituens* may be used for example to conclude that any object that is about the **effects of** POLLUTION OF RIVERS **in Australia** is also about the **effects of** POLLUTION **in Australia.**

Definition 4.3 *Let k and i be index expressions in a language $\mathcal{L}(T,C)$ such that i is an index subexpression of k. Furthermore, let k_i^j be the index expression k with i substituted by j. Then,*

$$i \vdash j \;\Rightarrow\; k \vdash_{MS} k_i^j$$

This rule of inference is called modus substituens *and is denoted by MS.*

The general idea of *modus substituens* is schematically represented in Figure 7.

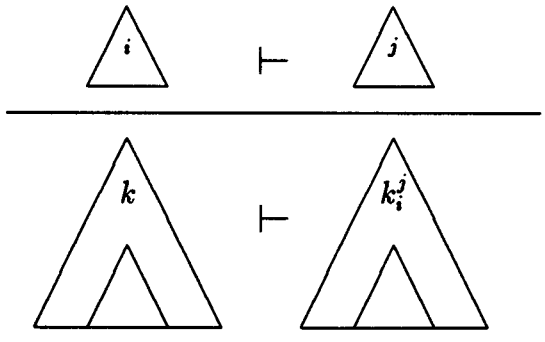

Figure 7: *Modus Substituens*

Note that *modus substituens* provides for context-free substitution. The disadvantage of this approach is well documented in the work of Chomsky. Within the realms of the Refinement Machine the problem manifests itself in the form of spurious index expressions. For example, from the index expression **effects of pollution of rivers in Australia** in the above example, **effects of rivers** can also be derived. However, it is highly unlikely that the object actually deals with this subject. Recently, attempts have been made at restricting conditions under which substitution can take place so as to prevent the occurrence of wild substitutions. A. Rosing [Ros91], for example, proposes that a term should only be substituted by one of its generalizations:

$$i \text{ ISA } j \;\Rightarrow\; k \vdash_{MS} k_i^j$$

Another approach is to allow substitutions to take place only within the context of a so-called term sequence: a term sequence is a sequence of expressions involving only the null connector. If one considers a term sequence as describing a particular context, then this context can often be implicitly described by any term subsequence which contains the last term. For example, the information conveyed by the expression **little o green o martians** is also implicitly contained in **green o martians**. Exploiting this observation to define a restricted form of *modus substituens*, the following is a valid derivation:

$$\textsf{invasion of LITTLE o GREEN o MARTIANS} \vdash \textsf{invasion of GREEN o MARTIANS}$$

Note that this restricted *modus substituens* provides a more context-sensitive notion of substitution.

4.2 The Lithoid Revisited

In the previous, three rules of strict inference have been introduced. Note that all rules apply to a single index expression, yielding a single index expression as well. From this observation, it follows that a strict derivation takes the form of a sequence of transformations on an index expression transforming it into another one. An immediate consequence is that the relevance of an object with respect to a given request can be established by deriving the request from a *single* index expression in the characterization of the object. This property is stated more formally in the following theorem, which is known as the *hook theorem*, signifying that a single characterization of an object acts as a 'hook' for its retrieval.

Theorem 4.1 *Let $\mathcal{L}(T,C)$ be a language of index expressions and let $q \in \mathcal{L}(T,C)$. If for some object O we have $\chi(O) \vdash q$, then there is an index expression $i \in \chi(O)$ such that $i \vdash q$.*

From this property, it will be evident that the lithoid constructed from a core set of index expressions constitutes all index expressions derivable from these index expressions by the Refinement Machine with *modus continens* for the only rule of strict inference.

5 Context-Free Plausible Inference

In Section 2 it has been mentioned that the plausible inference mechanism of an Information Disclosure Machine can be founded on the principle of minimal axiomatic extension. Here, we elaborate on this observation and take a closer look at strengthening for the purpose of plausible inference. In doing so, we assume that the axioms of an object characterization have the form of index expressions.

5.1 Rules of Context-Free Plausible Inference

The axioms of an object characterization can be strengthened by an operation called *refinement*. Informally speaking, refining an index expression is making it more specific. Refinement is typically achieved by adding a *connector-term* pair to a given index expression. For example, consider the index expression **information**. This expression can be refined into the index expression **need of information** which can in turn be refined into **people in need of information**; these refinements can be better understood by considering the lithoid depicted in Figure 8.

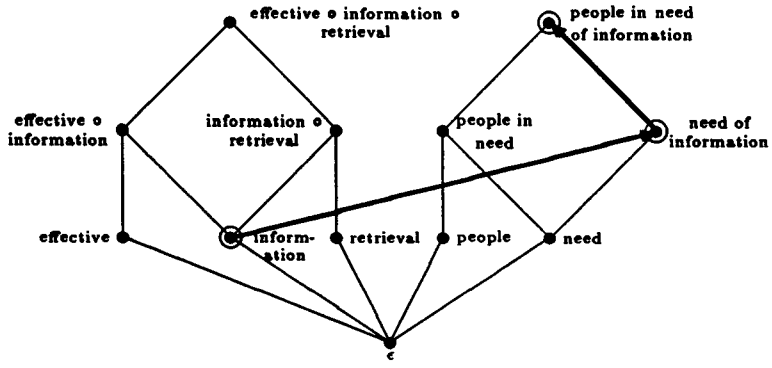

Figure 8: Refining Index Expressions

The refinements in this example are a direct result of the inverse $\trianglelefteq$-relation over the language of index expressions and therefore are in turn determined by the underlying lithoid. However, refinement can also be defined via the inverse of the ISA-relation between terms. For example, the expression **fish** can be refined into **salmon**. More formally, refinement is defined as follows.

Definition 5.1 *Let i and j be index expressions in a language $\mathcal{L}(T, C)$. We say that i can be* refined *into j, denoted as $i \twoheadrightarrow j$, if and only if one of the following conditions applies:*

1. *$i \trianglelefteq j$ and for all index expressions k such that $i \trianglelefteq k \trianglelefteq j$ it follows that $k = j \vee k = i$*
2. *i ISA j and for all index expressions k such that i ISA k ISA j it follows that $k = j \vee k = i$*

Now observe that refinement can be taken as the basis for plausible inference: by applying refinement inference is rendered plausible in the sense that the derivations involve index (sub)expressions that are no part of the original characterization of an object and are not deducible by strict inference.

