INFORMATION RETRIEVAL VIA THE ETS MODEL

by

David Clément

A Thesis Submitted in Partial Fulfillment of

the Requirements for the Degree of

Master of Computer Science

in the Graduate Academic Unit of Computer Science

THE UNIVERSITY OF NEW BRUNSWICK

March, 2003

Abstract

With the increasing amount of data appearing on the World Wide Web, the interest in new relevant models for information retrieval is growing. The purpose of this study was to introduce a new model that considers the *structure* of information. This model is based on a new mathematical formalism: the evolving transformation system (ETS) model developed by Goldfarb *et al.*.

A structural (ETS) representation of the personality traits of professors in Computer Science (relevant to the potential graduate students) based on their academic homepages is proposed. This ETS representation allows identifying classes of professors characterized by structural similarities in the chosen personality traits.

Since the model includes the structural information in the representations, it enables one to retrieve the relevant elements that appear quite different from the query. This would not be possible with any classical information retrieval models.

Acknowledgment

I would like to take this opportunity to thank my supervisors Dr Gerhard Dueck and Dr Lev Golfarb for their guidance, help, and encouragement throughout the work.

I also sincerely thanks the members of the ETS group and particularly Dave Gay and Oleg Golubitsky.

Partial funding for the research was provided by the Faculty of Computer Science, University of New Brunswick.

The work presented in this thesis has been performed in collaboration with Dr Goldfarb.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Background and motivation

Since the last decade, an important collection of knowledge has emerged. The World Wide Web (WWW) is actually the largest source of information for any search. This quantity of information is an advantage, but it is also an inconvenience: such an amount of data implies searching is more difficult. Moreover, the distributed architecture of the WWW implies heterogeneity of the format and of the nature of information.

Soon after the emergence of the WWW, to facilitate the search in this collection of data, some search engines appeared (AltaVista, Yahoo!, WebCrawler, and more recently Google). These services are provided by companies. The only source of profits for these companies comes from the sale of some advertisement space to other companies. The price of this space depends only on the number of visits to the site. These companies need a lot of people to visit their websites to benefit from their activity.

In contrast, the user uses only tools that can bring him or her some benefits. When someone uses a search engine, she or he will use only those that are the most effective for her or his need. His or her criteria for judging a search engine are generally the ease of use, the rapidity with which the query is treated, the precision of the response, and the comprehensiveness of the result.

Improving the quality of their tools is the main issue for the companies. Two threads are followed simultaneously: the first one is to collect all the possible information (The number of pages collected by Google exceeded three billion in January 2003) and the other one is to improve the quality of the retrieval.

Several models have been designed to get the best retrieval. All of these models are generally based on traditional mathematical models (including Boolean algebra, vector spaces, and theory of probability).

In all of the traditional models for information retrieval, the text is represented as a set of keywords or a set of ordered keywords. Clearly, the structure of the information in a document is also information, and one has to take it into account in order to have a reliable retrieval. This structure can be found at two levels: at the sentence level (the grammatical structure is important since the meaning of the sentence depends on the order of the words) and at an upper level (the level of the whole entity [such as paragraph or document depending on the precision of the information needed]).

The objective of this thesis is to present a first attempt at a *structural information retrieval* that is based on the ETS model, which is a new mathematical formalism developed by Goldfarb *et al.* [11]. The ETS model has initially been designed as a structural model for pattern recognition. Its advantages lie in the fact that the information is represented in a structural form and that the representation contains the constructive history of the "real" object. This model will be used in this thesis as a guide to represent the information.

1.2 Scope of the thesis

The purpose of this study is to show that the information structure is useful in the representation of a document to improve the effectiveness of the retrieval and that this cannot be achieved through classical models. A model based on the ETS formalism is applied to the corpus of academic homepages of professors in Computer Science. It is designed to allow a graduate student to select a supervisor. The objectives of this study address five aspects:

1. To review briefly the different models used in information retrieval,

2. To get a reasonable knowledge of the ETS model in order to apply it to an information retrieval problem,

3. To build an ETS representation of the professional profile of a professor based on his personal academic homepage. We have underestimated the complexity of this task. Even restricted to academic matters, developing a reasonable generative profile is a complex task for which I have not received any training.

4. To highlight the advantages of this representation for information retrieval, and

5. To find the main drawbacks of this method and the improvements that can be used in further work.

1.3 Thesis organization

In this thesis, Chapter 2 presents the basic current principles of information retrieval. The three main categories of models used in information retrieval are described: the set theoretic models, algebraic models, and probabilistic models. A marginal approach called structured text retrieval is also presented. Finally, a brief description of the measures of effectiveness of the retrieval is presented.

The ETS model and its basic principles are detailed in Chapter 3.

Chapter 4 describes the construction of the ETS representation of a professor's professional profile. The context of the study is explained in detail. The characteristics of the corpus useful to the construction of the ETS representation are developed, and finally, the representation is described.

Chapter 5 presents some classes based on the structural representations described in the previous chapter. The generating processes for the classes are highlighted.

Chapter 6 briefly explains the matching and retrieval processes for the model developed. It presents a critical analysis of the model and a comparison with one of the most used models: the Boolean model.

The thesis ends with Chapter 7, in which the major conclusions of the research are presented and recommendations are made for further research on the topic.

Chapter 2
Information Retrieval

The first appearance of the term *information retrieval* was in a paper by Mooers in 1952 [16]. The concept of information retrieval, as it is currently known emerged at the International Conference on Scientific Information held in Washington, D.C., in 1958. Thereafter, the works of Doyle and Salton established new methods of retrieval. The interest in information retrieval was initially dictated by the need for accurate sources of bibliographic work. Hence, these systems were mainly used by librarians to replace manual tools such as card catalogues and universal classification schemes. The increasing amount of data stored on each server and the nascent World Wide Web have led to a renewal of the interest in Information Retrieval Systems in the 1990s [4]. In this chapter, a conceptual view of information retrieval will be given. Then, different methods used by information retrieval systems will be detailed. The third section will be dedicated to the basic measures of efficiency in such systems.

2.1 Principles

Information retrieval addresses the representation, storage, organization of, and access to information items. It is indispensable to people to satisfy a need for information. Actually, there are only two ways to get information: the first one is simply asking other human if they have the information and the other one is to search this information using a tool—an information retrieval system.

An information retrieval system will be considered as an object working with the help of a computer but without the participation of a human being. The questioner will be called the "user". In this context, the basic principles can be explicated as follows:

- the system acts over a finite set of documents called a corpus,
- a user issues a query in a language defined by the system, and
- the information retrieval system returns a relevant subset of the set of documents.

the term information retrieval is often considered as synonymous with document retrieval and text retrieval, but there are some conceptual differences between these expressions. Document retrieval (or data retrieval) consists mainly in finding a set of documents containing one or more keywords that appear in the user query. However, a user is often more interested in finding information *about* a subject represented by her or his query than in documents containing the keywords. The search for information about a subject is the aim of information retrieval. Therefore, two questions arise: how can the user express her or his query and how should the documents be represented?

2.1.1 The user

A user consults an information retrieval system to satisfy an information need. The user has to transform her or his information need into a query in a language defined by the system. Typically, the corresponding expression is a conjunction of keywords that should convey the information. Immediately, a problem arises: the user may have a poorly defined interest or a broad information need. Another problem is due to the difficulty created by the need for the user to express her or his query in the language proposed by the system. One of the best languages for expressing this query would be the natural language since it is the language she or he knows best, but this language is still not perfect because the user may not express correctly what she or he means.

2.1.2 Representation of the documents

Indexing a document

Indexing is the action of building data structures from a text. There are three main methods for indexing: inverted files, suffix arrays, and signature files. Inverted files are discussed below. Suffix arrays consider the text as a string. A suffix is a substring of the text that goes from a fixed position to the end. Each position gives a unique suffix. The array constituted by all the suffixes is the suffix array. This suffix array is mainly used when the work needs to be performed on the substrings. Signature files are data structures based on hashing. The text is divided into text blocks. A mask is associated with each word, and another mask is associated with each text block. The text block mask is obtained by

ORing the masks of all words in the block. If a word is present in the block, the bits that are set in its mask will be set in the mask of the text block.

Inverted files

An inverted file is used to speed up the query process. A text is decomposed into a list of words called a vocabulary. To each of these words is attached a list of positions where the word appears. The set of all of these lists is called the occurrences. The space required for an inverted file is relatively small. It is presumed that the size of the vocabulary is $O(n^\beta)$ where n is the size of the text and, β is a constant between 0 and 1 depending on the text. In practice, β is between 0.4 and 0.6. The size of the occurrences is $O(n)$. Such a structure is used in the set theoretic models described in section 2.2.1.

Thesaurus

Generally, the documents of the corpus are represented by a set of keywords—index terms. To increase the number of documents retrieved, one can use a thesaurus.

A thesaurus is constituted by a list of important words in a chosen domain of knowledge, and for each of these words, a list of related words is provided. Generally, this relationship is a relationship of synonymity. This thesaurus can be used to reduce the number of index terms by providing a standard vocabulary. It can also be used to provide a hierarchy to broaden or to narrow the query of the user. One of the most important benefits is that the retrieval is based on concepts rather than on words.

2.2 Conventional Models

Baeza-Yates and Ribeiro-Neto distinguish three main models for information retrieval [2]: the Boolean model, the vector model, and the probabilistic model. The most classic model is the Boolean. It is based on sets: documents and queries are represented as sets of index terms. Gudivada, Raghavan, and Grosky called this model set theoretic [13]. The vector model is also called algebraic: the documents and queries are represented as vectors in a t-dimensional space. In the probabilistic model, probability theory guides the framework for representing documents and queries. A marginal approach consists in keeping a part of the structural information contained in the original text. This approach is called the structured text model.

The above four approaches are described in the following subsections.

An information retrieval model is defined formally in [2] by a quadruple

$$\langle \boldsymbol{D}, \boldsymbol{Q}, \mathcal{F}, R(q_i, d_j) \rangle,$$

where

- $\boldsymbol{D}$ is a set composed of representations for the documents in the collection,
- $\boldsymbol{Q}$ is a set composed of representations for the queries (the user information needs),
- $\mathcal{F}$ is a framework for modeling documents and query representations and their relationships, and

- $R(q_i, d_j)$ is a ranking function. This ranking defines an ordering among the documents with regard to the query q_i. The ranking function associates a real number with the query $q_i \in \boldsymbol{Q}$ and the document representation $d_j \in \boldsymbol{D}$.

2.2.1 Set theoretic models

Boolean model

Boolean logic has long been the most widely used framework for information retrieval. In a Boolean retrieval system, the terms of the query are linked by logical operators—AND, OR and NOT. The search engine returns only the documents in which the terms satisfy the Boolean expression of the query. It can include some functionalities to allow proximity or truncation searching. In such a model, each document either matches the query or does not. The corpus is therefore divided into two sets. There is no ranking, and there is no control over the number of documents retrieved. Moreover, all documents retrieved are assumed to have the same usefulness for the user.

Fuzzy set model

This model is based on fuzzy set theory. In this theory, the boundaries of a set are not well-defined. Each set A has a membership function μ_A associating with each element u in the universe U a value $\mu_A(u)$ in the interval $[0, 1]$. A value 1 is associated with the element if it is a full member of the class and, 0 corresponds to no membership. If B is another set and, if $\bar{A}$ is the complement of A, then,

$$\mu_{\bar{A}}(u) = 1 - \mu_A$$

$$\mu_{A\cup B}(u) = max(\mu_A(u), \mu_B(u))$$

$$\mu_{A\cap B}(u) = min(\mu_A(u), \mu_B(u))$$

The main idea is to construct a thesaurus and to use a correlation matrix $\vec{C}$ to represent the correlation of the terms in the thesaurus. The values in the matrix can be obtained by a clustering technique. Each correlation factor c_{ij} between two elements of the thesaurus, k_i and k_j, can be defined:

$$c_{ij} = \frac{n_{ij}}{n_i + n_j - n_{ij}}$$

where n_{ij} is the number of documents in which k_i and k_j appear, n_i is the number of documents which contain only k_i, and n_j is the number of documents which contain only k_j. $n_i + n_j - n_{ij}$ represents the number of documents in which k_i or k_j or both appear. Therefore, c_{ij} is the probability that a document containing k_i or k_j contains both.

The correlation matrix is used to define a membership function for each fuzzy set associated with a term k_i. The document d_j has a degree of membership of the set associated to a term k_i:

$$\mu_{i,j} = 1 - \prod_{k_l \in d_j} (1 - c_{il}).$$

A query q can be written in a disjunctive normal form. For example, the query $q = k_a \wedge (k_b \vee \bar{k_c})$ can be written as $q = (k_a \wedge k_b \wedge k_c) \vee (k_a \wedge k_b \wedge \bar{k_c}) \vee (k_a \wedge \bar{k_b} \wedge \bar{k_c})$ ($\wedge$ represents the operator AND and $\vee$ the operator OR).

The query's membership of the document d_j is defined then as:

$$\mu_{qj} = 1 - (1 - \mu_{aj}\mu_{bj}\mu_{cj})(1 - \mu_{aj}\mu_{bj}(1 - \mu_{cj}))(1 - \mu_{aj}(1 - \mu_{bj})(1 - \mu_{cj}))$$

This query membership is used to measure the relevance of the document d_j.

Extended Boolean model

The extended Boolean model has been proposed by Salton *et al.* [20]. This model is motivated by the following critique of the Boolean model. Consider a query $q = k_1 \wedge k_2$. If no document containing both k_1 and k_2 exists, all documents will be considered as irrelevant.

However a document which contains either k_1 or k_2 is more relevant than one that does not contain any of those. For a document d_j, if the weights w_{1j} and w_{2j} are respectively assigned to the pairs (k_1, d_j) and (k_2, d_j), the document and query can be plotted as in Figure 2.1. The document d_j has the coordinates (w_{1j}, w_{2j}).

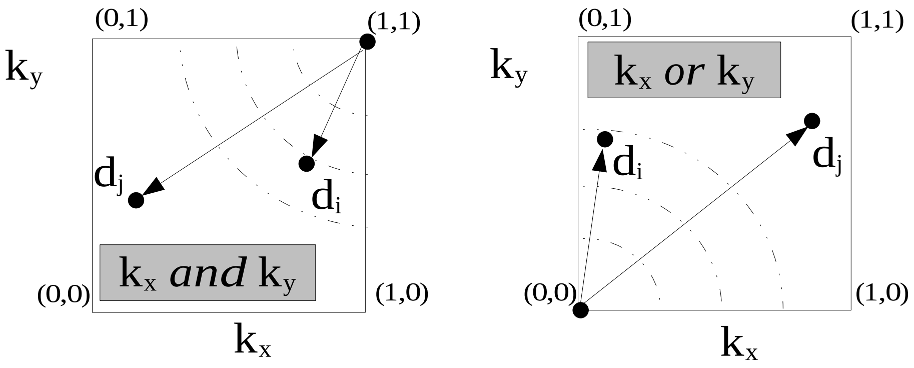

Figure 2.1: Extended Boolean model (from [2]).

Two particularities appear. For a conjunctive query $q_{and} = k_1 \wedge k_2$, the best point is $(1,1)$ whereas for a disjunctive query $q_{or} = k_1 \vee k_2$, the point $(0,0)$ is the point to be avoided. The similarity can be chosen as the following:

$$sim(q_{or}, d_j) = \sqrt{\frac{w_{1j}^2 + w_{2j}^2}{2}}$$

$$sim(q_{and}, d_j) = 1 - \sqrt{\frac{(1 - w_{1j})^2 + (1 - w_{2j})^2}{2}}$$

This formulation can be generalized to a t-dimensional space where t is the number of terms in the thesaurus.

2.2.2 Algebraic models

Vector model

The vector model was proposed to correct the problem associated with the use of only binary weights in the Boolean model. In the vector model, weight values are assigned to the terms in the query and in the documents; these weights can be different from 0 and 1. They are used to compute the degree of similarity between each document and the query. This value can then be used to assign an order of relevance for each document. Since some documents match the query only partially, this method adds some precision to the set of retrieved documents.

Formally, the vector model can be expressed as follows:

- For an index term k_i and a document d_j, a weight w_{ij}, positive and non-binary is assigned,

- For a query q, a weight w_{iq} is assigned for each index term k_i,
- A vector $\vec{q}$ is defined as $\vec{q} = (w_{1q}, w_{2q}, \cdots, w_{tq})$ where t is the number of chosen index terms,
- A document d_j is represented by a vector $\vec{d_j} = (w_{1j}, w_{2j}, \cdots, w_{tj})$, and
- The degree of similarity between q and d_j is measured by the correlation between the vectors $\vec{q}$ and $\vec{d_j}$. This correlation can be measured by the *cosine* of the angle between these vectors:

$$sim(q, d_j) = \cos(\vec{q}, \vec{d_j}) = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d_j}}{|\vec{d_j}|}.$$

The choice of the index term weights can be made by means of a clustering algorithm. This model assumes that the term vectors are linearly independent, but it also requires that they are pair wise orthogonal. For further information, one can refer to the work by Salton and McGill [21].

Generalized vector space model

This model has been proposed by Wong *et al.* in [23]. In the generalized vector space model, the term vectors are not assumed to be orthogonal, nor are they assumed to form a basis of the space. As previously mentioned, for a set of index terms $k_1, \ldots, k_t$, the weights w_{ij} are associated with the term-document pair (k_i, d_j). In each document, more than one term k_j can occur. All these co-occurrences can be represented by a set of 2^t minterms $m_1, m_2, \ldots, m_{2^t}$, with $m_1 = (0, 0, \ldots, 0)$, $m_2 = (1, 0, \ldots, 0)$, $\ldots$, $m_{2^t} = (1, 1, \ldots, 1)$. Thus, m_1 points to the documents containing none of the index terms k_j, m_2 to those documents that contain only k_1, and so on. Then, to each m_i is associated a vector $\vec{m_i}$ such that the $\vec{m_i}$ vectors are pairwise orthogonal:

$$\vec{m_1} = (1, 0, \ldots, 0)$$

$$\vec{m_2} = (0, 1, \ldots, 0)$$

$$\vdots$$

$$\vec{m_{2^t}} = (0, 0, \ldots, 1)$$

The $\vec{m_i}$'s form an orthonormal basis of a space of dimension 2^t. The set of functions g_i is defined such that $g_i(m_j)$ returns the i-th compound in the minterm m_j. To each index term k_i is associated an index term vector $\vec{k_i}$ which represents the normalized sum of the vectors associated with the minterms m_r such that k_i is in state 1 in m_r:

$$\vec{k_i} = \frac{\sum_{\{\forall r | g_i(m_r)=1\}} c_{ir} \vec{m_r}}{\sqrt{\sum_{\{\forall r | g_i(m_r)=1\}} c_{ir}^2}}$$

$$c_{ir} = \sum_{\{d_j | \forall l, g_l(\vec{d_j}) = g_l(m_r)\}} w_{ij}.$$

In the standard vector model, the document $\vec{d_j}$ is expressed as $\vec{d_j} = \sum_i w_{i,j} \vec{k_i}$ and the query $\vec{q}$ as $\vec{q} = \sum_i w_{i,q} \vec{k_i}$. In the generalized vector space model, $\vec{d_j}$ and $\vec{q}$ can be directly expressed in the space of the minterms $\vec{m_j}$ by the above equations. As previously, the similarity between $\vec{d_j}$ and $\vec{q}$ can be measured by the cosine of the angle in the space of vectors $\vec{m_j}$.