The first rule of plausible inference we discuss is based on the notion of refinement as introduced above. It is called *plausible inference by refinement.* The general idea of this rule is as follows. Assume for example that within the set of information objects characterized by **pollution** there are objects that deal with the **pollution of rivers**. On the basis of this, the index expression **pollution of rivers** may be derived from the index expression **pollution**. For the moment we will not consider the certainty of this derivation, but merely observe that it is plausible.

Definition 5.2 *Let i and j be index expressions in a language $\mathcal{L}(T, C)$. Then, if i can be refined into j, j can be plausibly derived from i, or:*

$$i \twoheadrightarrow j \;\Rightarrow\; i \vdash_{PR} j$$

This rule of inference is called plausible inference by refinement *and is denoted by PR.*

Note that *plausible inference by refinement* applies to a single index expression as was true for the rules of strict inference discussed in the previous section. It therefore drives a context-free type of inference. The same observation holds for the second rule of plausible inference, called *plausible substitution.* This rule bears a strong resemblance to *modus substituens.*

Definition 5.3 *Let k and i be index expressions in a language $\mathcal{L}(T,C)$ such that i is an index subexpression of k. Furthermore, let k_i^j be the index expression k with i substituted by j. Then,*

$$i \twoheadrightarrow j \quad \Rightarrow \quad k \mathrel{\sim\!\!\vdash}_{PS} k_i^j$$

This rule of inference is called plausible substitution *and is denoted by PS.*

Together with the rules of strict inference, *plausible inference by refinement* and *plausible substitution* can provide the driving mechanism with which index expressions can be plausibly derived from others. For example, the index expression **metals** can be derived from **pollution of rivers** as follows:

$$\begin{aligned} \textbf{pollution of rivers} \quad & \vdash_{MC} \\ \textbf{pollution} \quad & \mathrel{\sim\!\!\vdash}_{PR} \\ \textbf{pollution from metals} \quad & \vdash_{MC} \\ \textbf{metals} \quad & \end{aligned}$$

The following example demonstrates the use of *plausible substitution*:

$$\begin{aligned} \textbf{effects of } \textsc{pollution of rivers} \quad & \vdash_{MC} \\ \textbf{effects of } \textsc{pollution} \quad & \mathrel{\sim\!\!\vdash}_{PS} \\ \textbf{effects of } \textsc{pollution from metals} \quad & \vdash_{MC} \\ \textbf{effects of } \textsc{metals} \quad & \end{aligned}$$

Note that in the last example only a single plausible inference step is involved in the derivation of **effects of metals** from **effects of pollution of rivers**. This means that these expressions have a fairly high degree of similarity. So, if **effects of metals** is a request, and an information object O is characterized by **effects of pollution of rivers**, then it is fairly likely that O would be relevant.

5.2 Problems with Context-Free Plausible Inference

After having introduced two rules of plausible inference, we address the question how adequate these rules are. Consider three information objects O_1, O_2 and O_3. Object O_1 is about river pollution in Australia, object O_2 is about the effects of pollution in rivers and the third object, O_3, deals with air pollution in Holland. Assume that these objects have the following characterizations:

$$\begin{aligned} \chi(O_1) &= \{\textbf{river} \circ \textbf{pollution in australia}\} \\ \chi(O_2) &= \{\textbf{effects of pollution in river}\} \\ \chi(O_3) &= \{\textbf{air} \circ \textbf{pollution in holland}\} \end{aligned}$$

For the sake of the argument, the characterization of the second information object specifies the term **river** instead of the term **rivers**.

Now suppose that the request is **river** $\circ$ **pollution**. Intuitively, objects O_1 and O_2 would seem to be relevant with respect to this request whereas O_3 is not. Furthermore, imagine that this request is fed into an Information Disclosure Machine whose inference mechanism is driven by the rules of inference defined in the previous sections. Formally, the machine has the rules

of strict inference $S = \{MC, MG, MS\}$ and the rules of plausible inference $P = \{PR, PS\}$. We now unleash this machine and try to derive the request from the characterizations of the three objects. Object O_1 can be shown to be relevant by application of *modus continens*:

$$\textsf{river o pollution in australia} \vdash_{MC} \textsf{river o pollution}$$

As it is not possible to strictly derive the request from the characterization of object O_2, plausible inference is brought to bear:

$$\begin{aligned} \textsf{effects of pollution in river} &\vdash_{MC} \\ \textsf{pollution} &\mathrel{\sim}_{PR} \\ \textsf{river o pollution} & \end{aligned}$$

Since the above derivation involves only a single plausible inference step it may be concluded that the probability of relevance of object O_2 to the request q is fairly high. Considering the situation, this conforms with our intuition. *However* the following derivation also only involves one plausible inference step:

$$\begin{aligned} \textsf{air o pollution in holland} &\vdash_{MC} \\ \textsf{pollution} &\mathrel{\sim}_{PR} \\ \textsf{river o pollution} & \end{aligned}$$

This in fact means that on the basis of the above derivations the Information Disclosure Machine employed would assess the probability of relevance of the object dealing with air pollution in Holland, that is, the relevance of object O_3, as being the same as that of the object dealing with the effects of pollution in rivers, that is, of object O_2; furthermore, this probability would be assessed as being fairly high. In other words, in addition to the objects O_1 and O_2, also object O_3 would be returned by this Disclosure Machine in response to the request **river o pollution**. Clearly, the Disclosure Machine is too blunt: it is unable to distinguish between object O_2 which is very likely to deal with river pollution and object O_3 which clearly does not. Figure 9 schematically depicts the above derivations in terms of the underlying lithoid; the index expressions involved have been abbreviated for the sake of clarity.

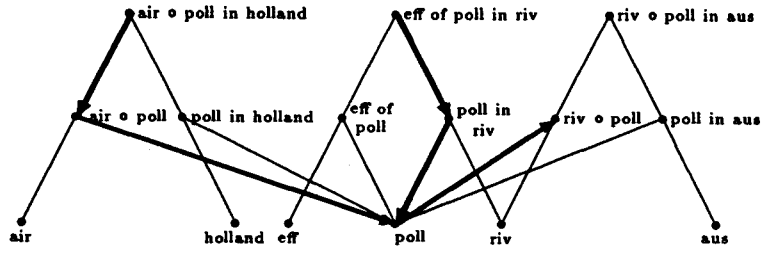

Figure 9: A Schematic Representation of Two Plausible Derivations

The reason for the inadequacy of an Information Disclosure Machine based on the rules of inference described above, lies in the fact that in applying these rules major parts of the

context provided by the initial characterization of an information object are discarded. For example, in the derivation

$$\textsf{effects of pollution in river} \vdash_{MC} \textsf{pollution}$$

the context mentioning the pollution being in rivers is thrown away and therefore cannot be used further in the derivation.