Latent semantic indexing model

The underlying idea of this model is that the meaning of the text relies on the concepts described in the document rather than on the terms used in the description of the document. The process of matching should therefore rely on the concepts rather than on the terms. This idea led Furnas *et al.* to propose the latent semantic indexing model [9]. Each document and each query is mapped to a lower dimensional space in which the retrieval should be more effective.

A matrix $\vec{M}$ with t rows and N columns is defined as a term-document association matrix: t is the number of index terms and N is the number of documents. To each element $M_{i,j}$ of the matrix is assigned a weight term w_{ij} corresponding to the pair (k_i, d_j). The singular value decomposition is used to decompose matrix $\vec{M}$:

$$\vec{M} = \vec{K}\vec{S}\vec{D}^t.$$

$\vec{S}$ is a diagonal matrix of singular values of size $r \times r$, where r is the rank of $\vec{M}$, $r = min(t, N)$. Let $\vec{M}_s$ be the matrix of rank s closest to $\vec{M}$ in the least square sense. Only the s largest values of $\vec{S}$ and the corresponding lines and columns of $\vec{K}$ and $\vec{D}^t$ are kept; the remaining values are deleted:

$$\vec{M}_s = \vec{K}_s\vec{S}_s\vec{D}_s^t.$$

The matrix $\vec{M}_s^t\vec{M}_s = (\vec{D}_s\vec{S}_s)(\vec{D}_s\vec{S}_s)^t$ gives the relationship between documents. For example, the relationship between d_i and d_j is the (i, j) element of this matrix. The matching with a query q is done by modeling the query as a pseudo-document in the original $\vec{M}$ term-document matrix.

Neural network model

This model has been described by Wilkinson in [22]. It uses neural networks with feedback and three layers. The first layer (input) represents the query terms, the second one represents the document terms, and the third represents the documents. This configuration returns a document even if it does not match the query but has a close meaning. For a detailed explanation, one can refer to [22].

2.2.3 Probabilistic models

Probabilistic model

This model was introduced by Robertson *et al.* in [19]. It is also known as the *binary independence retrieval* (BIR) model. Given a user query q, there is an ideal set of documents which contains all of the relevant documents and only these documents. The probabilistic model attempts to estimate the probability that the document d_j is relevant. The set of documents that are predicted to be relevant is denoted R. The similarity between the document d_j and the query q is computed by the ratio $P(d_j$ is relevant to $q)/P(d_j$ is not relevant to $q)$. This value can also be used to rank the documents.

The weights for the index terms are binary, $w_{ij} \in \{0,1\}$ and $w_{iq} \in \{0,1\}$. A query q is a subset of index terms. R is the set of documents that are relevant. Let $\bar{R}$ be the set of documents that are not relevant. $P(R|d_j)$ is the probability that document d_j is relevant to the query q, and $P(\bar{R}|d_j)$ is the probability that d_j is not relevant to q. The similarity between the document d_j and the query q

is the following:

$$\begin{aligned} sim(d_j, q) &= \frac{P(R|d_j)}{P(\bar{R}|d_j)} \\ &= \frac{P(d_j|R) \times P(R)}{P(d_j|\bar{R}) \times P(\bar{R})}. \end{aligned}$$

$P(d_j|R)$ is the probability of randomly selecting d_j in the set of relevant documents. $P(R)$ is the probability of randomly selecting a relevant document from the entire collection. $P(d_j|\bar{R})$ and $P(\bar{R})$ are similarly defined on the set of non-relevant documents. $P(R)$ and $P(\bar{R})$ are assumed to be the same for all documents and the index terms are assumed to be independent. Then, the similarity is computed as follows:

$$\begin{aligned} sim(d_j, q) &\approx \frac{P(d_j|R)}{P(d_j|\bar{R})} \\ &\approx \frac{(\prod_{g_i(d_j)=1} P(k_i|R)) \times (\prod_{g_i(d_j)=0} P(\bar{k}_i|R))}{(\prod_{g_i(d_j)=1} P(k_i|\bar{R})) \times (\prod_{g_i(d_j)=0} P(\bar{k}_i|\bar{R}))} \end{aligned}$$

$P(k_i|R)$ is the probability that the index term k_i is in a document randomly selected in R and $P(\bar{k}_i|R)$ is the probability that it is not; $P(k_i|R) + P(\bar{k}_i|R) = 1$.

Since $\log(\prod_{g_i(d_j)=1} P(k_i|R)) \approx \sum_{i=1}^{t} w_{iq} \times w_{ij} \times P(k_i|R)$, one can define a new similarity as the logarithm of the previous:

$$sim'(d_j, q) \approx \sum_{i=1}^{t} w_{iq} \times w_{ij} \times \left(\log \frac{P(k_i|R)}{1 - P(k_i|R)} + \log \frac{P(k_i|\bar{R})}{1 - P(k_i|\bar{R})} \right)$$

Some assumptions are needed to “guess” the relevant probability of the documents. First, it is presumed that $P(k_i|R)$ is constant for all index terms k_i, the typical value is 0.5. Secondly, it is also assumed that the distribution of the index terms in the set $\bar{R}$ is the same as in the whole collection. This assumption comes from the independence of the index terms. The probability of observing an index term k_i does not depend on presence of other index terms. Hence, the probability of observing k_i in the document containing the other index terms of the query is the same as the probability of observing k_i in the other documents. These assumptions yield:

$$P(k_i|R) = 0.5,$$

$$P(k_i|\bar{R}) = \frac{n_i}{N},$$

where n_i is the number of documents containing k_i and where N is the total number of documents.

For the following adjustments, the ranking can be improved. Let V be a subset of the first set of documents retrieved. For example, it can be the first r ranked documents where r is a threshold or r documents classified as most relevant by the user. V_i is defined as the subset of V composed of all of the documents containing k_i. Two assumptions are then made: $P(k_i|R)$ is approximated by the distribution of k_i in the documents retrieved so far, and $P(k_i|\bar{R})$ is approximated by assuming that the non-retrieved documents are not relevant:

$$P(k_i|R) = \frac{|V_i|}{|V|}$$

$$P(k_i|\bar{R}) = \frac{n_i - |V_i|}{N - |V|}.$$

Some improvements can be incorporated by modifying the previous values:

$$P(k_i|R) = \frac{|V_i| + 0.5}{|V| + 1}$$

$$P(k_i|\bar{R}) = \frac{n_i - |V_i| + 0.5}{N - |V| + 1}$$

or

$$P(k_i|R) = \frac{|V_i| + \frac{n_i}{N}}{|V| + 1}$$
$$P(k_i|\bar{R}) = \frac{n_i - |V_i| + \frac{n_i}{N}}{N - |V| + 1}.$$

These adjustments are useful in calculating the similarity between the query and a set of retrieved documents when $|V|$ is 1 or $|V_i|$ is 0.

Bayesian networks

The two next models are based on Bayesian networks. A Bayesian network is a directed acyclic graph in which the nodes represent random variables. The causal relationships between the variables are represented by the edges, and their strengths are represented by conditional probabilities. Hence, the parents of a node are "the cause" of the node. The nodes without parents are the roots of the network. Let G be a Bayesian network, x_i be a node in G, and Γ_{x_i} be the set of parent nodes of x_i. A set of functions $F_i(x_i, \Gamma_{x_i})$ specifies the influence of Γ_{x_i} on x_i:

$$\sum_{\forall x_i} F_i(x_i, \Gamma_{x_i}) = 1$$
$$0 \leq F_i(x_i, \Gamma_{x_i}) \leq 1$$

Inference network model

In this model, random variables are associated with index terms, documents, and queries. The event of observing the document d_j is represented by the random variable associated with this document. Observing document d_j asserts a belief upon the variables associated with its index terms. Index terms and document variables are represented as nodes in the network. Edges directed from a document node to an index term node indicate that observing this document improved the belief value of the term nodes.

The random variables associated with the user query are used to express the fact that the request for information has been satisfied. These variables are also represented by nodes in the network. Since these variables depend upon the query terms, the edges are directed from the index term nodes to the query nodes. Let $\vec{k}$ be a t-dimensional vector, $\vec{k} = (k_1, k_2, \ldots, k_t)$, where k_i is a binary random variable associated with an index term. There exist 2^t possible values for $\vec{k}$. The ranking of a document d_j with respect to a query q is computed as $P(q \wedge d_j)$.

$$\begin{aligned}
P(q \wedge d_j) &= \sum_{\forall \vec{k}} P(q \wedge d_j | \vec{k}) \times P(\vec{k}) \\
&= \sum_{\forall \vec{k}} P(q \wedge d_j \wedge \vec{k}) \\
&= \sum_{\forall \vec{k}} P(q | d_j \wedge \vec{k}) \times P(d_j \wedge \vec{k}) \\
&= \sum_{\forall \vec{k}} P(q | \vec{k}) \times P(\vec{k} | d_j) \times P(d_j)
\end{aligned}$$

Thus,

$$P(q \wedge d_j) = \sum_{\forall \vec{k}} P(q|\vec{k}) \times \prod_{\forall i | g_i(\vec{k})=1} P(k_i|d_j) \times \prod_{\forall i | g_i(\vec{k})=0} P(\bar{k}_i|d_j) \times P(d_j)$$

where $g_i(\vec{k})$ is the function such that $g_i(\vec{k}) = 0 \Leftrightarrow k_i = 1$ in $\vec{k}$.

Belief network model

Probability space. Let K be the set of all index terms. Let $U \subset K$ be a subset of K. With U is associated a vector $\vec{k}$ such that $g_i(\vec{k}) = 1 \Leftrightarrow k_i \in U$.

K is viewed as a concept space and each document as an elementary concept. U represents a document or a query. Each k_i represents the membership of the corresponding index term to the concept while the concept is represented by $\vec{k}$.

Belief network model. In this model, as in the previous one, the query q is represented by a network node associated with a random variable. $P(q)$ represents the degree of coverage of the space K by q. The same representation as above is used for a document d_j. Both q and d_j are represented by subsets of index terms. Contrary to the previous model, a document node is pointed to by the index terms that compose the document. The ranking of document d_j with respect to q is given by $P(d_j|q)$:

$$P(d_j|q) \quad = \quad P(d_j \wedge q)/P(q).$$

Since $P(q)$ is constant for all documents, the ranking is given by $P(d_j \wedge q)$. Using the formula derived in the previous section, one produces the following:

$$P(d_j|q) \quad \approx \quad \sum_{\forall u} P(d_j \wedge q|u) \times P(u).$$

2.2.4 Structured text retrieval models

These models have been developed to combine the information in the content with the information related to the structure of the documents. In this section, only two of these models will be outlined: one is based on non-overlapping lists and, the other one is based on a structure close to a tree called proximal nodes.

Model based on non-overlapping lists

In this model proposed by Burkowski in [6, 7], the text of the document is divided into text regions and is collected in several lists. If the division of the text into regions can be multiple, then there can be more than one list. The text can be divided into units such as chapters and sections. The text regions from different lists can overlap.

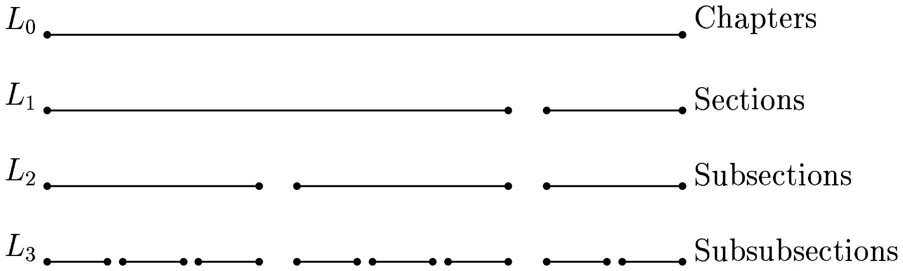

Figure 2.2: Example of the representation of a document structure.

An inverted text is built for the text regions; each structural component is an entry in the index. The query can be expressed as follows: select a region that does not contain another region or a region that contains a specified word.

Model based on proximal nodes

This model was proposed by Navarro and Baeza-Yates [1, 17, 18]. It allows one to upondefine independent hierarchical indexing structures over a text document. The nodes are the elements of the hierarchy (chapters, sections, paragraphs, etc). This query is then processed by matching the elements of the hierarchy from the top to the bottom until the match is successful. This model is more powerful than the model based on non-overlapping lists since the query formulated can be more complex.

2.3 Measure of the retrieval effectiveness

As explained in [3], the efficiency of a search engine can be measured by the ratio of the relevant documents and the retrieved documents. One focuses on a particular query; with respect to this query, one can build the set *Rel* of documents that are relevant and the set *Retr* of documents that are retrieved by the search engine. A good search engine will retrieve the set *Rel* and only this set. To know the real efficiency of a search engine, generally, two values can be used. If the main criterion selected is the exhaustiveness, one can use the measure *Recall* defined as

$$Recall = \frac{|Retr \cap Rel|}{|Rel|}.$$

Otherwise, if the criterion is the relevancy, the measure $Precision$ defined by:

$$Precision = \frac{|Retr \cap Rel|}{|Retr|}$$

can be used. It can easily be shown that a good tool will have a value close to one for the appropriate criterion (or both), whereas a less interesting one will have smaller values.

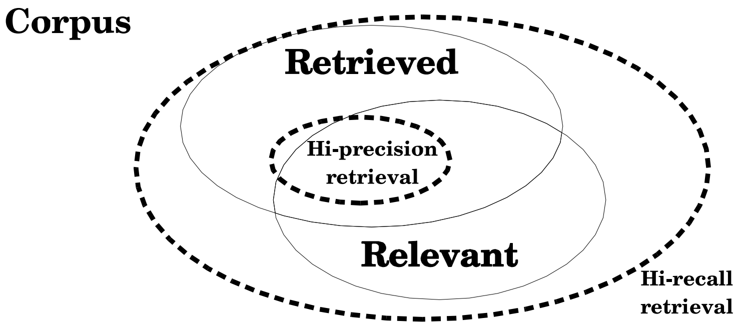

Figure 2.3: Comparison of the sets *Retr* and *Rel* (from [3]).

Chapter 3
ETS Model

The ETS model is a model for structural representations proposed by Goldfarb *et al.* (for the last documented version, see [11, 10]). It has initially been designed for pattern recognition but its advantages have also been shown in chemistry [12] and in genomics [14]. In this chapter, a short explanation of what a structural representation is will be given and the basic principles of the ETS model will be detailed.

3.1 Structural representations

In pattern recognition, two main approaches are generally considered [8, 5]. The first one is the *decision-theoretic* approach. Decision-theoretic methods mainly use numeric-valued features to represent patterns. These features are a set of characteristic measurements that are extracted from the patterns. The distances between the different patterns are measured only by traditional distances (based on the distance between the feature vectors). Separation of classes involves only partitioning of the feature space by one or more hyperplanes, or more generally hypersurfaces. This approach is purely quantitative; it assumes all relevant relationships can be represented numerically.

The second approach is a *structural approach.* This approach is also called syntactic when it is based on the Chomsky grammars. Syntactic and structural methods are based on the representation of the class by the characteristic way in which the subpatterns are related to each other. These relations are called the structure of the pattern. Chomsky grammars are the structural models most commonly used, together with their variations (graph grammars, tree grammars, etc. [15]).

3.2 The ETS model

This chapter focuses on the ETS model, a new structural model. The following definitions come from [11], although a completely revised version is in preparation. To maintain consistency with the source, we will use the same notation as in [11].

3.2.1 Primtypes (or primitive type)

The primtypes are the elements of a finite set Π. For each element (primtype) $\pi \in \Pi$, two sets, $init(\pi)$ and $term(\pi)$, are given. These two sets are subsets of a fixed set of abstract sites (or a-sites) A. They are respectively the sets of initial a-sites and terminal a-sites of the primtype π. The set of all a-sites is $sites(\pi) = init(\pi) \cup term(\pi)$.

Pictorially, the more intuitive representations for primtypes are spheres (Figure 3.1 left). The elements of the set of initial a-sites are represented by points in the upper hemisphere while the terminal a-sites of the primtype are represented in the lower hemisphere. The points on the equator will be the points in the intersection of both sets. To simplify the drawings, the primtypes will be drawn as circles (Figure 3.1 right).

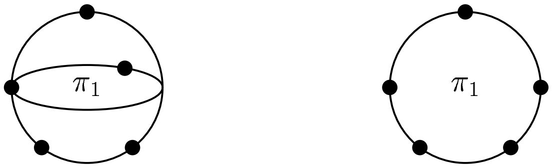

Figure 3.1: Representation of a primtype π_1.

3.2.2 Composites

Composites

The concrete sites, or simply sites, are the elements of a countably infinite set S.

The composites (or primitive composites or simply primitives) are the elements of the set Γ. For each $\gamma \in \Gamma$, we define three subsets of S: $init(\gamma)$, $term(\gamma)$ and $sites(\gamma)$ called respectively the set of initials, the set of terminals, and the set of all sites of the composite γ. These sets are constructed inductively as follows.

First, let λ be the null composite.

$$init(\lambda) = term(\lambda) = sites(\lambda) = \emptyset$$

For $\pi \in \Pi$, let f be a fixed injective mapping $f : sites(\pi) \to S$. f is called the site realization for primitive π.

The primitive $\pi\langle f\rangle$ is a primitive whose sets are defined as follows:

$$init(\pi\langle f\rangle) = f(init(\pi)),$$

$$term(\pi\langle f\rangle) = f(term(\pi)),$$

$$sites(\pi\langle f\rangle) = f(sites(\pi)).$$

$\pi\langle f\rangle$ is an element of Γ.

The primitive $\pi_1\langle f_1\rangle$ is represented in Figure 3.2. The initials and terminals are represented by the lines that start or end at the abstract sites.

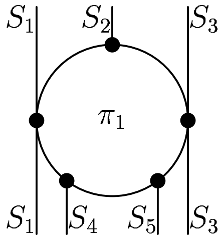

Figure 3.2: Representation of the primitive $\pi_1\langle f_1\rangle$ (π_1 is the primtype represented in Figure 3.1).

A new composite $\gamma' \in \Gamma$ is created from two primitives: $\gamma \in \Gamma$, $\gamma \neq \lambda$, and $\pi\langle f\rangle \in \Gamma$ satisfying

$$site(\gamma) \cap sites(\pi\langle f\rangle) = term(\gamma) \cap init(\pi\langle f\rangle).$$

The composite $\gamma\prime$ is defined by the expression:

$$\gamma \lhd \pi\langle f\rangle,$$

which means that the sets of concrete sites of γ' are defined as follows:

$$init(\gamma') = init(\gamma) \cup [init(\pi\langle f\rangle) \setminus term(\gamma)],$$

$$term(\gamma') = [term(\gamma) \setminus init(\pi\langle f\rangle)] \cup term(\pi\langle f\rangle),$$

$$sites(\gamma') = sites(\gamma) \cup sites(\pi\langle f\rangle).$$

The operation by which γ' is obtained is called the attachment of the primitive $\pi\langle f\rangle$ to γ. The attachment consists of connecting the identical sites in $term(\gamma)$ and $init(\pi\langle f\rangle)$.

Thus, each composite is inductively defined by its construction process:

$$\gamma = \pi_1\langle f_1\rangle \triangleleft \pi_2\langle f_2\rangle \triangleleft \cdots \triangleleft \pi_n\langle f_n\rangle.$$

The continuation sites of the composite γ are defined as:

$$cont(\gamma) = init(\gamma) \cap term(\gamma).$$

The attachment of two composites $\pi_1\langle f_1\rangle \triangleleft \pi_2\langle f_2\rangle$ and $\pi_3\langle f_3\rangle$ can be represented as in Figure 3.3. The initials of the primitive ($\pi_3\langle f_3\rangle$) are "linked" to the terminals of the primitive ($\pi_1\langle f_1\rangle \triangleleft \pi_2\langle f_2\rangle$).

Due to the construction of the composite γ, the number of elements in the union of the initial and terminal sites can be smaller than the total number of sites. The sets of internal and external sites are respectively defined by:

$$ext(\gamma) = init(\gamma) \cup term(\gamma),$$

$$int(\gamma) = sites(\gamma) \setminus ext(\gamma).$$

Site replacement

For $\gamma \in \Gamma$, the site replacement h is an injective mapping

$$h : sites(\gamma) \to S.$$

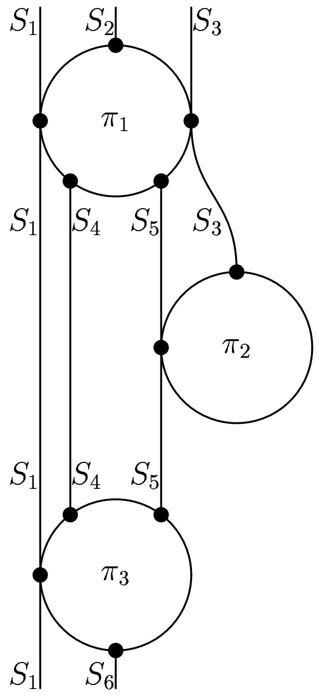

Figure 3.3: Representation of the attachment $(\pi_1\langle f_1\rangle \triangleleft \pi_2\langle f_2\rangle) \triangleleft \pi_3\langle f_3\rangle$.

The composite $\gamma\langle h\rangle$ is defined inductively as:

- $\lambda\langle h\rangle \overset{def}{=} \lambda$.

- If $\gamma = \pi\langle f\rangle$, then $\gamma\langle h\rangle \overset{def}{=} \pi\langle g\rangle$ where $g = h \circ f$.

- The site replacement is then defined inductively. Let $h' = h|_{sites(\alpha)}$, $h' : sites(\alpha) \rightarrow S$ and $g = h \circ f$. If $\gamma = \alpha \triangleleft \pi\langle f\rangle$ and $\alpha\langle h'\rangle$ has been constructed,

$$\gamma\langle h\rangle \overset{def}{=} \alpha\langle h'\rangle \triangleleft \pi\langle g\rangle.$$

The site replacement can be assimilated to the relabeling of the sites of a composite (See Figure 3.4).

The following relationships are true:

$$init(\gamma\langle h\rangle) = h(init(\gamma)),$$

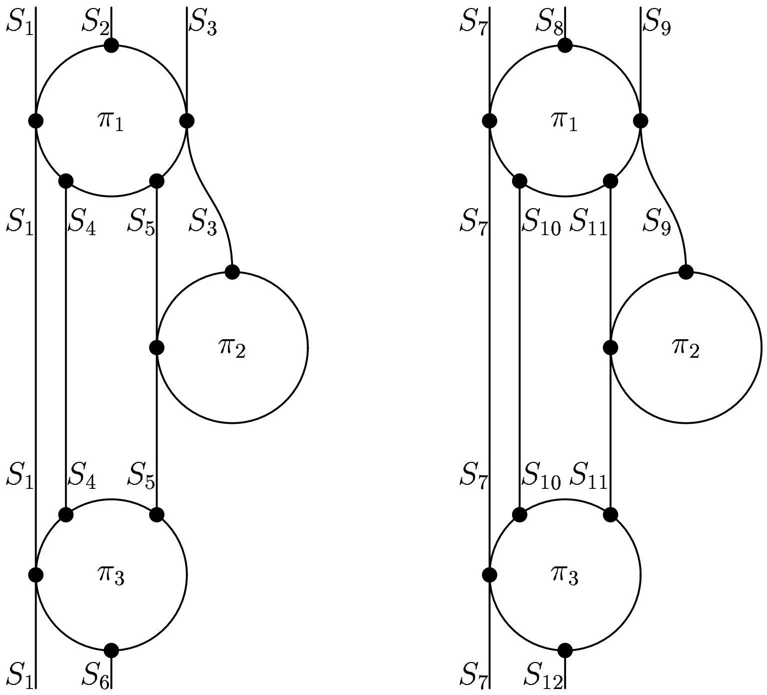

Figure 3.4: Site replacement. The internal sites of the composite $\gamma = (\pi_1\langle f_1\rangle \lhd \pi_2\langle f_2\rangle) \lhd \pi_3\langle f_3\rangle$ (left) are replaced by the mapping h to obtain the composite $\gamma\langle h\rangle$ (right).

$$term(\gamma\langle h\rangle) = h(term(\gamma)),$$

$$sites(\gamma\langle h\rangle) = h(sites(\gamma)).$$

For a demonstration of these properties and the following ones, refer to [11].

If γ is a composite, $\gamma \in \Gamma$, and $h_1 : sites(\gamma) \to S$ and $h_2 : sites(\gamma\langle h_1\rangle) \to S$ are two site replacements,

$$(\gamma\langle h_1\rangle)\langle h_2\rangle = \gamma\langle h_2 \circ h_1\rangle.$$

Let γ be a composite, and h a site replacement, $h : sites(\gamma) \to S$. If γ' is the composite such that $\gamma' = \gamma\langle h\rangle$, there exists the site replacement $h' : sites(\gamma') \to S$ such that $\gamma'\langle h'\rangle = \gamma$.

Similarity

Two composites α and β will be called similar if there exists a site replacement $h : site(\beta) \to S$ such that

$$h|_{ext(\beta)} = id$$

where id represents the identity and

$$\alpha = \beta\langle h \rangle.$$

The similarity between α and β is denoted by $\alpha \approx \beta$.

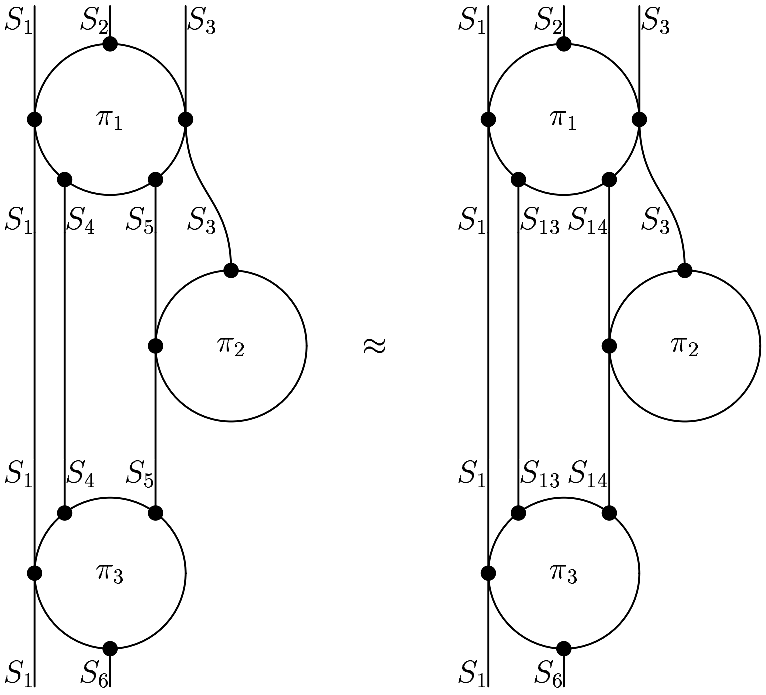

Figure 3.5: Similarity between two composites.

From this definition, we can infer that two composites are similar only if we can relabel the internal sites of one and get the other (Figure 3.5).

Let α and β be two similar composites and h the site replacement such that

$\alpha = \beta\langle h \rangle$, then

$$init(\alpha) = init(\beta),$$

$$term(\alpha) = term(\beta),$$

$$int(\alpha) = h(int(\beta)).$$

Composition

Let α and β be two composites. If α and β satisfy the condition

$$sites(\alpha) \cap sites(\beta) = term(\alpha) \cap init(\beta)$$

the composition of the two composites exists. It is denoted

$$\alpha \triangleleft \beta$$

and defined by induction on β as:

- $\alpha \triangleleft \lambda \overset{def}{=} \alpha$
- $\alpha \triangleleft \pi\langle f \rangle \overset{def}{=} \begin{cases} \pi\langle f \rangle, & \alpha = \lambda \\ \alpha \triangleleft \pi\langle f \rangle, & \alpha \neq \lambda \end{cases}$
- If $\beta = \gamma \triangleleft \pi\langle f \rangle$ and $\alpha \triangleleft \gamma$ has been constructed,

$$\alpha \triangleleft \beta \overset{def}{=} (\alpha \triangleleft \gamma) \triangleleft \pi\langle f \rangle.$$

The composition of two composites can be represented as in Figure 3.6. The initials of the composite $(\pi_4\langle f_4 \rangle \triangleleft \pi_5\langle f_5 \rangle)$ are "linked" to the terminals of the composite $(\pi_1\langle f_1 \rangle \triangleleft \pi_2\langle f_2 \rangle)$.

The sets of initial sites, of terminal sites, of all sites, and of continuation sites of the composition are therefore as follows:

$$init(\alpha \triangleleft \beta) = init(\alpha) \cup [init(\beta) \setminus term(\alpha)]$$

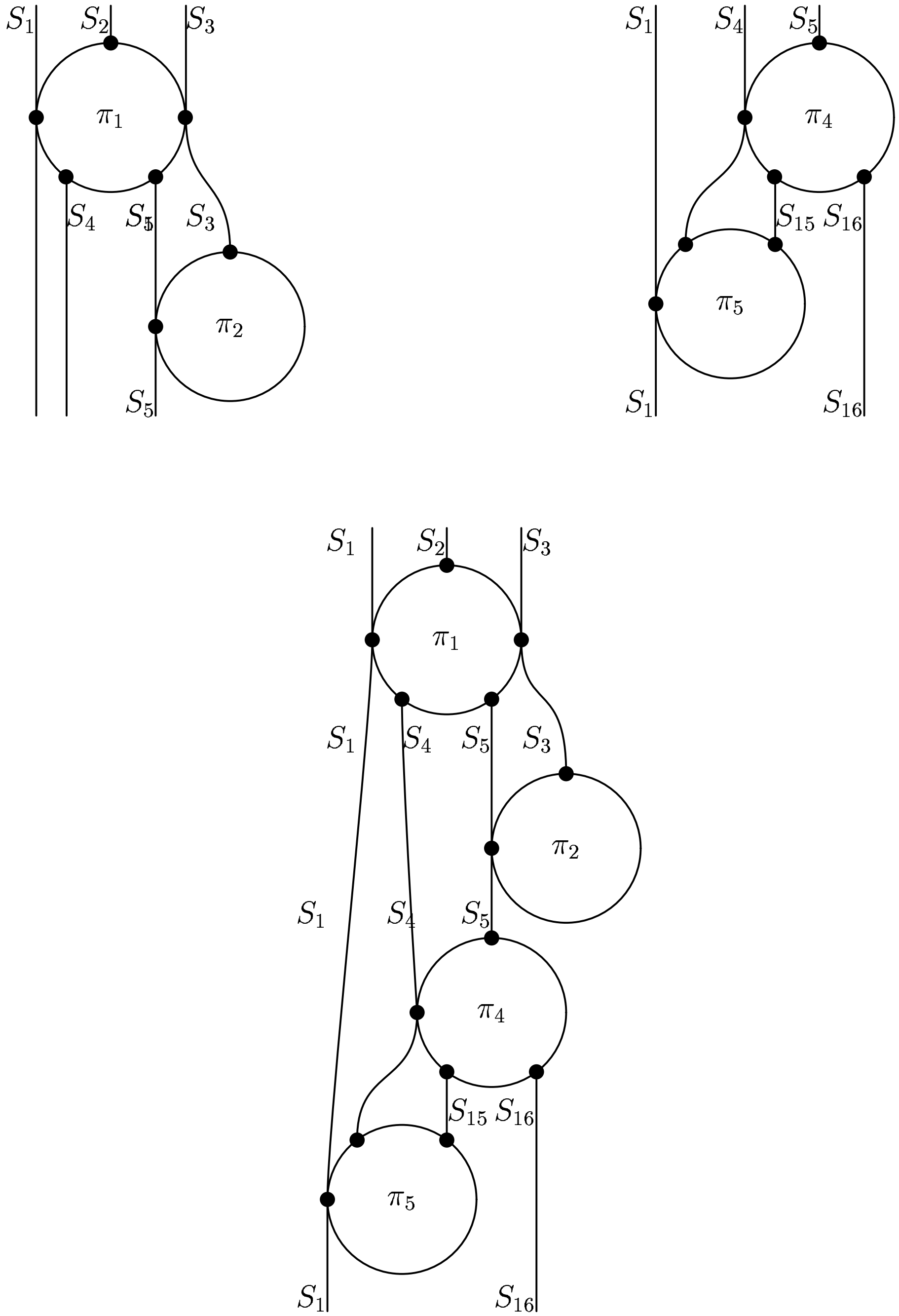

Figure 3.6: Composition of two composites $(\pi_1\langle f_1\rangle \lhd \pi_2\langle f_2\rangle) \lhd (\pi_4\langle f_4\rangle \lhd \pi_5\langle f_5\rangle)$.

$$term(\alpha \triangleleft \beta) = [term(\alpha) \setminus init(\beta)] \cup term(\beta)$$

$$sites(\alpha \triangleleft \beta) = sites(\alpha) \cup sites(\beta)$$

$$cont(\alpha \triangleleft \beta) = [cont(\alpha) \cap cont(\beta)] \cup [cont(\alpha) \setminus sites(\beta)] \cup [cont(\beta) \setminus sites(\alpha)]$$

The composition is associative: let α, β and γ be three composites for which the compositions $\alpha \triangleleft \beta$ and $(\alpha \triangleleft \beta) \triangleleft \gamma$ exist; then $\beta \triangleleft \gamma$ and $\alpha \triangleleft (\beta \triangleleft \gamma)$ exist and

$$(\alpha \triangleleft \beta) \triangleleft \gamma = \alpha \triangleleft (\beta \triangleleft \gamma) = \alpha \triangleleft \beta \triangleleft \gamma.$$

The demonstration of this property can be found in [11].

Semantic identities

Let α and β be two composites. The semantic identity between α and β is defined by

$$\begin{aligned} & init(\alpha) = init(\beta) \\ \text{and} \quad & term(\alpha) = term(\beta) \end{aligned}$$

and is denoted

$$\alpha \equiv \beta.$$

It means that these two composites are indistinguishable in an object representation from an external point of view.

Semantic equivalence relation

Let $\mathcal{I}$ be a specified set of semantic identities. The semantic equivalence relation or semantic relation induced by this set is denoted $\sim$ and defined on the set of composites Γ as follows:

- If $\alpha \equiv \beta$, $(\alpha, \beta) \in \mathcal{I}$, $\alpha \sim \beta$.

- If $\alpha \sim \beta$ and

  $$f : sites(\alpha) \rightarrow S$$

  $$g : sites(\beta) \rightarrow S$$

  are externally consistent, *i.e.* satisfy

  $$f|_{ext(\alpha)} = g|_{ext(\beta)},$$

  then

  $$\alpha\langle f \rangle \sim \beta\langle g \rangle.$$

- If $\alpha \sim \beta$, $\gamma \sim \delta$ and $\alpha \lhd \gamma$ and $\beta \lhd \delta$ exist, then

  $$\alpha \lhd \gamma \sim \beta \lhd \delta.$$

The relation $\sim$ can be expressed as:

For α, $\beta \in \Gamma$,

$$\alpha \sim \beta \Leftrightarrow \left\{ \begin{array}{l} \exists n \in \mathbb{N} \mid \forall i \in [1, n] \; \exists \alpha_i \in \Gamma \mid \\ \qquad\qquad \alpha_i \equiv \alpha_{i-1} \\ \qquad \text{or} \quad \alpha_i = \alpha_{i-1}\langle f_i \rangle \\ \qquad\qquad \text{with } f_i : site(\alpha_{i-1}) \rightarrow S \text{ such that } f_i|_{ext(\alpha_{i-1})} = id \\ \text{with } \alpha_0 = \alpha \text{ and } \alpha_n = \beta. \end{array} \right.$$

The following properties hold:

- If two composites are semantically equivalent, they have identical initial and terminal sites.

- If $\alpha \approx \beta$, then $\alpha \sim \beta$.

Site equivalence identity

Let $\pi\langle f\rangle$ be a primitive and h a site replacement satisfying:

$$h[init(\pi\langle f\rangle)] = init(\pi\langle f\rangle)$$

and

$$h[term(\pi\langle f\rangle)] = term(\pi\langle f\rangle).$$

The semantic identity

$$\pi\langle f\rangle \equiv \pi\langle h \circ f\rangle$$

is called the site equivalence identity. The set of all possible such identities will be denoted $Eqsite(\Pi)$.

3.2.3 Istructs (or instance structs)

Istructs (or instance structs)

Let Π be a specified set of primtypes and $\mathcal{I}$ a specified set of semantic identities. The instance structs or istructs (for $(\Pi,\mathcal{I})$) are the elements of the quotient set

$$\Theta = \Gamma/\sim = \{[\gamma] | \gamma \in \Gamma\}$$

where $\sim$ is the semantic relation induced by $\mathcal{I}$ and Γ is the set of primtypes constructed from Π.

Istruct $[\gamma]$, $\gamma \in \Gamma$, is also denoted $\boldsymbol{\gamma}$. For istruct $\boldsymbol{\gamma}$, three sets of sites are defined:

$$init(\boldsymbol{\gamma}) = init(\gamma),$$

$$term(\boldsymbol{\gamma}) = term(\gamma),$$

and

$$ext(\boldsymbol{\gamma}) = ext(\gamma).$$

Istruct $[\lambda]$ or $\boldsymbol{\lambda}$ is the empty istruct.

The set Θ could be thought of as a the set of canonical elements of Γ.