6 The Refinement Machine: Rules of Plausible Inference

Recall that the intention is to define the inference mechanism for the Refinement Machine. So far, a strict inference mechanism has been proposed in Section 4. It remains to define the plausible inference mechanism of the machine. It will be evident from the previous section that the Refinement Machine should not be based on context-free plausible inference. In this section, we present a new approach to plausible inference based on so-called belief networks. These have emerged from research into the representation and manipulation of uncertain information in knowledge-based systems; for an overview of this research area, we refer the reader to [LvdG91]. The present section is organized as follows. In Section 6.1 a brief introduction is given to belief networks as a general formalism for probability-based plausible reasoning. Section 6.2 features how to build a belief network from a lithoid of index expressions. Note that even though belief networks are based on probability theory and hence implicitly involve negation, this can be handled without adopting a Closed World Assumption in the sense as described in Section 2.

6.1 Introduction to Belief Networks

Informally speaking, a belief network is a graphical representation of a problem domain depicting the statistical variables discerned in the domain and their probabilistic interrelationships. These interrelationships are quantified by means of conditional probabilities. Several algorithms have been devised to efficiently compute probabilities of interest from such a representation. Before introducing the belief network formalism in Section 6.1.2, we provide some preliminaries. Section 6.1.3 describes reasoning with a belief network.

6.1.1 Preliminaries from Probability Theory

In providing some preliminaries from probability theory, we take an algebraic point of view and start with a brief review of the notion of a Boolean algebra. A *Boolean algebra* $\mathcal{B}$ is a set of elements with two binary operations $\wedge$ (conjunction) and $\vee$ (disjunction), a unary operation $\neg$ (negation) and two constants *false* and *true* which (by equality according to logical truth tables) adhere to the usual axioms. A subset of (algebraically independent) elements $\mathcal{V} = \{v_1, \ldots, v_n\}$, $n \geq 1$, of a Boolean algebra $\mathcal{B}$ is said to be a set of *generators* for $\mathcal{B}$ if each element of $\mathcal{B}$ can be represented in terms of the elements $v_i \in \mathcal{V}$, $i = 1, \ldots, n$, and the operations $\wedge$, $\vee$ and $\neg$. We will use $\mathcal{B}(v_1, \ldots, v_n)$ to denote the Boolean algebra $\mathcal{B}$ generated by $\mathcal{V}$. In the sequel, we will often take the point of view of a Boolean algebra $\mathcal{B}(v_1, \ldots, v_n)$ being 'spanned' by a set of variables V_i, $i = 1, \ldots, n$, each taking values from $\{v_i, \neg v_i\}$ where v_i is taken to represent $V_i = true$ and $\neg v_i$ denotes $V_i = false$. From now on, such a variable will be called a *probabilistic variable.*

Now, let $V = \{V_1, \ldots, V_n\}$ be the set of probabilistic variables spanning a Boolean algebra $\mathcal{B}(v_1, \ldots, v_n)$ and let V' be a subset of V with $m \geq 1$ variables. A conjunction of length m in which for each $V_i \in V'$ either v_i or $\neg v_i$ occurs is called a *configuration* of V'. For example, the conjunction $\neg v_1 \wedge v_2 \wedge \neg v_5$ is a configuration of the set $V' = \{V_1, V_2, V_5\}$. The conjunction of length m in which each probabilistic variable $V_i \in V'$ is named only, that is, specified without its value, is called a *configuration template* of V'. For example, the configuration template of $V' = \{V_1, V_2, V_5\}$ is $V_1 \wedge V_2 \wedge V_5$. Note that the configuration $\neg v_1 \wedge v_2 \wedge \neg v_5$ mentioned above can be obtained from the configuration template $V_1 \wedge V_2 \wedge V_5$ by filling in $\neg v_1$, v_2 and $\neg v_5$ for the variables V_1, V_2 and V_5 respectively.

On a Boolean algebra $\mathcal{B}(v_1, \ldots, v_n)$ we define a joint probability distribution. As it can easily be shown that the probability of an event is equivalent to the probability of the truth of the proposition asserting the occurrence of the event, it follows that a joint probability distribution on a Boolean algebra has the usual properties. In addition, conditional probabilities are defined as customary.

To conclude these preliminaries, the notion of an independency relation between probabilistic variables is introduced.

Definition 6.1 *Let $\mathcal{B}(v_1, \ldots, v_n)$, $n \geq 1$, be a Boolean algebra and let Pr be a joint probability distribution on $\mathcal{B}(v_1, \ldots, v_n)$. Let $V = \{V_1, \ldots, V_n\}$ be a set of probabilistic variables spanning $\mathcal{B}(v_1, \ldots, v_n)$. Now, let $X, Y, Z \subseteq V$ be sets of variables and let C_X, C_Y and C_Z be the configuration templates for the sets X, Y and Z, respectively. The set of variables X is said to be* conditionally independent *of Y given Z if $Pr(C_X | C_Y \wedge C_Z) = Pr(C_X | C_Z)$.*

For further details, the reader is referred to [vdG90].

6.1.2 The Belief Network Formalism

The notion of a belief network is introduced informally before a formal definition is given. It has been mentioned before that belief networks provide a formalism for representing a problem domain, or to be more precise, a joint probability distribution on a domain. To this end, a belief network comprises two parts: a *qualitative representation* and a *quantitative representation* of the domain. The qualitative part of a belief network takes the form of an acyclic digraph $G = (V(G), A(G))$ with vertices $V(G) = \{V_1, \ldots, V_n\}$, $n \geq 1$, and arcs $A(G)$. Each vertex V_i in $V(G)$ is taken to represent a probabilistic variable. An arc $(V_i, V_j) \in A(G)$ is taken to represent a direct 'influential' relationship between the linked variables V_i and V_j; absence of an arc between two vertices means that the corresponding variables do not influence each other directly. Associated with the graphical part of a belief network is a numerical assessment of the 'strengths' of the represented relationships: with each vertex is associated a set of (conditional) probabilities which describe the influence of the values of the predecessors of the vertex on the values of the vertex itself.

We now define the notion of a belief network more formally.

Definition 6.2 *A* belief network *is a tuple $B = (G, \Gamma)$ where*

1. *$G = (V(G), A(G))$ is an acyclic digraph with vertices $V(G) = \{V_1, \ldots, V_n\}$, $n \geq 1$, and*
2. *$\Gamma = \{\gamma_{V_i} | V_i \in V(G)\}$ is a set of real-valued nonnegative functions $\gamma_{V_i} : \{v_i, \neg v_i\} \times \{c_{\pi(V_i)}\} \rightarrow [0, 1]$, called (conditional probability)* assessment functions, *such that for each configuration $c_{\pi(V_i)}$ of the set of parents $\pi(V_i)$ of V_i in G we have that $\gamma_{V_i}(\neg v_i | c_{\pi(V_i)}) = 1 - \gamma_{V_i}(v_i | c_{\pi(V_i)})$, $i = 1, \ldots, n$.*

Note that in the previous definition V_i is viewed as a vertex from the graph and as a probabilistic variable, alternatively.

In order to link the qualitative and quantitative parts of a belief network, a probabilistic meaning is assigned to the topology of the digraph of the network. We do note elaborate on the probabilistic meaning of the graphical part of a belief network in detail. For the purpose of this paper, it suffices to introduce this meaning only informally: two sets of variables X and Y are taken to be conditionally independent given a third set of variables Z if all paths in the underlying graph of the digraph from a vertex in X to a vertex in Y are blocked by a vertex in Z. Further details may be found in [Pea88].

The following theorem now states that the initial assessment functions of a belief network provide all information necessary for uniquely defining a joint probability distribution on the variables discerned that respects the independency relationships portrayed by the graphical part of the network. Henceforth, we will call this the joint probability distribution defined by the network.

Theorem 6.1 *Let $B = (G, \Gamma)$ be a belief network where $V(G) = \{V_1, \ldots, V_n\}$, $n \geq 1$. Let $\mathcal{B}(v_1, \ldots, v_n)$ be the free Boolean algebra spanned by $V(G)$. Then,*

$$Pr(C_{V(G)}) = \prod_{V_i \in V(G)} \gamma_{V_i}(V_i | C_{\pi(V_i)})$$

defines a joint probability distribution Pr on $\mathcal{B}(v_1, \ldots, v_n)$ that respects the independency relation portrayed by G.

Note that since the conditional probability assessment functions of a belief network uniquely define a joint probability distribution on the Boolean algebra generated by the vertex set of the graph, any probability of interest can be computed from these functions.

6.1.3 Reasoning with a Belief Network

In the previous section, the notion of a belief network was introduced as a means for representing a joint probability distribution. For making probabilistic statements concerning the variables discerned in the problem domain, two algorithms are associated with a belief network:

- an algorithm for (efficiently) computing probabilities of interest from the network, and
- an algorithm for processing evidence, that is, for entering evidence into the network and subsequently (efficiently) computing the revised probability distribution given the evidence; this process is generally called *evidence propagation.*

We have mentioned before that any probability of interest can be computed from the conditional probability assessment functions. Equally, the impact of a piece of evidence concerning a specific variable, on each of the other variables can be computed from these functions. Now, observe that the conditional probability assessment functions describe the joint probability distribution locally for each vertex and its predecessors. Calculation of a (revised) probability from the joint probability distribution defined by the assessment functions in a straightforward manner, however, will generally not be restricted to performing computations which are local in terms of the graphical part of the belief network. In the literature therefore, several

less naive algorithms for computing probabilities of interest from a belief network and for processing evidence in the network have been proposed. The most well-known is the set of algorithms presented by J. Pearl [Pea88]. The basic idea of these algorithms is that the topology of the graph of a belief network is exploited as a computational architecture. The vertices of the graph are taken as autonomous objects having a local processor capable of performing certain probabilistic computations and a local memory in which the associated conditional probability assessment function is stored; the arcs of the graph are viewed as (bi-directional) communication channels through which the objects can send each other messages. Updating the joint probability distribution and computing local probabilities essentially entails each probabilistic variable, that is, each vertex, combining its own local information with messages it receives from its neighbours providing it with further information about the joint probability distribution. Another set of elegant algorithms based on the statistical theory of Markov random fields, has been proposed by S.L. Lauritzen and D.J. Spiegelhalter [LS88].

Although all algorithms proposed for evidence propagation are based on probability theory, they differ considerably with respect to the algorithms employed and their complexity. It should be noted that in general probabilistic inference in belief networks without any restrictions is NP-hard [Coo90]. However, only small restrictions on the topology of the graphical part of the belief network suffice to render the schemes mentioned above polynomial in the number of variables discerned.

6.2 An Index Expression Belief Network

In Section 3, it has been mentioned that the lithoid is a useful structure for information disclosure as it can be viewed by the searcher as a conceptual space through which he or she can browse in order to locate index expressions which they feel are good descriptions of their information need; this process was referred to earlier as *query by navigation*. In this section, we further exploit the lithoid as the basis of the inference mechanism of the Refinement Machine. To this end, a belief network of index expressions is generated from the lithoid. We recall from the previous that a belief network comprises both a qualitative and a quantitative representation of a problem domain. These components will be discussed separately in the Sections 6.2.2 and 6.2.3. Before doing so, however, we show that within a probabilistic context negation can be introduced without adopting a Closed World Assumption in the sense described in Section 2. The section will be concluded by a discussion as to how the resulting belief network and associated algorithms constitute the inference mechanism of the Refinement Machine.

6.2.1 Introducing Negation

In information disclosure, one way of bringing negation into play is to adopt a Closed World Assumption, stating that if a request q is not deducible from an object characterization $\chi(O)$ by the strict inference mechanism, then the object O is assumed *not* to be relevant with respect to q. As was argued in Section 2, this can be a dubious assumption. This observation is further illustrated by an example. Consider an information object O which is characterized by the index expression **pollution of rivers** only. The request **river o pollution** cannot be proven from this characterization using the strict rules of inference. An Information Disclosure Machine operating under a Closed World Assumption therefore would conclude that O is not relevant with respect to this request. Intuitively, however, O would seem to have a high probability

of relevance as the expressions pollution of rivers and river o pollution are very similar: our belief in the relevance of the object with respect to the request is directly proportional to the similarity between the index expression in its characterization and the request.