Istructs site replacement

Let $\boldsymbol{\gamma} \in \Theta$. An istruct site replacement is an injective mapping $\boldsymbol{h} : ext(\boldsymbol{\gamma}) \to S$. $\boldsymbol{\gamma}\langle \boldsymbol{h} \rangle$ is defined as

$$\boldsymbol{\gamma}\langle \boldsymbol{h} \rangle = [\gamma\langle h \rangle],$$

where γ is an element of the class of $\boldsymbol{\gamma}$ and h is a site replacement $h : sites(\gamma) \to S$ satisfying:

$$h|_{ext(\gamma)} = \boldsymbol{h}.$$

Istructs composition

Let $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ be two istructs such that

$$ext(\boldsymbol{\alpha}) \cap ext(\boldsymbol{\beta}) = term(\boldsymbol{\alpha}) \cap init(\boldsymbol{\beta}).$$

The composition of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is defined as

$$\boldsymbol{\alpha} \triangleleft \boldsymbol{\beta} = [\alpha \triangleleft \beta]$$

where α and β are respectively elements of the classes of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, and $\alpha \triangleleft \beta$ exists.

The composition of istructs is associative.

Parallel compositions

Let $\alpha, \beta \in \Gamma$ such that

$$sites(\alpha) \cap sites(\beta) = cont(\alpha) \cap cont(\beta).$$

The parallel composition of α and β is denoted

$$\alpha || \beta$$

and is the composite $\alpha \lhd \beta$.

If

$$\alpha = \pi_1\langle f_1\rangle || \pi_2\langle f_2\rangle$$

and

$$\beta = \pi_2\langle f_2\rangle || \pi_1\langle f_1\rangle,$$

the semantic identity

$$\alpha \equiv \beta$$

is a commutative identity. The set of all such possible identities will be denoted $Comm(\Pi)$.

3.2.4 Itransformations or instance transformations

Itransformations or instance transformations

Let Π and $\mathcal{I}$ be respectively a set of primtypes and a set of semantic identities. Θ is the corresponding set of istructs. An instance transformation, or itransformation, is a pair $\boldsymbol{\tau} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are two istructs) such that there exists

$\boldsymbol{\delta} \in \Theta$ satisfying

$$\boldsymbol{\beta} = \boldsymbol{\alpha} \triangleleft \boldsymbol{\delta}.$$

The istruct $\boldsymbol{\alpha}$ will be called the context of the itransformation $\boldsymbol{\tau}$ and

$$ext(\boldsymbol{\tau}) \overset{def}{=} ext(\boldsymbol{\alpha}) \cup ext(\boldsymbol{\beta}).$$

If $\boldsymbol{\alpha} = [\lambda]$, the itransformation is called context free. $\mathcal{T}$ will denote the set of all itransformations for $(\Pi, \mathcal{I})$.

Itransformation site replacement

For an itransformation $\boldsymbol{\tau} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, an itransformation site replacement is an injective mapping $h : ext(\boldsymbol{\tau}) \to S$. The itransformation $\boldsymbol{\tau}\langle h \rangle$ is defined as

$$\boldsymbol{\tau}\langle h \rangle \overset{def}{=} (\boldsymbol{\alpha}\langle h|_{ext(\boldsymbol{\alpha})}\rangle, \boldsymbol{\beta}\langle h|_{ext(\boldsymbol{\beta})}\rangle).$$

Itransformation of istruct

For an istruct $\boldsymbol{\gamma}$ and an itransformation $\boldsymbol{\tau} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ satisfying

$$\boldsymbol{\gamma} = \boldsymbol{\gamma}_{front} \triangleleft \boldsymbol{\alpha},$$

the $\boldsymbol{\tau}$-itransformation of an istruct $\boldsymbol{\gamma}$, denoted $\boldsymbol{\gamma} \triangleleft \boldsymbol{\tau}$, is defined as

$$\boldsymbol{\gamma}_{front} \triangleleft \boldsymbol{\beta}.$$

The action of an itransformation $\boldsymbol{\tau}$ can be viewed as an attachment of $\boldsymbol{\tau}$ to the istruct which contains the context of this itransformation. The itransformation does not delete any information from the istruct since the context is not erased. Therefore, this itransformation is an evolutionary transformation. It is the origin of the name of this model (evolving transformations system – ETS).

Let $\boldsymbol{\gamma}, \boldsymbol{\gamma}' \in \Theta$, $\boldsymbol{\tau} = (\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{T}$.

$$\boldsymbol{\gamma} \xrightarrow{\boldsymbol{\tau}} \boldsymbol{\gamma}'$$

signifies that there exists a site replacement $\boldsymbol{h}$ such that

$$\boldsymbol{\gamma}' = \boldsymbol{\gamma} \triangleleft \boldsymbol{\tau}\langle \boldsymbol{h} \rangle.$$

Let $\boldsymbol{\gamma}$ be an istruct. Its set of immediate ancestors is defined as

$$\mathcal{AI}(\boldsymbol{\gamma}) \overset{def}{=} \{\boldsymbol{\alpha} \in \Theta | (\boldsymbol{\alpha} \neq \boldsymbol{\gamma}) \text{ and } (\exists \boldsymbol{\tau} \in \mathcal{T} | \boldsymbol{\alpha} \xrightarrow{\boldsymbol{\tau}} \boldsymbol{\gamma})\}.$$

Induction axiom for istructs

Let Θ' be a subset of Θ. If

$$\boldsymbol{\lambda} \in \Theta'$$

$$\text{and} \quad \forall \boldsymbol{\gamma} \in \Theta \quad [\mathcal{AI}(\boldsymbol{\gamma}) \subseteq \Theta' \Rightarrow \boldsymbol{\gamma} \in \Theta'],$$

then $\Theta' = \Theta$.

Inductive structure

Let Π be a finite set of primtypes, $\mathcal{I}$ be a set of semantic identities, Θ the corresponding set of istructs, and $\mathcal{T}$ the set of itransformations. $(\Pi, \mathcal{T})$ will be called an inductive structure only if the induction axiom for istructs holds.

3.2.5 Transformation systems and classes

Structs, transformations

Let istruct $\boldsymbol{\gamma} \in \Theta$. The struct corresponding to $\boldsymbol{\gamma}$ is defined as

$$\bar{\boldsymbol{\gamma}} \overset{def}{=} \{\boldsymbol{\gamma}\langle \boldsymbol{h} \rangle | \boldsymbol{h} \text{ is a site replacement }\}.$$

Let itransformation $\boldsymbol{\tau} \in \mathcal{T}$. The transformation corresponding to $\boldsymbol{\tau}$ is defined as

$$\bar{\boldsymbol{\tau}} \overset{def}{=} \{\boldsymbol{\tau}\langle \boldsymbol{h} \rangle | \boldsymbol{h} \text{ is a site replacement } \}.$$

In the inductive structure $(\Pi, \mathcal{I})$, the set of structs will be denoted as

$$\bar{\boldsymbol{\Theta}} \overset{def}{=} \{\bar{\gamma} | \boldsymbol{\gamma} \in \Theta\}$$

and the set of transformations will be denoted as

$$\bar{\mathcal{T}} \overset{def}{=} \{\bar{\boldsymbol{\tau}} | \boldsymbol{\tau} \in \mathcal{T}\}.$$

A transformation set is a finite set of transformations $T \subset \bar{\mathcal{T}}$.

Ipaths

Let T be a transformation set in $(\Pi, \mathcal{I})$. Let $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n$ be itransformations such that $\bar{\boldsymbol{\tau}}_1, \ldots, \bar{\boldsymbol{\tau}}_n \in T$ and $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{n+1}$ be istructs such that

$$\boldsymbol{\alpha}_{i+1} = \boldsymbol{\alpha}_i \triangleleft \boldsymbol{\tau}_i, \; i \in [1, n].$$

The tuple

$$(\boldsymbol{\alpha}_1, \boldsymbol{\tau}_1, \boldsymbol{\alpha}_2, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_n, \boldsymbol{\alpha}_{n+1})$$

is a T-ipath or ipath from $\boldsymbol{\alpha}_1$ to $\boldsymbol{\alpha}_{n+1}$. The set of all T-ipaths will be denoted by IP_T.

Let c be an ipath. The beginning and end of ipath c are respectively

$$begin(c) \overset{def}{=} \boldsymbol{\alpha}_1$$

and

$$end(c) \overset{def}{=} \boldsymbol{\alpha}_{n+1}.$$

The length of c is n:

$$|c| \overset{def}{=} n.$$

Let c_1, $c_2 \in IP_T$,

$$c_1 = (\boldsymbol{\alpha}_1, \boldsymbol{\tau}_1, \boldsymbol{\alpha}_2, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_n, \boldsymbol{\alpha}_{n+1})$$

and

$$c_2 = (\boldsymbol{\beta}_1, \boldsymbol{\delta}_1, \boldsymbol{\beta}_2, \boldsymbol{\delta}_2, \ldots, \boldsymbol{\delta}_m, \boldsymbol{\beta}_{m+1}).$$

If $m = n$ and there exist site replacements $g_i : ext(\boldsymbol{\alpha}_i) \to S$, $i \in [1, n+1]$ and $h_j : ext(\boldsymbol{\tau}_j) \to S$, $j \in [1, n]$ such that

$$\boldsymbol{\beta}_i = \boldsymbol{\alpha}_i\langle g_i\rangle,\ i \in [1, n+1],$$

$$\boldsymbol{\delta}_j = \boldsymbol{\tau}_j\langle h_j\rangle,\ j \in [1, n],$$

and for $j \in [1, n]$

$$h_i|_{ext(\boldsymbol{\tau}_j)\cap ext(\boldsymbol{\alpha}_j)} = g_i|_{ext(\boldsymbol{\tau}_j)\cap ext(\boldsymbol{\alpha}_j)},$$

$$h_i|_{ext(\boldsymbol{\tau}_j)\cap ext(\boldsymbol{\alpha}_{j+1})} = g_i|_{ext(\boldsymbol{\tau}_j)\cap ext(\boldsymbol{\alpha}_{j+1})},$$

c_1 and c_2 will be said to be equivalent: $c_1 \sim c_2$.

Paths

Let c be an ipath. The T-path, or simply path, corresponding to c is:

$$\bar{c} \overset{def}{=} \{c' \in IP_T | c' \sim c\}.$$

For a path $\bar{c}$, the beginning and end of $\bar{c}$ are respectively $begin(\bar{c}) \overset{def}{=} begin(c)$ and $end(\bar{c}) \overset{def}{=} end(c)$; the length of path $\bar{c}$ is $|\bar{c}| \overset{def}{=} |c|$.

Let $n \geq 0$. The set of all paths of length n is denoted P_T^n and the set of all paths is denoted by $P_T \overset{def}{=} \bigcup_{n=0}^{\infty} P_T^n$.

Let $\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}} \in \bar{\boldsymbol{\Theta}}$. The sets of all paths and all paths of length n from $\bar{\boldsymbol{\alpha}}$ to $\bar{\boldsymbol{\beta}}$ are, respectively,

$$P_T(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}) \overset{def}{=} \{p \in P_T | begin(p) = \bar{\boldsymbol{\alpha}} \text{ and } end(p) = \bar{\boldsymbol{\beta}}\}$$

and

$$P_T^n(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}) \overset{def}{=} P_T(\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}) \cap P_T^n.$$

Path composition

Let $c_1 = (\boldsymbol{\alpha}_1, \boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n, \boldsymbol{\alpha}_{n+1})$ and $c_2 = (\boldsymbol{\alpha}_{n+1}, \boldsymbol{\tau}_{n+1}, \ldots, \boldsymbol{\tau}_{n+m}, \boldsymbol{\alpha}_{n+m+1})$ be T-ipaths. The composition of c_1 and c_2 is

$$c_2 \circ c_1 \overset{def}{=} (\boldsymbol{\alpha}_1, \boldsymbol{\tau}_1, \boldsymbol{\alpha}_2, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_{n+m}, \boldsymbol{\alpha}_{n+m+1}).$$

If $end(c_1) \neq begin(c_2)$, by definition, $c_2 \circ c_1 = c_1$.

Let $p_1, p_2 \in P_T$ be two paths such that $end(p_1) = begin(p_2)$. The path

$$p_2 \circ p_1 \overset{def}{=} \bar{c}$$

defines the composition of p_1 and p_2 with $c = c_2 \circ c_1$, $c_1 \in p_1$, $c_2 \in p_2$, and $end(c_1) = begin(c_2)$.

If $end(p_1) \neq begin(p_2)$, by definition, $p_2 \circ p_1 = p_1$.

Elementary paths

Let T be a transformation set in an inductive structure $(\Pi, \mathcal{I})$ and $\bar{\boldsymbol{\gamma}} \in \bar{\boldsymbol{\Theta}}$ be a struct. The set of elementary paths from $\bar{\boldsymbol{\gamma}}$ is

$$EP_T(\bar{\boldsymbol{\gamma}}) \overset{def}{=} \bigcup_{\bar{\boldsymbol{\alpha}} \in \bar{\boldsymbol{\Theta}}} P_T^1(\bar{\boldsymbol{\gamma}}, \bar{\boldsymbol{\alpha}}).$$

The transformation for an elementary path $\bar{c} \in EP_T(\bar{\boldsymbol{\gamma}})$, where $c = (\boldsymbol{\gamma}, \boldsymbol{\tau}, \boldsymbol{\alpha})$, is $\bar{\boldsymbol{\tau}} \in \bar{\mathcal{T}}$.

Path embedding

Let $c_1 = (\boldsymbol{\alpha}_1, \boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n, \boldsymbol{\alpha}_{n+1})$ and $c_2 = (\boldsymbol{\beta}_1, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_m, \boldsymbol{\beta}_{m+1})$ be two T-ipaths. If $m = n$ and there exists an istruct $\boldsymbol{\gamma}$ such that $\boldsymbol{\beta}_1 = \boldsymbol{\gamma} \triangleleft \boldsymbol{\alpha}_1$ and $\forall i \in [1, n]$, $\boldsymbol{\delta}_i = \boldsymbol{\tau}_i$, c_1 can be embedded in c_2. This fact is denoted by $c_1 \hookrightarrow c_2$.

Path p_1 can be embedded in path p_2 ($p_1 \hookrightarrow p_2$), if there exist ipaths $c_1 \in p_1$ and $c_2 \in p_2$ such that c_1 can be embedded in c_2.

Transformation system

A weighted transformation set is a pair $WT = (T, l)$, where T is a transformation set and $l : T \to \mathbb{R}_+$ is a mapping.

Let $\bar{\boldsymbol{\kappa}} \in \bar{\Theta}$ be a struct, called progenitor, and let (T, l) be a weighted transformation set with $T = \{\bar{\boldsymbol{\tau}}_1, \bar{\boldsymbol{\tau}}_2, \ldots, \bar{\boldsymbol{\tau}}_m\}$. A transformation system is a triple $\boldsymbol{TS} = (T, l, \bar{\boldsymbol{\kappa}})$.

The set of structs generated by $\boldsymbol{TS}$ is the set

$$TS \overset{def}{=} \{\bar{\gamma} \in \bar{\Theta} | P_T(\bar{\boldsymbol{\kappa}}, \bar{\gamma}) \neq \emptyset\}.$$

Let $\boldsymbol{TS} = (T, l, \bar{\boldsymbol{\kappa}})$ and $c = (\boldsymbol{\alpha}_1, \boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_n, \boldsymbol{\alpha}_{n+1}) \in IP_T$ be respectively a transformation system and an ipath. The number

$$l(\bar{c}) = l(c) \overset{def}{=} l(\boldsymbol{\tau}_1) + \cdots + l(\boldsymbol{\tau}_n)$$

is the duration of ipath c (and path $\bar{c}$).

The generating process $G_{\boldsymbol{TS}}$ is a countable state Markov stochastic process defined as follows:

1. The states of G are elements of the set TS of structs generated by $\boldsymbol{TS}$.

2. The amount of time which G spends in state $\bar{\boldsymbol{\gamma}}$ is a random variable distributed exponentially with mean

$$L = \frac{1}{\sum_{p \in EP_T(\bar{\boldsymbol{\gamma}})} 1/l(p)}.$$

3. When G leaves the state $\bar{\boldsymbol{\gamma}}$, it chooses randomly an elementary path $p \in EP_T(\bar{\boldsymbol{\gamma}})$ with probability

$$\frac{L}{l(p)}.$$

4. All random variables in 2 and 3 are mutually independent.

Typicality measure

Let $\boldsymbol{TS}$ be a transformation system, G, the generating process for $\boldsymbol{TS}$, $E_G(\bar{\boldsymbol{\gamma}})$, the expected time spent by G in state $\bar{\boldsymbol{\gamma}}$, and let $\bar{E_G}$ be defined as:

$$\bar{E_G} \overset{def}{=} \sum_{\bar{\boldsymbol{\gamma}} \in TS} E_G(\bar{\boldsymbol{\gamma}}).$$

$\boldsymbol{TS}$ satisfies the existence condition of the typicality measure if $\bar{E_G}$ is finite. Such a transformation system is called class transformation system or simply class.

C will denote the set of elements of class $\boldsymbol{C}$.

If $\boldsymbol{C}$ is a class, G is the generating process for $\boldsymbol{C}$, and $\bar{\gamma} \in C$, the $\boldsymbol{C}$-typicality measure or typicality is the measure $\nu_{\boldsymbol{C}}$ on $\bar{\Theta}$ defined as

$$\nu_{\boldsymbol{C}}(\bar{\gamma}) \overset{def}{=} \frac{E_G(\bar{\gamma})}{\bar{E_G}}.$$

For a class $\boldsymbol{C} = \boldsymbol{TS}$ and a path $p \in P_{\boldsymbol{C}} \overset{def}{=} P_{\boldsymbol{TS}}$, the probability of p is defined as

$$\mu_C(p) \overset{def}{=} P(G \text{ passes any path } p' \text{ into which } p \text{ can be embedded}).$$

3.3 Example

This example is from [10].

The set of primtypes used to represent rectilinear planar shapes is given in Figure 3.7.

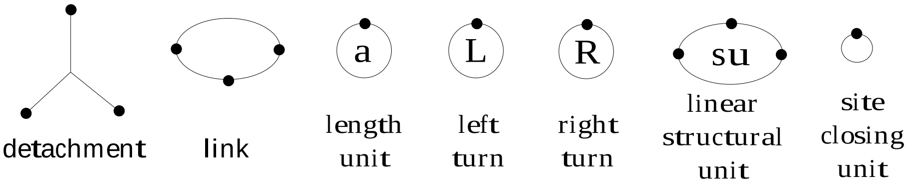

Figure 3.7: The set of primtypes for the class of rectilinear planar shapes.

The class of shapes selected for this example is the class of crosses. Figure 3.8 represents the progenitor of the class of crosses; it can be viewed as a unitary cross.