Probability theory provides a means to introduce negation and at the same time avoid a Closed World Assumption. The notion of negation subsistent in probability theory further yields a suitable, mathematically sound mechanism for discriminating between degrees of relevance. We assume a joint probability distribution Pr being defined on a set of index expressions indicating prior probabilities of being associated with a relevant object. The probability of relevance of an object O with respect to a given request q can be equated with the probability of q given the context described by the characterization of O. Analogous to the hook theorem, which states that a request can be proven from a single index expression in an object's characterization, we adopt the view that the probability of relevance of an object is the maximal conditional probability generated from individual index expressions in its characterization. The probability of relevance is defined more formally in the following definition.

Definition 6.3 *Let $\mathcal{L}(T,C)$ be a language of index expressions. Let O be an object, let $\chi(O) \subseteq \mathcal{L}(T,C)$ be its characterization and let $q \in \mathcal{L}(T,C)$ be a request. Furthermore, let Pr be a joint probability distribution defined on $\mathcal{L}(T,C)$. Then, the* probability of relevance *of O with respect to q, denoted by $P_{\text{Rel}}(O,q)$, is defined as*

$$P_{\text{Rel}}(O,q) \;=\; max\{Pr(q|i) \mid i \in \chi(O)\}$$

Note that an alternative definition of the probability of relevance would be to take the entire characterization of an object as evidence, that is, to take $P_{\text{Rel}}(O,q)$ defined as

$$P_{\text{Rel}}(O,q) \;=\; Pr(q|\chi(O))$$

This alternative definition will yield a different behaviour of the resulting Disclosure Machine; it will be a matter of experimentation to decide which definition is more appropriate in practice.

The probabilistic approach indicated above is combined with the strict inference mechanism of the Refinement Machine as defined in Section 4 in the following sense. Consider an object O and a given request q. If q can be deduced from an index expression $i \in \chi(O)$ of O by strict inference, then the probability of q given i is maximal, that is,

$$i \vdash q \;\Rightarrow\; Pr(q|i) = 1$$

So, in this case the probability of relevance of O with respect to q equals 1. Conversely, only if $Pr(q|i) = 0$ for all $i \in \chi(O)$ does the Refinement Machine conclude that O is not relevant with respect to q.

6.2.2 The Qualitative Part of the Belief Network

Building a belief network of index expressions is now addressed. Recall from the previous section that the qualitative part of a belief network is an acyclic digraph. The vertices in this digraph are taken to represent probabilistic variables and the arcs are taken as influential relationships between these variables. In constructing the qualitative part of a belief network of index expressions, therefore, the probabilistic variables involved and the relationships between these variables have to be identified.

In Section 3.3 we have introduced the notion of a power index expression and have indicated that the union of the power index expressions for a given core set of index expressions forms a lithoid. The searcher exploits this lithoid for information disclosure by moving across it refining or enlarging a current focus. For a specific search some of the contexts in the lithoid are possibly relevant and some are not in the sense of the negation introduced in the previous. Therefore, we interpret each of the vertices of the lithoid to define a probabilistic variable. In order to distinguish between the variables defined by the index expressions in the lithoid and the index expressions themselves, we will adhere to the notational convention introduced in Section 6.1: for example, POLLUTION OF RIVERS denotes the probabilistic variable taking a value from the set of values {pollution of rivers, ¬pollution of rivers}.

Now recall that the index expressions in a power index expression are partially ordered by the is-subexpression-of relation $\trianglelefteq$. It follows that the probabilistic variables we have defined are partially ordered as well. Therefore, the lithoid can be taken to define the (undirected) topology of the graphical part of a belief network. The edges of the resulting undirected graph indicate the partial ordering on the probabilistic variables discerned. These edges are assigned a direction using the inverse $\trianglelefteq$-relation since for the purpose of information disclosure we are interested, for example, in the probability of the index expression pollution of rivers given the separate terms pollution and rivers. From $\trianglelefteq$ being a partial ordering the digraph resulting from the transformation described above is guaranteed to be acyclic. Finally, we note that in constructing the digraph from the lithoid the empty index expression may be omitted as it is not information bearing to the disclosure. As an example consider the lithoid depicted in Figure 6; the corresponding digraph constructed from this lithoid is shown in Figure 10.

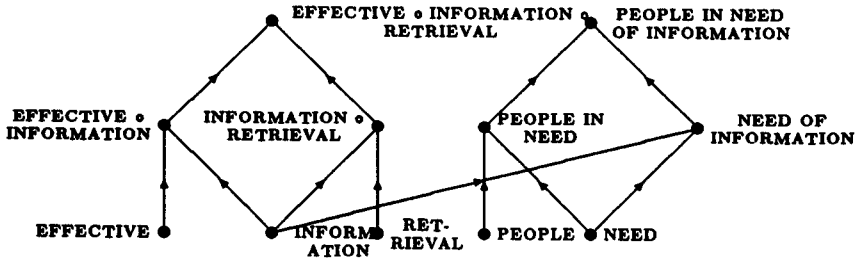

Figure 10: An Example Digraph

6.2.3 The Quantitative Part of the Belief Network

In the previous subsection, the qualitative part of a belief network of index expressions has been constructed. In this section this representation is completed by a set of conditional probability assessment functions quantifying the strengths of the relationships between the probabilistic variables defined. Different assessment functions will be presented for different types of vertices in the digraph.

We begin by looking at vertices having an in-degree equal to zero. For the variables corresponding with these vertices we have to specify prior probabilities on the values such a variable can take. From the associated lithoid we see that these variables correspond to unary index expressions, or terms. In the context of information disclosure relative to a given set of information objects $\mathcal{O}$, it is reasonable to assume that a term that occurs frequently

has a higher probability of being in a relevant object than a term that occurs infrequently. The prior probabilities on the values of a term variable may therefore be computed from the occurrence frequency of the term relative to the set of objects $\mathcal{O}$. That is, for a variable T for a term t we compute the value $\gamma_T(t)$ of the assessment function γ_T associated with T using

$$\gamma_T(t) \approx \eta f(t)$$

where η is some normalizing factor. Note that the complementary function value $\gamma_T(\neg t)$ can be computed using the equality $\gamma_T(\neg t) = 1 - \gamma_T(t)$. This approach to estimating the probability of occurrence of a term is common in information retrieval [WY90].

We now turn to vertices for which the set of parents consists of vertices with in-degree zero. These vertices correspond with variables that represent binary index expressions which are constructed from two terms via the addition of a connector. For example, the binary index expression **pollution of rivers** is constructed from the terms **pollution** and **rivers** by adding the connector **of**; the graphical part of a belief network comprising variables for these index expressions only is depicted in Figure 11. We use this example to define the conditional probability assessment function for the variable `POLLUTION OF RIVERS`; for the sake of brevity, we use `P OF R` to denote `POLLUTION OF RIVERS`.

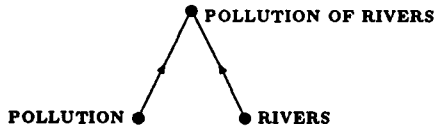

Figure 11: A Simple Belief Network

For the conditional probability assessment function $\gamma_{\texttt{P OF R}}$ we have to specify eight values: the four function values

$$\begin{aligned}
\gamma_{\texttt{P OF R}}(\textsf{pollution of rivers}|\textsf{pollution} \wedge \textsf{rivers}) &= w \\
\gamma_{\texttt{P OF R}}(\textsf{pollution of rivers}|\neg\textsf{pollution} \wedge \textsf{rivers}) &= x \\
\gamma_{\texttt{P OF R}}(\textsf{pollution of rivers}|\textsf{pollution} \wedge \neg\textsf{rivers}) &= y \\
\gamma_{\texttt{P OF R}}(\textsf{pollution of rivers}|\neg\textsf{pollution} \wedge \neg\textsf{rivers}) &= z
\end{aligned}$$

and the complementary ones. Now consider the function value

$$\gamma_{\texttt{P OF R}}(\textsf{pollution of rivers}|\textsf{pollution} \wedge \textsf{rivers}) = w$$

In terms of information disclosure, this function value has the following meaning: given that we know that an object O is about **pollution** and that we know that O is about **rivers**, then the probability that O is about **pollution of rivers** equals w. Later in this section we will discuss a method for computing suitable probability estimates based on frequency analyses of connectors in binary index expressions. First, however, we consider the function values x, y and z indicated above and show that these values are necessarily equal to zero.

Theorem 6.2 *Let $B = (G, \Gamma)$ be a belief network. For each triple of probabilistic variables $I, J, K \in V(G)$ such that $(I, K), (J, K) \in A(G)$, we have that*

$$\gamma_K(k|I \wedge J) = 0$$

for $I = \neg i$ or $J = \neg j$.

Proof: Let Pr be the joint probability distribution defined by the belief network B as in Section 6.1. From the construction of the digraph G and $(I, K) \in A(G)$ we have that $i \trianglelefteq k$. Therefore, $k \vdash i$ which implies $Pr(i|k) = 1$. Analogously, we find $Pr(j|k) = 1$. From $Pr(i|k) = 1$ and $Pr(j|k) = 1$, we have $Pr(i \wedge j|k) = 1$. Using marginalization, $Pr(i \wedge j|k) = 1$ implies $Pr(i \wedge \neg j|k) + Pr(\neg i \wedge j|k) + Pr(\neg i \wedge \neg j|k) = 0$. Now observe that from $Pr(\neg i \wedge \neg j|k) = 0$ we find $Pr(k|\neg i \wedge \neg j) \cdot Pr(\neg i \wedge \neg j) = 0$. So, at least one of the probabilities $Pr(k|\neg i \wedge \neg j)$ and $Pr(\neg i \wedge \neg j)$ equals zero. We know that $Pr(\neg i \wedge \neg j) > 0$ because $Pr(\neg i \wedge \neg j) = 0$ would imply that there is no information object which is *not* about expression i and *not* about j, which clearly is not the case. Therefore, $Pr(k|\neg i \wedge \neg j) = 0$, and so $\gamma_K(k|\neg i \wedge \neg j) = 0$. Using similar arguments, we find $\gamma_K(k|i \wedge \neg j) = \gamma_K(k|\neg i \wedge j) = 0$.

□

Note that the theorem states a direct result of harbouring maximal belief in the consequences of the strict inference mechanism.

The question remains as to how the function value w can be obtained. One method is to analyze the frequencies of occurrence of connectors in binary index expressions. Recently such an analysis has been carried out [Ros91]. Index expressions were derived from the titles of the Cranfield document collection. Using these expressions, a lithoid was constructed as described in Section 3. An analysis of the index expressions in the resulting lithoid revealed that approximately fifteen percent of binary expressions involve the of connector. In our example, therefore, the value w can be approximated by 0.15, that is,

$$\gamma_{\text{P OF R}}(\text{pollution of rivers}|\text{pollution} \wedge \text{rivers}) \approx 0.15$$

For binary index expressions involving other connectors, the estimates in the (incomplete) table shown in Figure 12 can be used. For a full report on connector probabilities, the reader is referred to [Bru92]. Note that using these connector probabilities provides only rough estimates of the probabilities required.

Up to this point, we have considered variables that represent terms or binary index expressions and have defined associated conditional probability assessment functions for these variables. Attention will now be focussed on variables representing n-ary index expressions, $n \geq 3$. From the construction of a lithoid it will be evident that an n-ary index expression is formed by combining two index expressions of degree $n - 1$ that overlap in one term. For example, the ternary index expression **people in need of information** is constructed from the binary index expressions **people in need** and **need of information** on the basis of the term **need** appearing in both expressions. The analysis of the titles of the Cranfield document collection cited above revealed that if two index expressions of degree $n - 1$ combined into an n-ary index expression, $n \geq 3$, then they did so uniquely. So, for two index expressions i and j combining into an index expression k, the function value $\gamma_K(k|i \wedge j)$ therefore can be taken to be

$$\gamma_K(k|i \wedge j) = 1$$

Theorem 6.2 also applies here, so

$$\begin{aligned} \gamma_K(k|\neg i \wedge j) &= 0 \\ \gamma_K(k|i \wedge \neg j) &= 0 \\ \gamma_K(k|\neg i \wedge \neg j) &= 0 \end{aligned}$$

Connector	*Probability*
o	0.5366
and	0.0492
as	0.0004
at	0.0348
between	0.0052
by	0.0061
for	0.0327
from	0.0039
in	0.0632
of	0.1529
on	0.0370
or	0.0026
over	0.0066
through	0.0035
to	0.0170
with	0.0248

Figure 12: Some Connector Probabilities

Note that for larger document collections the assessment function given above may not be acurate. However, it is expected that for larger sets of information objects there equally exists some small value of n such that for probabilistic variables representing n-ary index expressions the probability assessment shown above is appropriate as well.