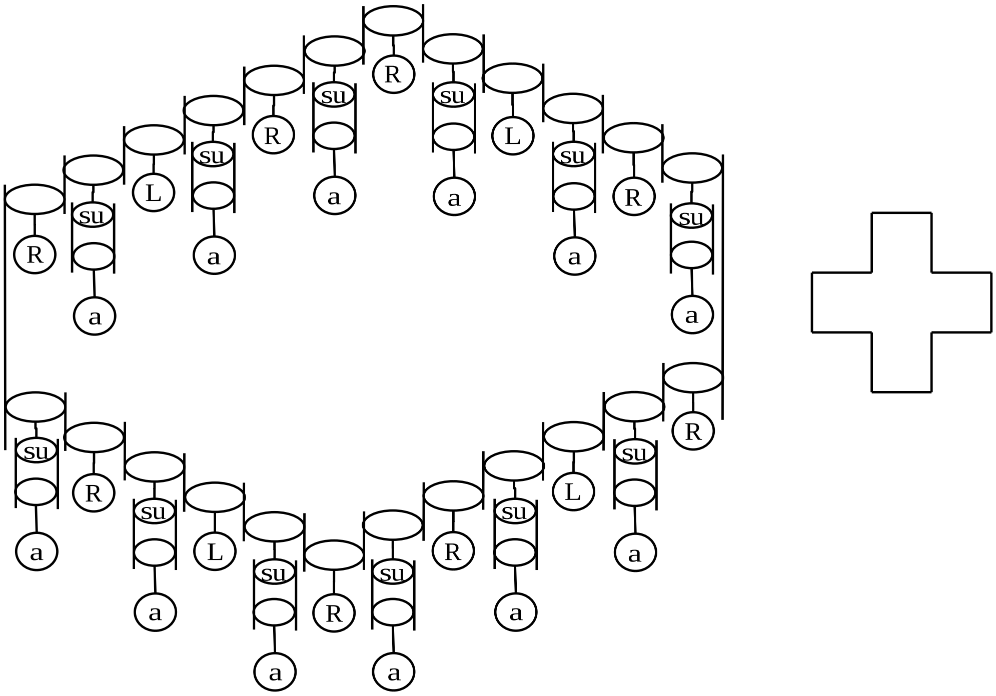

Figure 3.8: The progenitor for the class of crosses.

The transformations are represented in Figure 3.9. The transformation on the left increases the height of a leaf; the one on the right increases simultaneously the width of two opposite leaves.

By applying successively the two transformations, one can generate all the crosses. Figure 3.10 presents an example of a shape from this class.

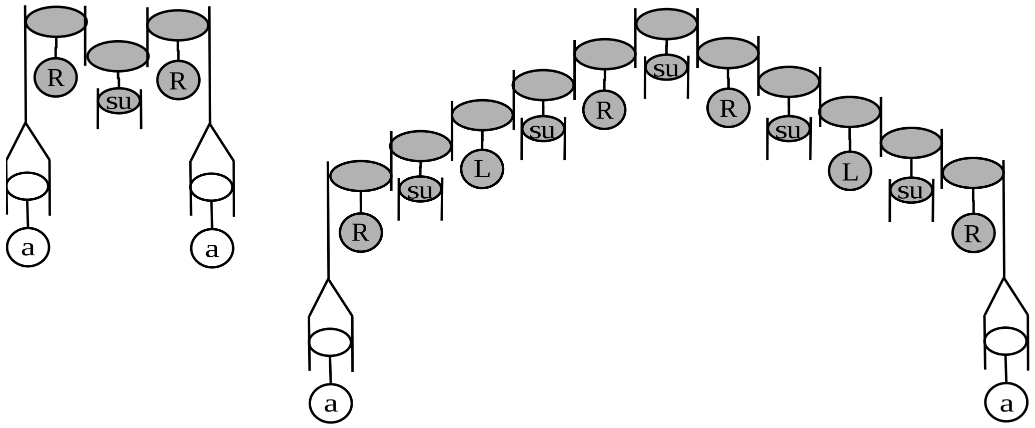

Figure 3.9: The transformations for the class of crosses. The shaded part is the context.

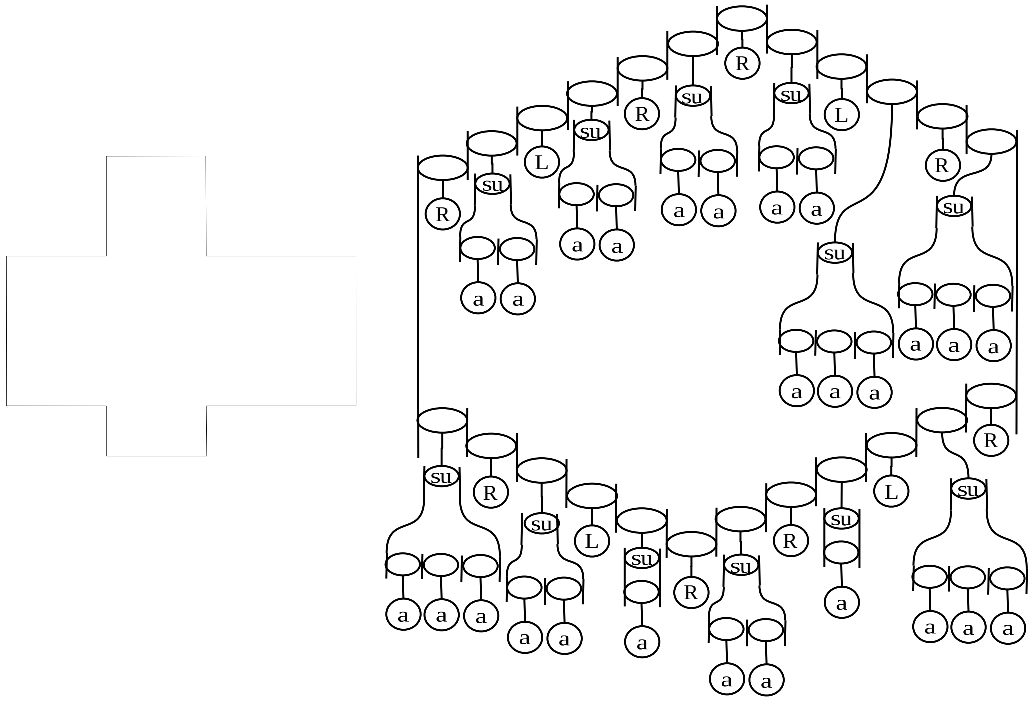

Figure 3.10: A cross (left) and its representation (right).

Chapter 4
The chosen ETS homepage representation

This chapter is dedicated to the description of the ETS representation developed for a particular information retrieval task. The first section introduces the context in which the model will be used. The second section highlights the properties of the corpus (ensemble of documents in which the search is performed) defined for this study. The adaptation of the ETS model to an information retrieval task is finally detailed in the third section. The last section of this chapter provides examples of structural representations created from the developed model.

4.1 Preliminary discussion

A first attempt at a structural information retrieval, based on the ETS model, is presented. For this attempt, the following context was selected. A student is looking for a supervisor to enroll in a graduate program in Computer Science. She or he wants to select his supervisor based on criteria such as the research area and the previous work of the professor; but she or he is also interested in the “relevant” traits of personality of the professor. For example, she or he may want to know how much freedom the supervisor gives to her or his graduate students. Academic homepages have been selected as the means to gather all this information. The corpus was, therefore, defined as the set of the academic homepages of professors working in Computer Science. Each of these homepages is used to extract the academic profile of the professor in a structural ETS form (presented in detail in section 4.3). However the ETS representation produced from the web pages can be completed with extra information obtained from any other source. To find out which professor the student would like to work with, she or he issues a query in an ETS form, *i.e.* the query is either a partial representation, a complete representation, or a set of transformations and a progenitor. A system matches the query with the ETS representations produced from the academic web pages, and a subset is retrieved. This subset is ranked by the typicality of each representation computed either with the set of transformations or within the class of the partial representation given by the student in the query.

4.2 The characteristics of the corpus

Within the corpus, composed of the academic homepages of the professors working in Computer Science, several regularities were observed. Indeed, all professors present nearly identical information (e.g. publications, research projects) in their homepages. Most of the information can be classified into the categories presented below; the most common titles found in the web pages, corresponding to these categories are also quoted.

Education Education, resume, CV, etc.

Extra academic positions Editorial board, professional activities, institute membership, activities, committees, positions, administrative duty, etc.

Research projects Research interests, current projects, curriculum projects, research projects, etc.

Publications Recent publications, papers, technical reports, recent books, books, etc.

Courses Recent courses, current courses, teaching, recent news, etc.

Students Student research opportunities, students, theses supervised, prospective students, etc.

Extra professional information Biography, biographical and personal, family, hobbies, musical interest, non-professional activities, etc.

Research group Affiliation, research groups, laboratories, etc.

Awards Awards, honors, etc.

Talks Recent talks, debate contribution, conference participation, etc.

Funding Recent funding, grants, etc.

Links Funny links, links, pictures, etc.

Other information such as the patents of the professor can be present, but they are less common. These categories were used as the basis for the ETS representations of the academic homepages.

4.3 ETS homepage representation

An ETS representation of the professional profile of the professors based on their homepages has been developed. A complete representation is made up of six placeholders (present in each representation), and a list of attributes that can be attached either to the placeholders or to the other attributes. The placeholders are described first, then the attributes, and finally the possible attachments.

4.3.1 Placeholders

Six placeholders are present in each representation. These placeholders are used for ease of construction and understanding of the ETS representation and are present in each representation. These placeholders are illustrated in Figure 4.1.

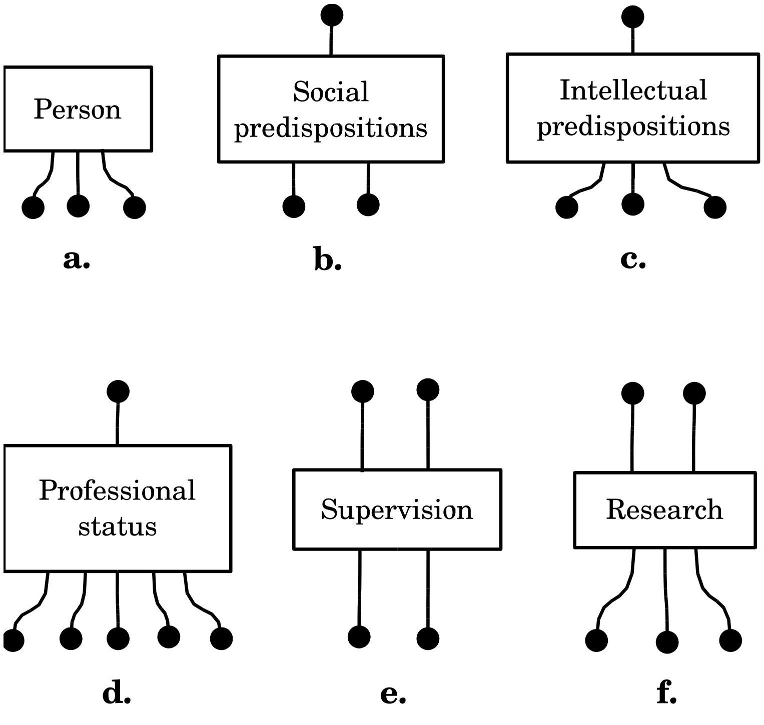

Figure 4.1: Primtypes used as placeholders.

Person :

The placeholder **Person** is used as a root for the representation (Figure 4.1 a.). Since the structural representation of the academic homepage characterizes the academic profile of the professor, all other primitives are connected to it, directly or indirectly. Three placeholders, which describe three aspects of the professional profile of the professor, are directly attached to the placeholder **Person**; they correspond to the placeholder **Social predispositions**, the placeholder **Intellectual predispositions**, and the placeholder **Professional status**.

Social predispositions :

The attributes attached to the placeholder **Social predispositions** characterize the intrinsic qualities of the person in his relations with others (Figure 4.1 b.). For example, a professor can be open and enjoy relations with other people or can be more reserved and prefer to have only a few contacts.

Intellectual predispositions :

The attributes attached to the placeholder **Intellectual predispositions** represent the intellectual predispositions of the professor (Figure 4.1 c.). The professor can be either an abstract or a concrete thinker, she or he can prefer working on interdisciplinary subjects or investigating a subject in depth.

Professional status :

The attributes attached to the placeholder **Professional status** deal with the career choices of the professor (Figure 4.1 d.). This includes research and supervision.

Supervision style :

The placeholder **Supervision style** is attached to the placeholder **Professional status** (Figure 4.1 e.). The attributes attached to the placeholder **Supervision style** deal with the relationships between the professor as supervisor and her or his graduate students. This relation is characterized by the freedom she or he wants to give to her or his students. Her or his experience is also taken into account.

Research :

The placeholder **Research** is attached to the placeholder **Professional status** (Figure 4.1 f.). The attributes attached to the placeholder **Research** describe the research work of the professor. They express the importance of the research for the professor and the way she or he carries out her or his research.

4.3.2 Attributes

This section presents the attributes used in the ETS representation of the homepages. Figure 4.2 illustrates these attributes. They are represented as primtypes attached either to the placeholders or between themselves. Words have been selected to represent these attributes. However, the meanings of these words are subject to different interpretations. The descriptions below are given to remove the possible ambiguity.

In the description of the attributes, those characterizing the same aspect of the professional profile of the professor are presented together. The one that best fits the professor will be selected and attached in the structural representation.

Casual / Formal :

These attributes describe the rigidity of the professor with regard to the establishment.

Casual: A casual person shows little concern with the rules and prefers to behave in harmony with her or his feelings (Figure 4.2 a.).

Formal: A formal person always follows the established customs and rules. She or he is very strict in her or his relations with others (Figure 4.2 b.).

> This information can be extracted from the web page by the formality of the professor's language, ideas and page layout.

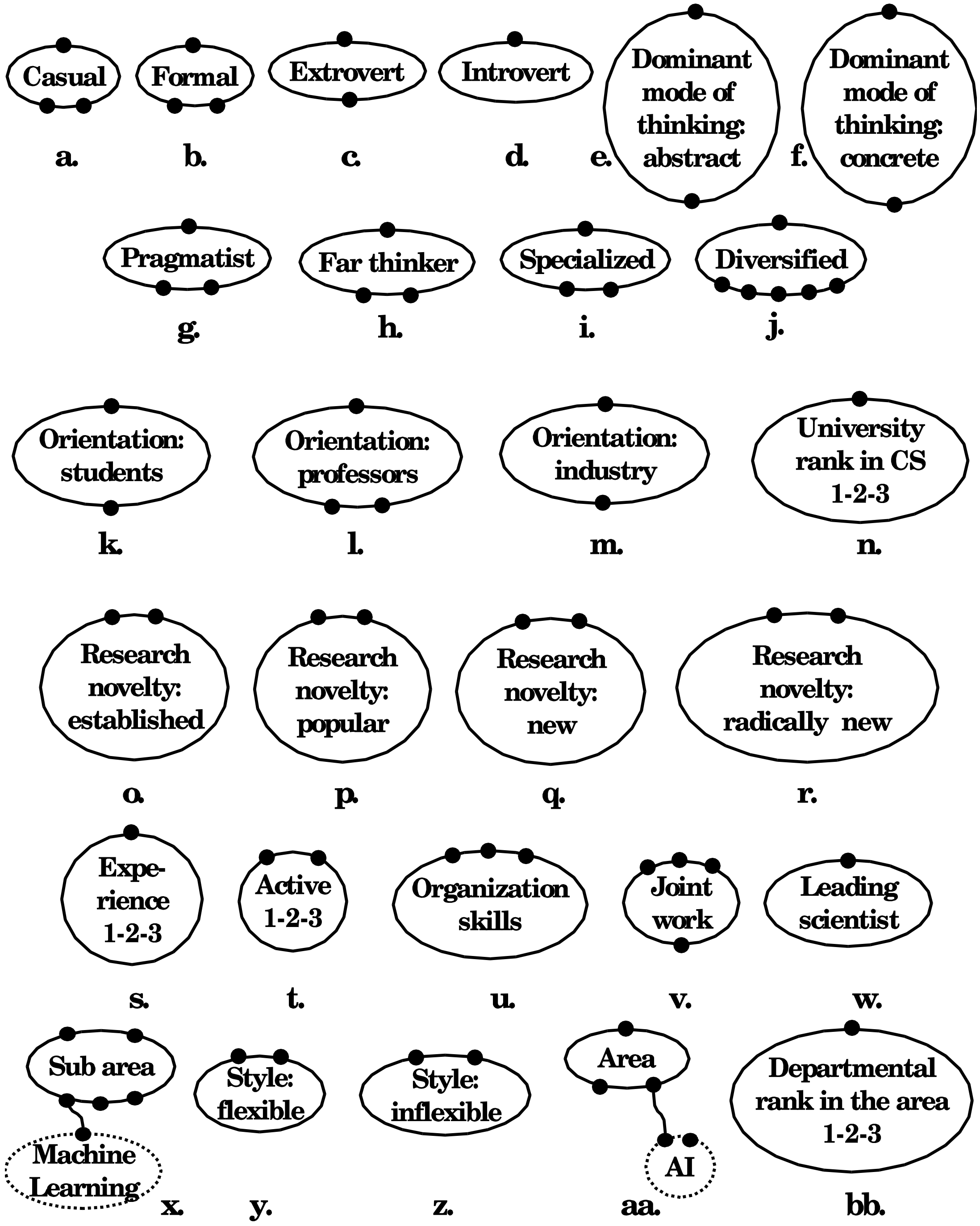

Figure 4.2: Primtypes used in the representation.

Extrovert / Introvert :

People can be interested in their own thoughts or in others. This leads to two definitions (Oxford Dictionary):

Extrovert: A person predominantly concerned with external things or objective considerations (Figure 4.2 c.).

Introvert: A person predominantly concerned with her or his own thoughts and feelings rather than with external things (Figure 4.2 d.).

> Some indicators to decide whether the professor is more extroverted or introverted are the presence or absence of personal information, and/or information about the family or friends of the professor, etc.

Dominant mode of thinking: Abstract / Concrete :

Some professors prefer working on concrete things, like software engineering, whereas others prefer working on more abstract things, like the development of theoretical models.

Abstract thinking characterizes professors who prefer working on things theoretical or disassociated from any specific instance (Figure 4.2 e.).

Concrete thinking characterizes people who prefer working on concrete things, i.e. belonging to immediate experience of actual things or events (Figure 4.2 f.).

> The area in which the professor studies and her or his research work are good indicators of the professor's mode of thinking.

Pragmatist / Far-thinker :

Pragmatist: A pragmatist works on short-term applications. Her or his approach to problems is very practical. She or he works to earn quick results and renown (Figure 4.2 g.).

Far-thinker: The person has a long-range vision. She or he has a pioneering role and tries to work for things that may not have any application for several decades.The person thinks that she or he has a part to play in Science. She or he works to make Science progress.

> This aspect of the professional profile of the professor can be retrieved from the titles of the publications, or/and in a paragraph dedicated to the presentation of her or his research work.