Now, for all variables discerned a conditional probability assessment function has been specified. The digraph constructed in the previous section and this set of assessment functions together define a belief network of index expressions. We conclude by observing that the approach presented differs from the one proposed H. Turtle and W.B. Croft [TC90], in the respect that in our approach the belief network exists purely within the realm of the descriptor language. Recently belief networks have also been investigated in conjunction with term phrases [CTL91].

6.3 The Index Expression Belief Network and Plausible Inference

Recall that the intention of Section 6 is to define the plausible inference mechanism of the Refinement Machine. The basic idea now is to take the index expression belief network built from a core set of index expressions as outlined before and its associated algorithms, to define a rule of plausible inference. Before defining this rule of inference we observe that the Refinement Machine restricted to the rule of strict inference *modus continens* is implicitly embedded in the index expression belief network; this follows from the topology of the graphical part of an index expression belief network being obtained from the lithoid which in turn defines the set of index expressions deducible via *modus continens* from a core set of index expressions, and the way the joint probability distribution on the index expressions is defined.

Plausible reasoning with the index expression belief network now is taken to define plausible inference in the Refinement Machine as illustrated by the following example. Consider once again the index expression belief network depicted in Figure 11. From the two separate terms **pollution** and **rivers**, the binary index expression **pollution of rivers** may be derived with

probability Pr(pollution of rivers|pollution $\wedge$ rivers). In terms of logic, this is equivalent to the plausible inference step

$$\text{pollution}, \text{rivers} \quad \sim\!\!\vdash \quad \text{pollution of rivers}$$

This inference step can be seen as a step in which the connector of is "guessed"; the strength of the guess is represented by the associated conditional probability. By generalizing this example, the plausible inference mechanism of the Refinement Machine is defined as follows.

Definition 6.4 *Let* $i_1, \ldots, i_n$, $n \geq 1$, *and* k *be index expressions in* $\mathcal{L}(T, C)$. *Then,*

$$Pr(k|i_1 \wedge \cdots \wedge i_n) > 0 \quad \Rightarrow \quad i_1, \ldots, i_n \sim\!\!\vdash_{PI} k$$

This rule of inference is termed plausible inference by probabilistic deduction *and is denoted by PI.*

In summary, the Refinement Machine Δ is defined as $\Delta = \langle D, \{MC\}, \{PI\} \rangle$ assuming the Disclosure Structure D.

7 Experimental Results

In the previous sections we have defined the Refinement Machine as a concretization of the Information Disclosure Machine. The effectiveness of this Refinement Machine is the theme of the present section. As a starting point for discussion of the effectiveness of the Refinement Machine, consider once again the example depicted in Figure 9 at the end of Section 5. The reader is reminded that this example was used to show how a Disclosure Machine with a context-free plausible inference mechanism would assign the same probability of relevance to a document about air pollution in Holland as to one about the effects of pollution in rivers in response to a request river o pollution.

The same problem will now be presented to the Refinement Machine as defined in the previous section. This entails that an index expression belief network is constructed from the core set of index expressions from the three information objects O_1, O_2 and O_3 having the following characterizations:

$$\begin{aligned} \chi(O_1) &= \{\text{river o pollution in australia}\} \\ \chi(O_2) &= \{\text{effects of pollution in river}\} \\ \chi(O_3) &= \{\text{air o pollution in holland}\} \end{aligned}$$

The graphical part of the resulting belief network is depicted in Figure 13; again, for reasons of clarity the names of the probabilistic variables have been abbreviated. The associated conditional probability assessment functions are defined as outlined before. Recall that the probability of a term was estimated using normalized occurrence frequencies. In this example, the term pollution occurs three times, once in each document, resulting in the prior probability Pr(pollution) = 0.33. The table shown in Figure 14 summarizes the term probabilities for this example. The conditional probability assessment functions for variables representing binary index expressions are defined using the normalized connector occurrence frequencies shown in Figure 12.

As a testing vehicle, the IDEAL system was used [SB90]. IDEAL is an environment for building and reasoning with belief networks. A number of evidence propagation algorithms

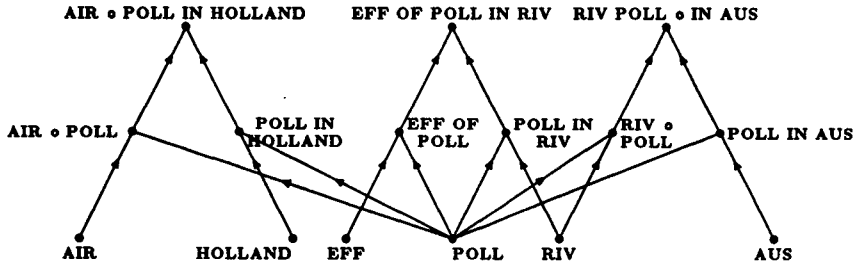

Figure 13: The Index Expression Belief Network for the Pollution Example

t	$f(t)$	$Pr(t)$
pollution	3	0.33
river	2	0.22
effects	1	0.11
australia	1	0.11
air	1	0.11
holland	1	0.11

Figure 14: Term Probabilities for the Pollution Example

are supported by IDEAL; in our case, the Lauritzen and Spiegelhalter algorithm was used. The experiment was run as follows. The index expression characterizing an object was entered as evidence into the belief network and subsequently propagated; thereafter the probability of the request was computed from the network. The results of this experiment for each of the objects are summarized in the table shown in Figure 15.