Specialized / Diversified :

These attributes characterize the diversification of the professor's work: she or he may prefer having a broad area of study, may be interested in various research topics, or may be specialized in a narrowly defined area (Figure 4.2 i and j.).

> Whether the professor is specialized or diversified can be deduced from the web pages by analyzing the different research interests of the person and by analyzing the titles of the publications.

Orientation: Students / Professors / Industry :

This feature expresses whether the professor is interested in working with students (Figure 4.2 k.), other professors (Figure 4.2 l.), or industry (Figure 4.2 m.). One, two, or three of these attributes can be attached to the structural representation. It is assumed that the orientation of the professor can be deduced from the homepage: for example, if the homepage is designed particularly to interest students, the orientation of the professor will be towards the students.

The orientation towards students can be inferred, for example, from the homepage by the presence of a list of thesis topics, of a list of present or past students, or of a paragraph explaining the requirements to be a graduate student.

A particular emphasis on the research work, on the publications, on the previous collaborations with other universities denotes an orientation towards professors.

The orientation towards industry can be deduced, for example, by the presence on the web page of one or more laboratory logos, by the description of the work done with the laboratories, or by a biography written in the third person singular.

University rank in CS :

This attribute expresses the rank of the university in Computer Science. The value 3 represents top or leading universities, the value 2 represents good or renowned universities, and the value 1 represents universities that cannot be classified into either of the previous categories (Figure 4.2 n.).

This classification can be done statically for all universities. Since the number of universities in the two first categories is finite and relatively small, these two sets can be built. The lists of the top and of the good universities can then be used to find the university rank. Knowing the name of the university, this name is searched in both lists. If it is found in one of the lists, the university rank is the rank of the list (levels 2 and 3). If it cannot be found, it is classified as “other” (level 1).

Research novelty: Established / Popular / New / Radically new :

This part of the representation characterizes the domain in which the professor does her or his research: four levels were defined.

Established: The research area is now well known. Many research works have been published (Figure 4.2 o.).

Popular: A lot of research has already been done in this area but more needs to be done. The interest in this area is still growing (Figure 4.2 p.).

New: Only a few research works have been published in this area (Figure 4.2 q.).

Radically new: The professor develops a completely new approach in this area (Figure 4.2 r.).

This information can be retrieved from the web page using the titles of the publications and the presentation of the professor's research work. A clustering technique can be used in combination with a static list. A static list can be created for the established and popular research areas. A clustering algorithm can be used to differentiate the new and radically new research. Radically new research will use some terms or term associations that are unusual.

Experience :

The attribute **Experience** is used to express the status of the professor: whether she or he is experienced or not. The degree of experience is used for two domains: the experience in research and the experience in supervision (Figure 4.2 s.). Three levels were defined:

- Level 1: The professor is new (in her or his domain or in supervision). She or he has no or little experience.
- Level 2: The professor has some experience.
- Level 3: The professor is quite experienced. She or he has been working in her or his domain or with graduate students for years.

In research, the level of experience is inferred from the date of the degree of the professor, and/or the number of articles she or he has published in a given domain. In supervision, the level of experience can be defined using the number of theses the professor has supervised.

Active :

The attribute **Active** represents the importance that the professor attaches to her or his research work. The value 1 will be assigned to a professor who is not really involved in research, whereas the value 3 will be used for a professor dedicated to her or his research work (Figure 4.2 t.).

The publications are the best indicator to find out the importance the professor attaches to his research work. Only the publications of the last two or three years are used to perform this measure. A threshold expressed as a number of publications per year can be used to define each level.

Organization skills :

The attribute **Organization skills** is present in the representation if the professor is skilled at organizing either her or his work, her or his ideas, or the work of a team (Figure 4.2 u.).

The organizational skills of the professor appear in the web pages in the layout of the ideas. It appears in the publications as well; the way they are organized can be an indicator. If the professor has an administrative position, it may be deduced that she or he has organizational skills.

Joint work :

This attribute is attached to the representation when the professor collaborates with other people (Figure 4.2 v.).

It can be determined whether the professor is involved in joint work or not by examining the list of publications, for example. If the professor is co-author with other professors, it can be assumed that she or he often does joint work. Another clue is presence or absence of links on her or his homepage to some laboratories in which she or he works.

Leading scientist :

The attribute **Leading scientist** is attached to the structural representation if the professor has initiated her or his domain of research or given a new direction in her or his area (Figure 4.2 w.). A leading scientist has been successful because she or he has developed a new and interesting theory for example.

Leading scientists are usually well known in their area.

Area :

This attribute is related to the research area of the professor (Figure 4.2 aa.). For this study, five areas corresponding to **Artificial Intelligence (AI)**, **Theory**, **Hardware**, **Software Engineering**, and **Systems** were defined. These five areas are broad enough to classify all the homepages studied. Only one area is associated to the structural representation of the academic homepage of the professor.

A list of keywords is defined for each area. The homepages are then classified, depending on the presence or absence of the various keywords.

Sub-area :

This is the sub-area in which the professor works (Figure 4.2 x.). This sub-area has to be broad enough to be used for several professors. For example, a good sub area for AI is machine learning. A professor can work in several sub-areas.

A list of keywords for each sub-area can be defined. Finding the sub-areas can be done by searching these keywords in the web page and then by associating the most relevant sub-areas to each web page.

Style: Flexible / Inflexible :

This attribute expresses the freedom the professor gives to her or his graduate students. The professor may let the student select the direction in which she or he wants to lead her or his research work (Figure 4.2 y.) or she or he may want the student to follow exactly her or his instruction to achieve her or his research work (Figure 4.2 z.). These attributes are not compulsary.

The flexibility of the professor can appear in the enumeration of the requirements needed by her or his future graduate students, in the description of the projects proposed to the students, or in both.

Departmental rank in the area :

This section represents the reputation of the Computer Science Department in the area in which the professor works (Figure 4.2 bb.). Three integers are used to characterize the departmental rank in the area: the value 1 is assigned to an unknown department in the area, the value 2 to a department with a good reputation in the area, and the value 3 to a leading department in the area.

The values can be assigned statically: a list of the leading (criterion 3) and known departments (criterion 2) for each area can be used to classify all of the departments.

Some of the classifications described previously are subjective and cannot be determined with accuracy. Further studies will be needed to automate this extraction of information. To have more objective classifications, the criteria should be chosen more objective, or the information should be completed by other sources.

4.3.3 Attachments between placeholders and attributes

The attachments between the different primitives have been defined to highlight the precedence of a primitive on the ones attached to it. A primitive will be attached to another if it is a consequence of the previous, or if it is less important and both characterize the same aspect of the professional profile, or if they convey a meaning by their association. Figure 4.6 presents the layout of the primitives with all of their possible attachments. By convention, the attachment illustrated in Figure 4.3 a. means that θ is either attached to π or ρ but not both; the attachment illustrated in Figure 4.3 b. means that either π or ρ can be attached to σ but not both.

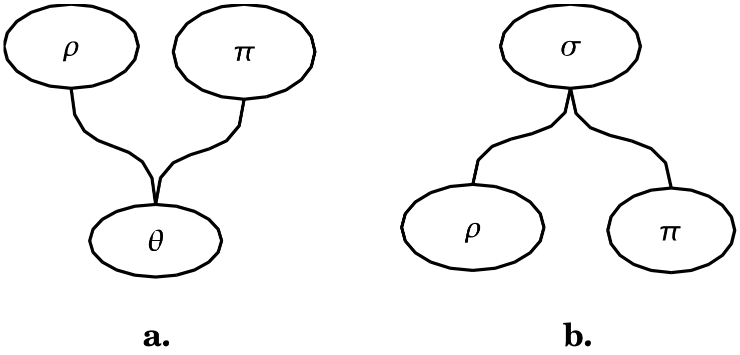

Figure 4.3: Representations of the primitives when two different attachments are possible.

The attachment illustrated in Figure 4.4 means that the three primitives **Orientation: students**, **Orientation: professors**, and **Orientation: industry** can be attached together regardless of the order but that if another primitive has to be attached to this group, it will be attached to the primitive **Orientation: professors**.

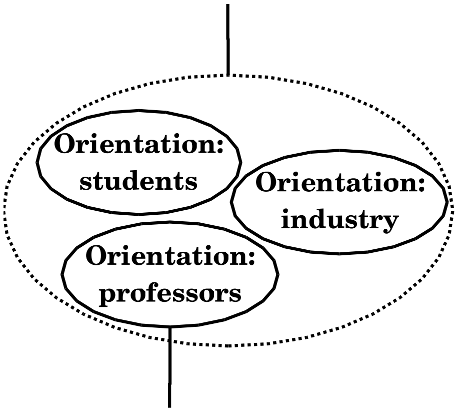

Figure 4.4: Representations of the primitives when each order of attachment exists.

The attachment shown in Figure 4.5 a. is used for ease of representation. The dotted line means that several sub-areas can be attached one by one in the structural representation. To the last sub-area is attached the primitive area. One of the five defined areas (*i.e.*, **AI**, **Theory**, **Hardware**, **Software Engineering**, or **Systems**) is then attached to the primitive **Area**. To simplify the representation, in Figures 4.5 a. and 4.6, only the area **Software Engineering** is presented (dotted ellipse). An actual attachment with all the sub-areas represented will be drawn as in Figure 4.5 b.

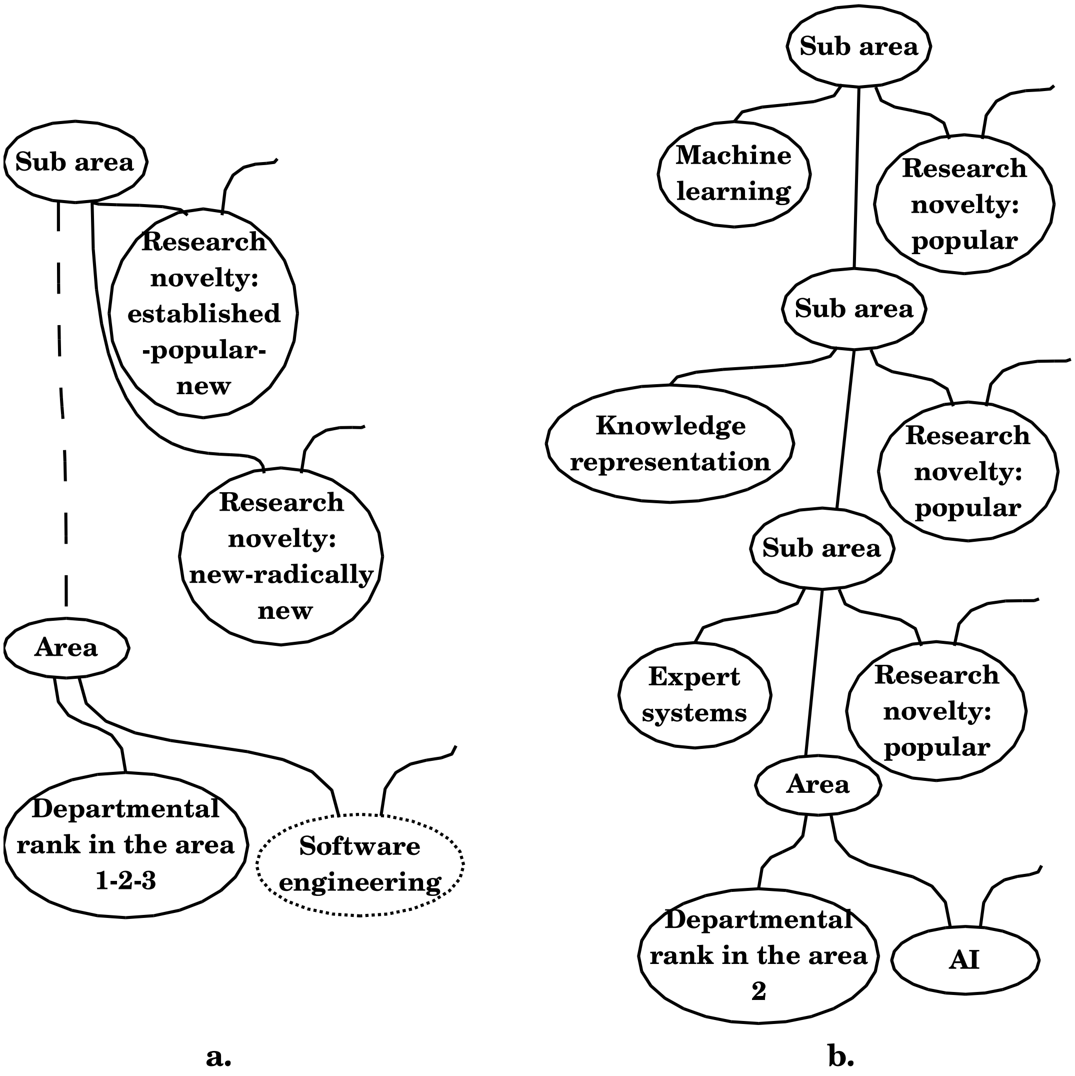
Sub area
Research novelty: established -popular- new
Research novelty: new-radically new
Area
Departmental rank in the area 1-2-3
Software engineering
a.
Sub area
Machine learning
Research novelty: popular
Sub area
Knowledge representation
Research novelty: popular
Sub area
Expert systems
Research novelty: popular
Area
Departmental rank in the area 2
AI
b.

Figure 4.5: A representation of the sub-areas in a modeled form (a.) and in a practical form (b.).

4.4 Examples

The ETS representations of approximately fifty homepages have been built. The academic homepages have been selected among the web pages of professors in Computer Science working in various Canadian universities. Three representations are presented below.

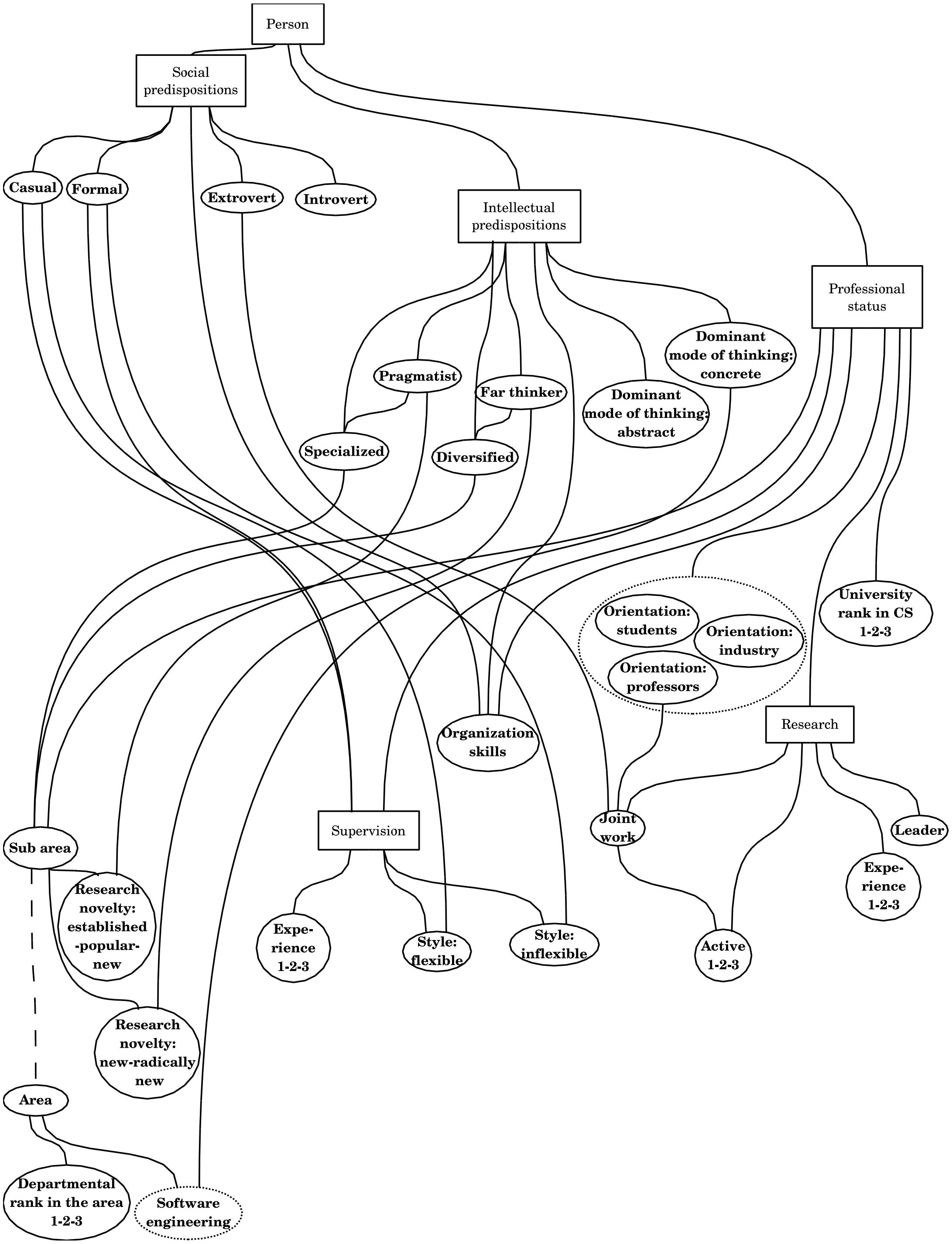

Figure 4.6: All of the primitives and their possible attachments.

Figure 4.7 represents the ETS representation of the homepage of professor Dave Mason at Ryerson Polytechnic University, Toronto. The URL of this homepage is http://www.sarg.ryerson.ca/∼dmason/. This professor is casual and extroverted. He allows some flexibility for his graduate students. He does joint work with other professors, and is very interested in research. He is a concrete thinker, and works mainly on a popular sub area: Software Reliability.

The ETS representation of the homepage of professor Holger Hoos at the University of British Columbia (http://www.cs.ubc.ca/∼hoos/) is presented in Figure 4.8. This professor is more formal and introverted. He is an abstract thinker specialized in AI. Two sub-areas are predominant in his research work: Propositional Satisfiability and Ant Colony Optimization. Both sub-areas are popular. He does joint work with other professors.

The third example (Figure 4.9) is the representation of the homepage of professor Joanne Atlee (http://se.uwaterloo.ca/∼jmatlee/) at the University of Waterloo. This professor is more casual and introverted. She shows great organization skills. She is a concrete thinker specialized in Software Analyses. She is very interested in research work and collaborates with other professors.