Evidence	$Pr(\text{river o pollution}\mid Evidence)$
effects of pollution in river	0.55
river o pollution in australia	1
air o pollution in holland	0.12

Figure 15: The Computed Probabilities of Relevance

Comparing these results with the results obtained for the same example in Section 5, we observe that they are encouraging. In contrast with a Disclosure Machine employing a context-free plausible inference mechanism, the Refinement Machine shows a substantial differentiation between $Pr(\text{river o pollution}\mid\text{effects of pollution in river})$ and $Pr(\text{river o pollution}\mid$ air o pollution in holland). Note that the probability

$$Pr(\text{river o pollution}\mid\text{river o pollution in australia}) \;=\; 1$$

is a consequence of the request river o pollution being strictly deducible from the index expression river o pollution in australia via *modus continens.*

8 Conclusions

In this paper the Refinement Machine has been presented as a concretization of an Information Disclosure Machine. The Refinement Machine features the language of index expressions

for characterizing information objects and provides both a strict and a plausible inference mechanism.

The work presented has been motivated by recent work on a logic-based approach to information disclosure. Although this approach is generally considered theoretically elegant, many questions regarding its feasibility has been raised. As to this respect, the Refinement Machine appears to be very promising for further investigation. This is because it brings together two facets which are (more or less) generally considered as being necessary for realizing significant improvements in disclosure effectiveness. On the one hand, there is the language of index expressions, which is more expressive than languages of terms or term phrases, and on the other hand, there is the disclosure aspect derived from the inference mechanism provided by the framework of belief networks.

As to its effectiveness, we are currently testing the Refinement Machine in the context of larger examples than the one presented in this paper. In addition, we would like to investigate enhanced Refinement Machines which include *modus generans* and *modus substituens*; the problem here is the efficient construction of the enhanced lithoid. There are also many open problems concerning the feasibility of the Refinement Machine. The lithoid which is the basis of the belief network, for example, can grow very large: the number of vertices may be exponential in the number of index expressions in the core set of index expressions generated from a given document collection. For larger object bases it is not feasible to store and reason with the entire resulting belief network. One of the results of the ESPRIT project APPED is an implementation of query by navigation which generates the relevant part of the lithoid dynamically. Future research will be directed towards developing a strategy for dynamic generation of relevant parts of a belief network to be incorporated into the Refinement Machine. Another area of interest is using the Refinement Machine to support interlayer navigation within stratified hypermedia [BvdW92]. We hope to communicate our further research results in future publications.

References

[BBB91] R. Bosman, R. Bouwman, and P.D. Bruza. The effectiveness of navigable information disclosure systems. In: G.A.M. Kempen, editor, *Proceedings of the Informatiewetenschap 1991 conference*, pages 55–69, 1991.

[BC89] C. Berrut and Y. Chiaramella. Indexing medical reports in a multimedia environment: the RIME experimental approach. In: *Proceedings of the Twelfth ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86, 1989.

[Bru90] P.D. Bruza. Hyperindices: A novel aid for searching in hypermedia. In: A.Rizk, N.Streitz, and J.Andre, editors, *Proceedings of the European Conference on Hypertext - ECHT 90*, pages 109–122, 1990.

[Bru92] P.D. Bruza. *Stratified Information Disclosure: A Synthesis between Information Retrieval and Hypermedia.* PhD thesis, University of Nijmegen, 1992 (to appear).

[BvdW90] P.D. Bruza and T.P. van der Weide. Two level hypermedia - an improved architecture for hypertext. In: A.M.Tjoa and R.Wagner, editors, *Proceedings of the*

Data Base and Expert System Applications Conference (DEXA 90), pages 76–83, 1990.

[BvdW92] P.D. Bruza and T.P. van der Weide. Stratified hypermedia structures for information disclosure. *The Computer Journal*, 1992 (to appear).

[Coo90] G.F. Cooper. The computational complexity of probabalistic inference using Bayesian belief networks. *Artificial Intelligence*, vol. 42, pages 393–405, 1990.

[Cra86] T.C. Craven. *String Indexing.* Academic Press, Inc, 1986.

[CTL91] W. Bruce Croft, H.R. Turtle, and D.D. Lewis. The use of phrases and structured queries in information retrieval. In: A. Bookstein, Y. Chiaramella, G. Salton, and V.V. Raghavan, editors, *Proceedings of the 14th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45, 1991.

[Far80] J. Farradane. Relational indexing part I. *Journal of Information Science*, vol. 1(5), pages 267–276, 1980.

[LS88] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *The Journal of the Royal Statistical Society*, vol. 50, pages 157–224, 1988.

[LvdG91] P.J.F. Lucas and L.C. van der Gaag. *Principles of Expert Systems.* Addison Wesley, 1991.

[Nie86] J. Nie. An outline of a general model for information retrieval systems. In: *Proceedings of the Ninth ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 495–506, 1986.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufman Publishers, Palo Alto, 1988.

[Rij86a] C.J. van Rijsbergen. A new theoretical framework for information retrieval. In: *Proceedings of the 9th International ACMSIGIR Conference on Research and Development in Information Retrieval*, pages 194–200, 1986.

[Rij86b] C.J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, vol. 29(6), pages 481–485, 1986.

[Rij89] C.J. van Rijsbergen. Towards an information logic. In: *Proceedings of the Twelfth ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86, 1989.

[Ros91] A. Rosing. *An Evaluation of the Hyperindex Machine.* Master's thesis, University of Nijmegen, 1991.

[Sal89] G. Salton. *Automatic Text Processing: The Translation, Analysis and Retrieval of Information by Computer.* Addison-Wesley Publishing Company, 1989.

[SB90] S. Srinivas and J. Breese. *IDEAL: Influence Diagram Evaluation and Analysis in Lisp, Documentation and Users Guide.* Technical Memorandum no. 23, Rockwell International Science Center, Palo Alto, 1990.

[SvR90] T.M.T. Sembok and C.J. van Rijsbergen. SILOL: A Simple Logical-Linguistic Document Retrieval System. *Information Processing and Management*, vol. 26(1), pages 111–134, 1990.

[TC90] H. Turtle and W. Bruce Croft. Inference networks for document retrieval. In: J.L. Vidick, editor, *Proceedings of the 13th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, 1990.

[vdG90] L.C. van der Gaag. *Probability-Based Models for Plausible Reasoning.* PhD thesis, University of Amsterdam, 1990.

[Wea88] M.T. Weaver. *A Frame-Based Language in Information Retrieval.* Technical Report TR88-25, Virginia Polytechnic, 1988.

[WY90] S.K.M. Wong and Y.Y. Yao. A probabilistic inference model for information retrieval. *Information Systems*, vol. 16(3), pages 301–321, 1990.