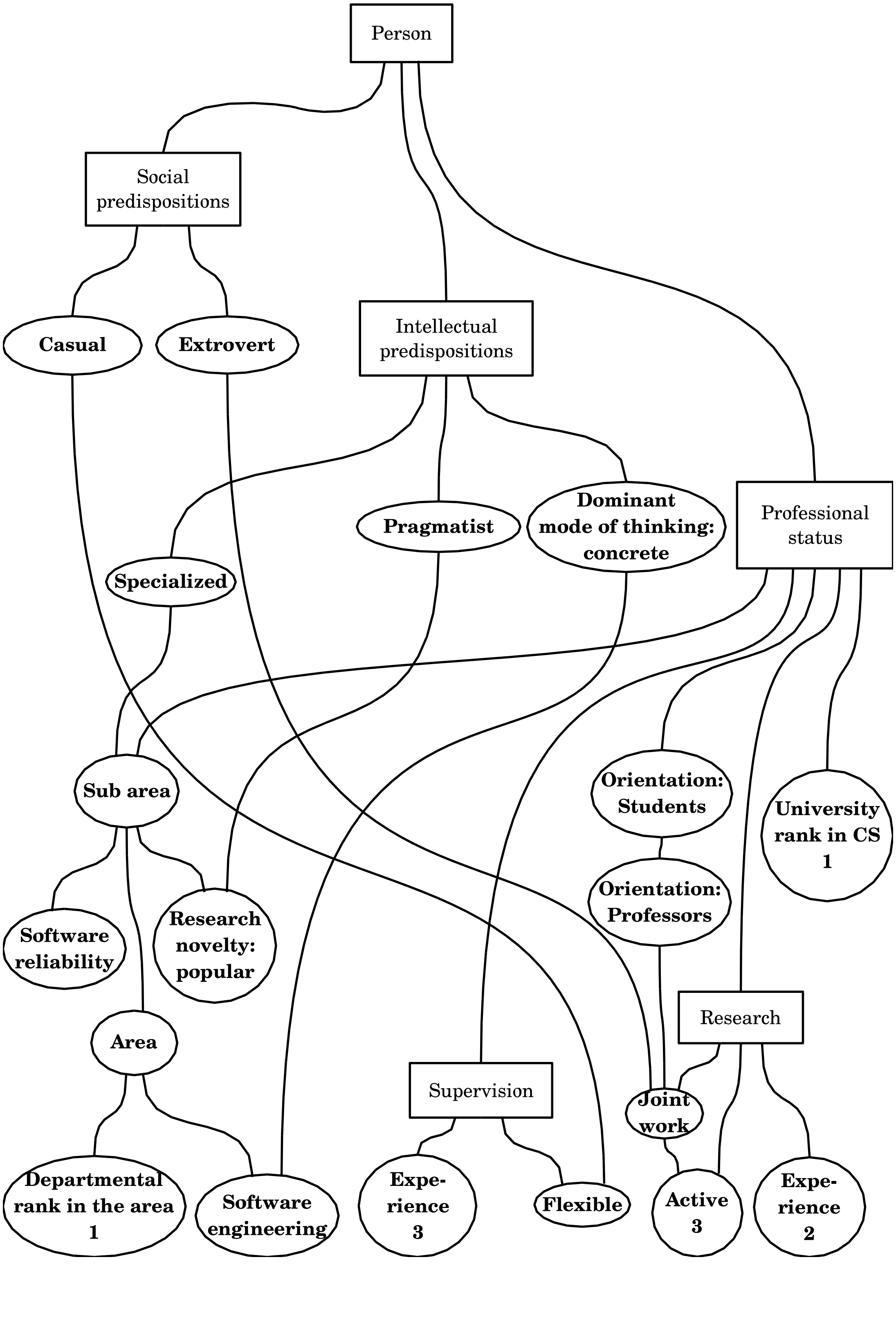

Figure 4.7: Representation of the homepage *http://www.sarg.ryerson.ca/∼dmason/*.

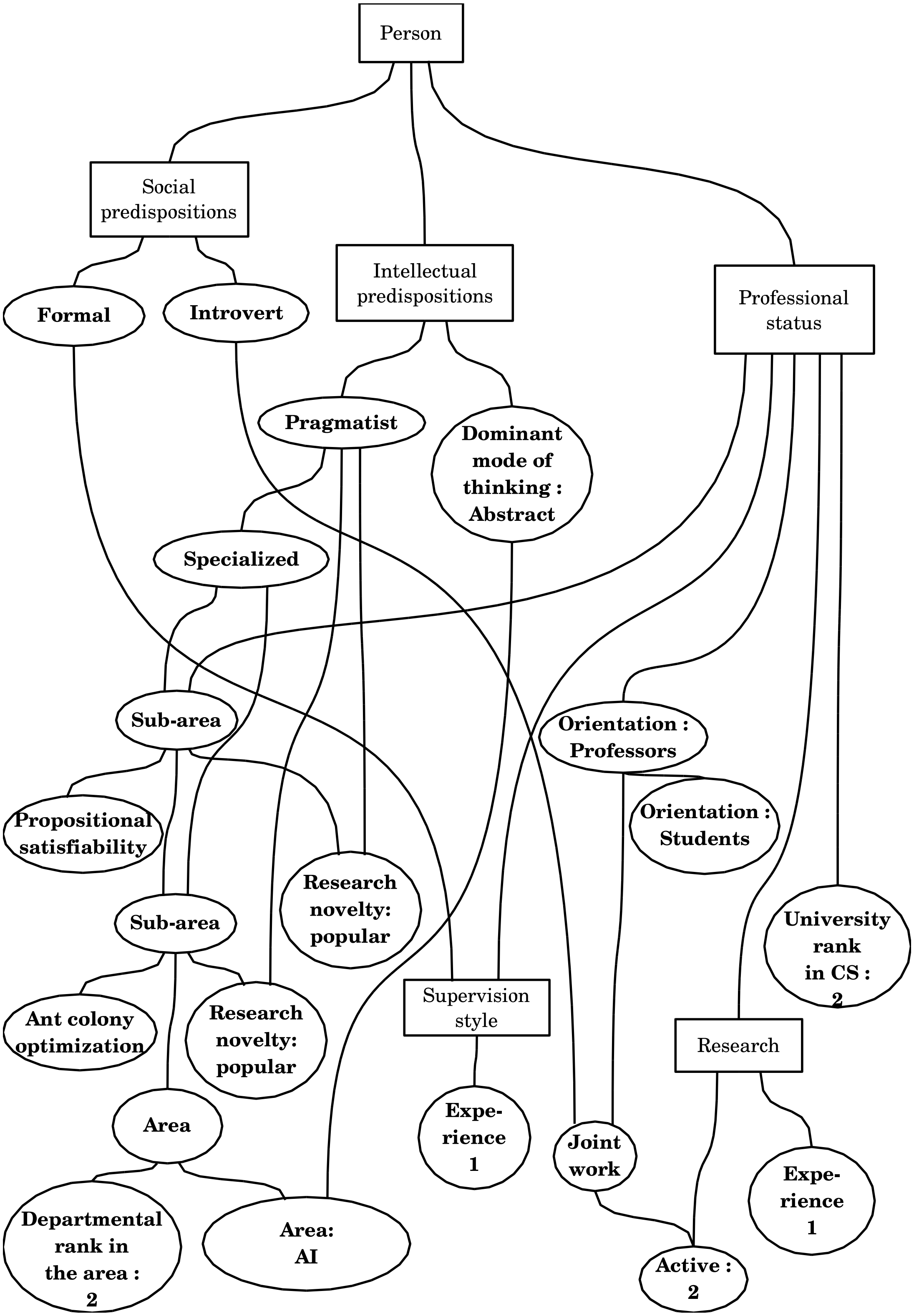

Figure 4.8: Representation of the homepage *http://www.cs.ubc.ca/∼hoos/*.

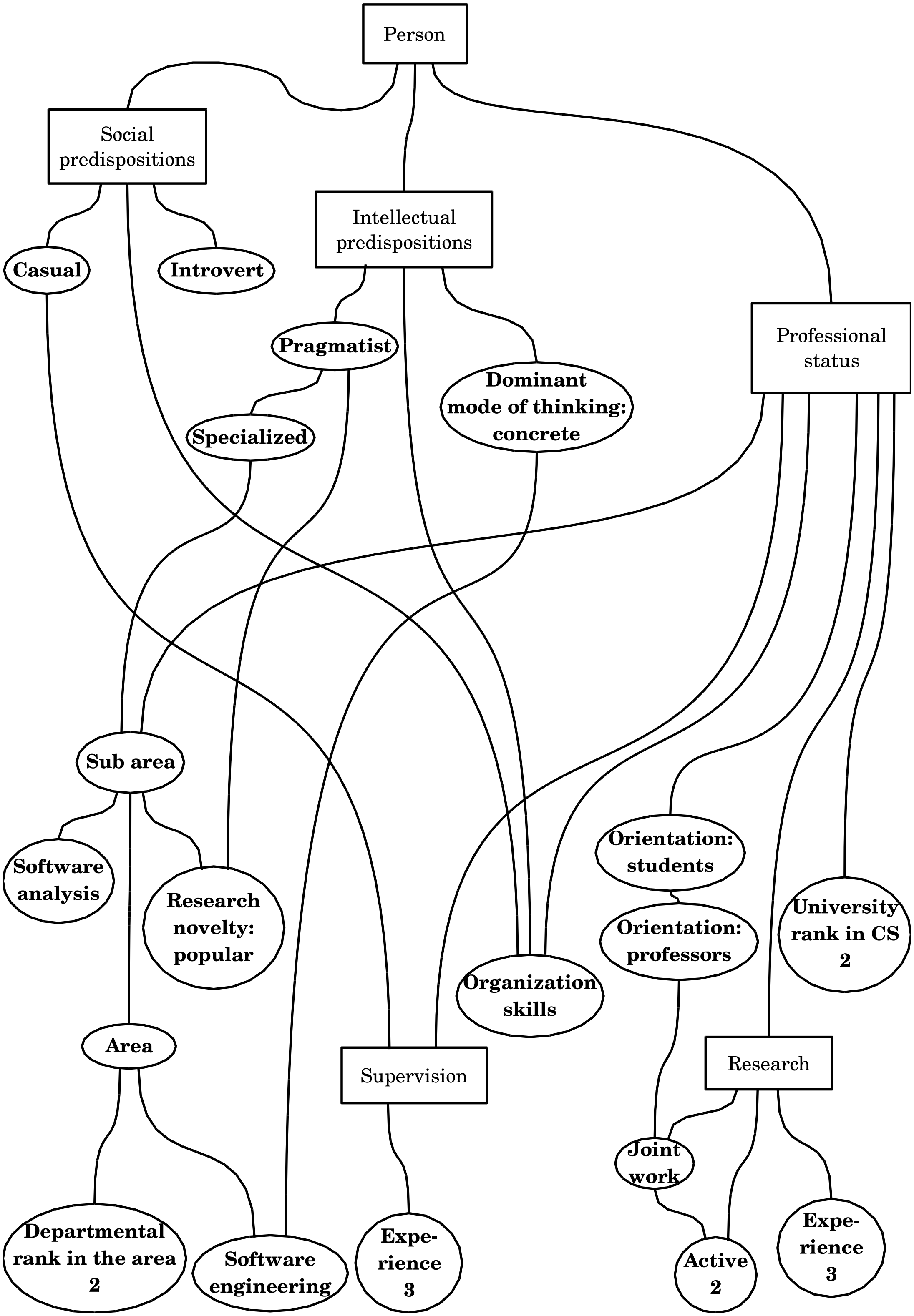

Figure 4.9: Representation of the homepage *http://se.uwaterloo.ca/∼jmatlee/*.

Chapter 5
Classes

The ETS representation of academic homepages allows the highlighting of classes of professors who show similar features in their professional profile. This chapter presents classes built from the ETS representation described in the previous chapter.

In the ETS model, classes are described by their generative process. To describe a class, a progenitor and a set of weighted transformations are required.

A large number of classes can be built. Only one class and two of its sub-classes are presented in this study.

The class selected is the class Γ of professors who are more likely to be successful than are the average professors. Here, "successful" means successful either in their research work, or in their academic career. For example, a professor oriented towards industry will be considered successful if he manages to interest industry, to have a good reputation in the industrial community, and then to obtain funded projects. A professor interested in fundamental research will be successful if he manages to be renown in the scientific community or to develop a new formalism.

5.1 Progenitor

The progenitor of a class is the common ancestor of all of the elements in the class. It is made up of all of the placeholders and attributes common to all elements of the class.

The progenitor $\bar{\boldsymbol{\kappa}}$ of the class Γ (presented in Figure 5.1) contains five placeholders: **Person**, **Social predispositions**, **Intellectual predispositions**, **Professional status**, and **Research**. Diversification is assumed to increase the chances of success of a professor. In the structural model, the attribute **Diversified** is attached to the placeholder **Intellectual predispositions**; this placeholder is already present in the progenitor. The attribute **Diversified** can therefore be attached to the placeholder **Intellectual predispositions** to form the progenitor.

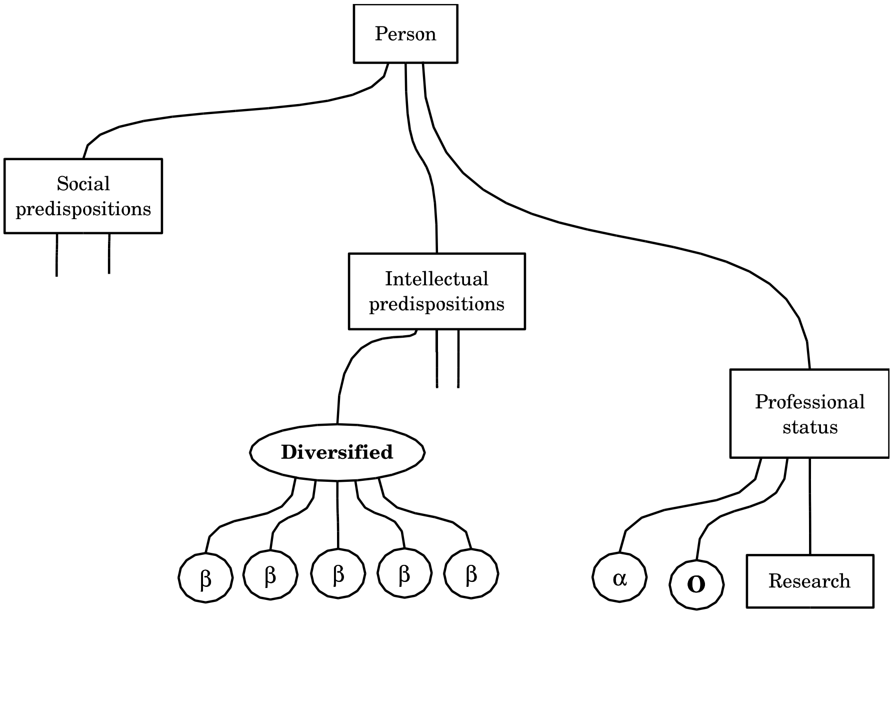

Figure 5.1: Progenitor $\bar{\boldsymbol{\kappa}}$ of class Γ.

The β primitives attached to the attribute **Diversified** are used as contexts for the transformations that attach the sub-areas in which the professor works. A diversified professor can lead research works in a maximum of five sub-areas (if the professor has specialized, a maximum of two sub-areas can be attached). This is why five β primitives are represented in Figure 5.1.

The primitives α and O are used as contexts for transformations that *chain* other attributes characterizing the professional profile of the professors of the class. A transformation that attaches an attribute to the primitive α or O will also attach α or O to the added attribute. The context (α or O) is therefore still present after the use of a transformation; this context allows successive applications of transformations.

Chaining also shows an order of importance among the primitives. Because the context is re-created, the transformations, which are used to chain the elements, can be applied in different orders. These orders can, therefore, reflect the relative importance of each of the primitives.

5.2 Transformations

A class is defined by a progenitor and a set of weighted transformations. The set of transformations for the class Γ is presented in Figures 5.2, 5.3, and 5.4. The context required to apply the transformations is represented by the shaded primitives.

The transformations τ_1 to τ_4 attach attributes that characterize the relationship

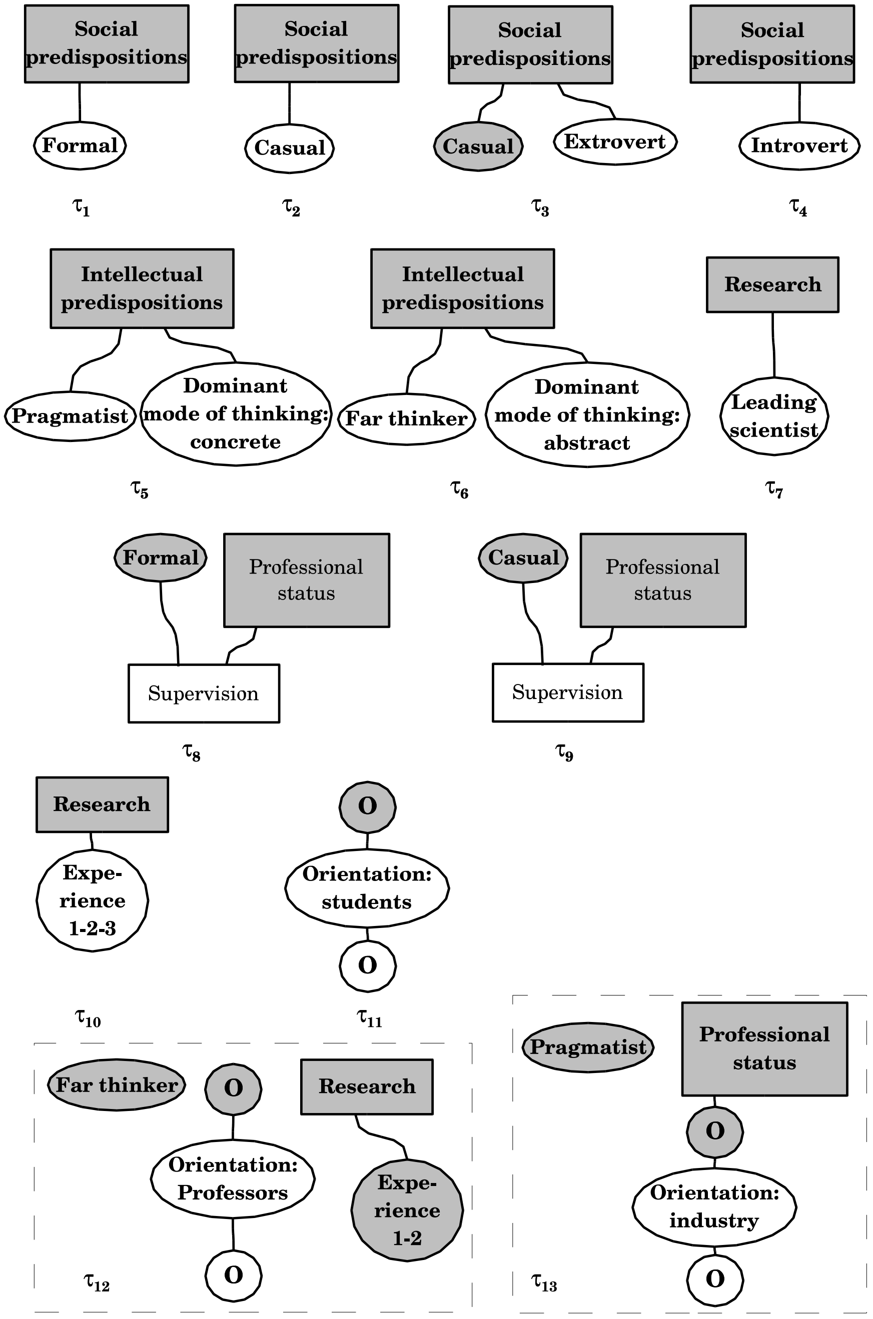

Figure 5.2: Transformations for the class Γ.

between the professor and other people. It was presumed that this aspect of the professional profile of the professor was not involved in determining class membership. All of the attributes characterizing this aspect of the professional

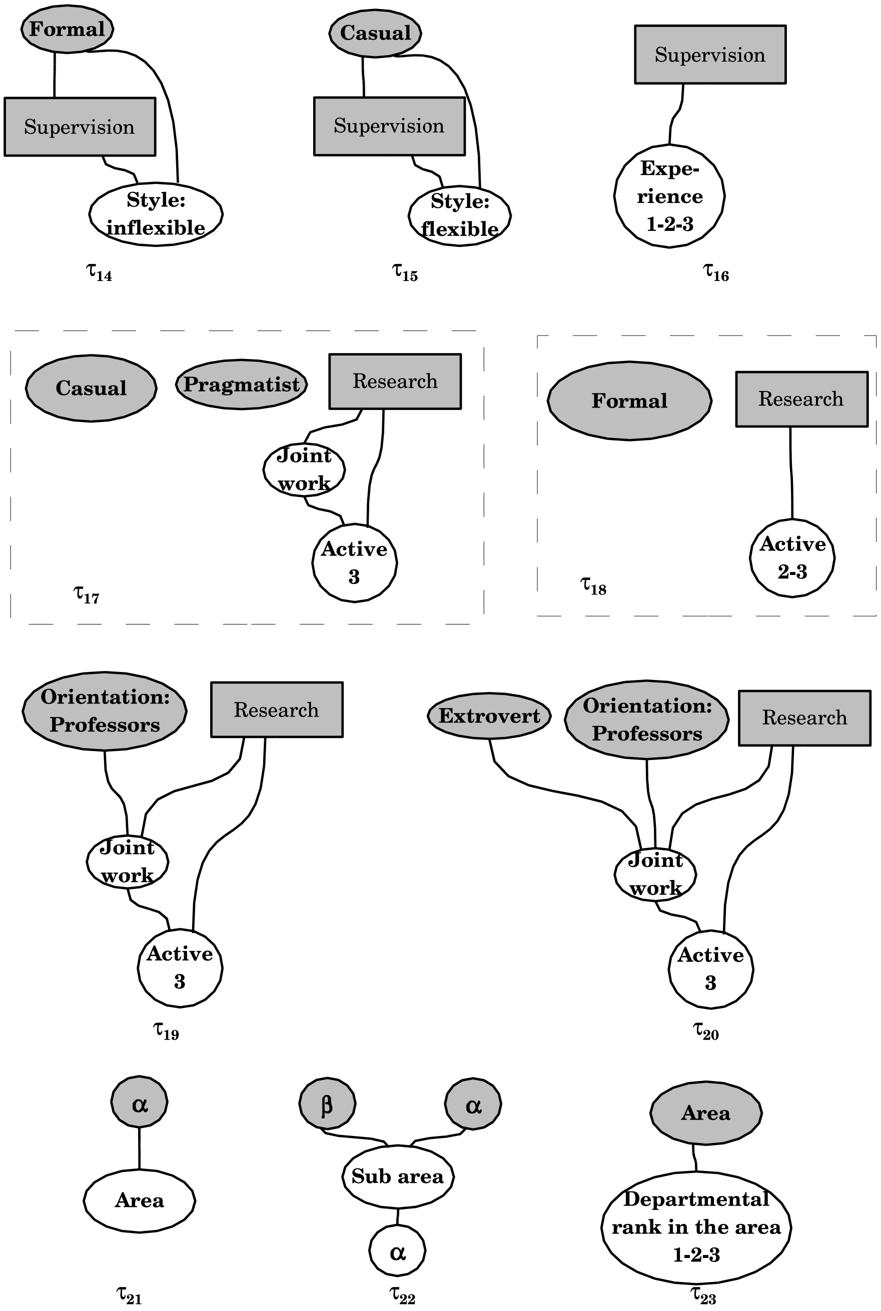

Figure 5.3: Transformations for the class Γ (cont.).

profile of the professor can, therefore, be attached. However no transformation allows the attachment of the attributes **Formal** and **Extrovert** to the same element because it is unlikely to find those features combined in the same person.

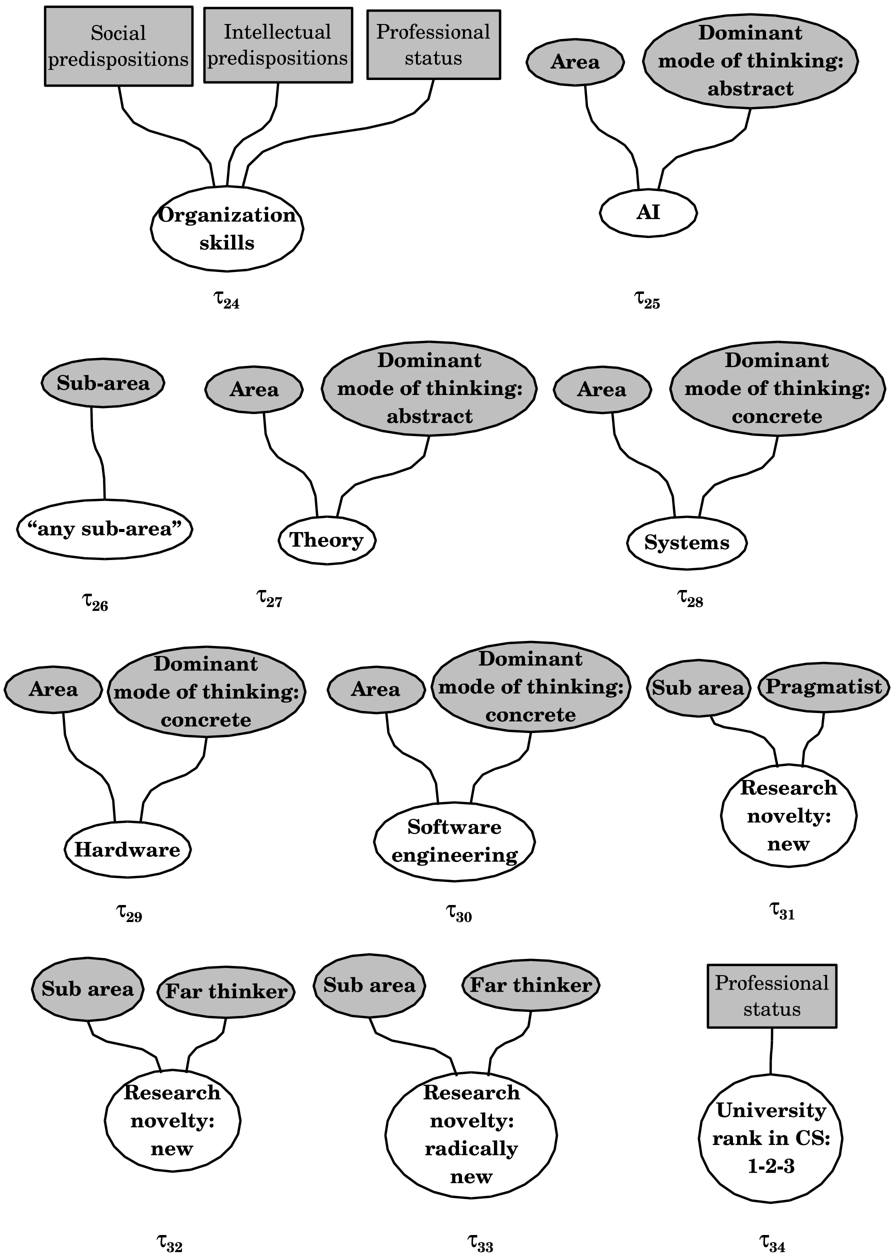

Figure 5.4: Transformations for the class Γ (cont.).

Transformations τ_5 or τ_6 are involved in determining class membership. It is presumed that the professor has a large chance of being successful either if she or he is a far-thinker and has a dominant mode of thinking which is abstract, or if she or he is a pragmatist and has a dominant mode of thinking which is concrete.

Transformation τ_7 attaches the attribute **Leading scientist**. This means that the professor is successful in her or his research work. This transformation determines class membership. If an element has this attribute, it clearly belongs to the class.

Transformations τ_8 and τ_9 add the placeholder **Supervision**.

Transformation τ_{10} attaches the degree of research experience of the professor. In reality, τ_{10} represents three different transformations. τ_{10-1} is the transformation τ_{10} with a degree of experience 1, τ_{10-2} is the transformation τ_{10} with a degree of experience 2, and so on.

Transformation τ_{11} attaches the attribute **Orientation: students**.

Transformation τ_{12} attaches the attribute **Orientation: professor** if the attribute **Far thinker** and the group made up of the placeholder **Research** and the attribute **Experience** with a degree 1 or 2 are already present in the representation. This means that a "successful" professor who is a far thinker and not yet very experienced will certainly be mainly interested in working with other professors to propose her or his opinions to the scientific community. She or he will also do joint work to develop her or his ideas.

Transformation τ_{13} means that a "successful" pragmatic professor will be more interested in working with the industry to complete numerous projects and to gain renown within the industrial community.

Transformations τ_{14} and τ_{15} attach the attributes characterizing the supervision style of the professor. It was presumed that a formal professor would be more likely to be inflexible whereas a casual professor will be more likely to be flexible.

Transformation τ_{16} represents three transformations (τ_{16-1}, τ_{16-2}, τ_{16-3}). Each of these transformations characterizes a degree of experience: 1, 2, or 3. This is done in a manner similar to transformation τ_{10}, above.

Transformation τ_{17} means that a "successful" professor who is casual and pragmatic will probably do joint work and be very active.

Transformation τ_{18} means that a "successful" formal professor will certainly prefer spending more time working on her or his own; she or he will probably not collaborate, but she or he will be active.

Transformations τ_{19} and τ_{20} mean that a "successful" professor oriented toward professors will certainly do joint work and be very active. Moreover, if she or he is an extrovert, it may have an effect on her or his joint work.

Transformations τ_{21} and τ_{22} add the **Area** and the **sub-areas**.

Transformation τ_{23} attaches the departmental rank in the area. Because the rank can be equal to 1, 2, or 3, the respective transformations are τ_{23-1}, τ_{23-2}, and τ_{23-3}.

Transformation τ_{24} adds the organizational skills of the professor. This attribute will be added if the professor is skillful in organizing either her or his work or her or his students' projects.

Transformations τ_{25}, τ_{27}, τ_{28}, τ_{29} and τ_{30} add the area in which the professor works.

Transformation τ_{26} attaches the sub areas in which the professor works.

Transformations τ_{31}, τ_{32} and τ_{33} attach the attributes characterizing the novelty of the work of the professor in each sub-area.

Transformation τ_{34} attaches the rank of the university in Computer Science.

Figure 5.5 shows the element $\bar{\gamma}$ of Γ defined as:

$$\begin{aligned}\bar{\boldsymbol{\gamma}} = \; & \bar{\boldsymbol{\kappa}} \triangleleft \tau_2 \triangleleft \tau_4 \triangleleft \tau_6 \triangleleft \tau_{10-1} \triangleleft \tau_{12} \triangleleft \tau_9 \triangleleft \tau_{16-1} \triangleleft \tau_{19} \triangleleft \tau_{22} \triangleleft \tau_{26-\text{Machine learning}} \triangleleft \tau_{33} \triangleleft \tau_{22} \\ & \triangleleft \tau_{26-\text{Expert systems}} \triangleleft \tau_{32} \triangleleft \tau_{22} \triangleleft \tau_{26-\text{Knowledge representation}} \triangleleft \tau_{32} \triangleleft \tau_{25} \triangleleft \tau_{23-1} \triangleleft \tau_{34-1}\end{aligned}$$

5.3 Typicality

The typicality of an element in a class is computed using the typicality of each of the transformations present in its constructive history. This typicality can be a weighted sum, a product, or any other statistical computation. The computational method used should include the weights of all the transformations in the constructive history, i.e. if two elements differ only by one transformation, one has the transformation α and the other the transformation β, and α and β have different weights, the typicality of the elements should be different.

In this model, the typicality of an element has been defined as the sum of the weights of the transformations applied to the progenitor to obtain this element.

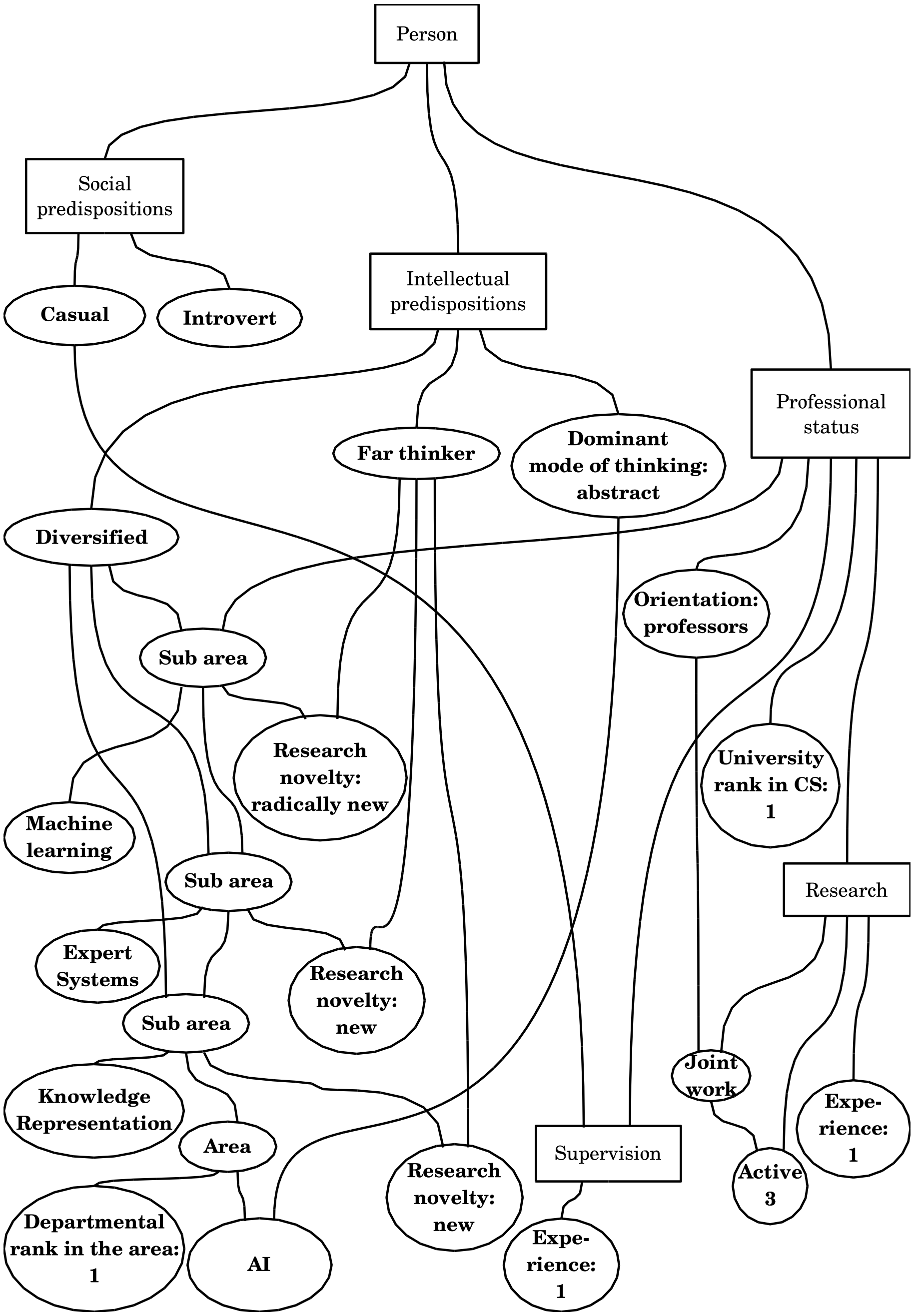

Figure 5.5: An element $\bar{\gamma}$ of Γ.

With this computation of the typicality, an incomplete representation will have a lower typicality than has a full representation. This computation will facilitate the matching part of the retrieval.

A weight or typicality value is assigned to each transformation to describe the importance of this transformation within the class. If a transformation is present in the definition of several classes, the typicality value of this transformation will vary from one class to another. The typicality values have been selected as follows. A weight of approximately 100 will be assigned to the transformations that determine class membership (if such a transformation is present in the constructive history of the element, it is likely that the element will belong to the class). Weights of 30 and 10 will be assigned to the transformations that are respectively very significant and significant for the class. If such a transformation is present in the constructive history of the element, it increases its chance of belonging to the class.

When applying some transformations, it may happen that the elements become less typical within the class or no longer belong to the class. Such transformations are called anti-features. The weight associated with such a transformation is negative.

The transformations which are neither significant nor anti-features of the class are called noise transformations. These transformations will have a weight either positive or negative, that is close to 0.

The weights assigned to the transformations presented in section 5.2 are summarized in Table 5.1.

Transformation	Typicality
τ_1	0.0
τ_2	0.0
τ_3	0.0
τ_4	0.0
τ_5	30.0
τ_6	30.0
τ_7	100.0
τ_8	0.0
τ_9	0.0
τ_{10}	0.0
τ_{11}	10.0
τ_{12}	25.0
τ_{13}	35.0
τ_{14}	25.0
τ_{15}	15.0
τ_{16}	0.0
τ_{17}	20.0
τ_{18}	10.0
τ_{19}	15.0
τ_{20}	17.0
τ_{21}	0.0
τ_{22}	0.0
τ_{23}	0.0
τ_{24}	30.0
τ_{25}	0.0
τ_{26}	0.0
τ_{27}	0.0
τ_{28}	0.0
τ_{29}	0.0
τ_{30}	0.0
τ_{31}	30.0
τ_{32}	20.0
τ_{33}	30.0
τ_{34}	0.0

Table 5.1: Typicalities of the transformations for the class Γ.

The typicality $\nu(\bar{\gamma})$ of the element $\bar{\gamma}$ (presented in Figure 5.5) belonging to class

Γ is computed as follows (using the notations presented in Chapter 3):

$$\begin{aligned}
\nu(\bar{\gamma}) &= l(\tau_2) + l(\tau_4) + l(\tau_6) + l(\tau_{10-1}) + l(\tau_{12}) + l(\tau_9) + l(\tau_{16-1}) + l(\tau_{19}) \\
&\quad + l(\tau_{22}) + l(\tau_{26-\text{Machine learning}}) + l(\tau_{33}) + l(\tau_{22}) + l(\tau_{26-\text{Expert systems}}) \\
&\quad + l(\tau_{32}) + l(\tau_{22}) + l(\tau_{26-\text{Knowledge representation}}) + l(\tau_{32}) + l(\tau_{25}) \\
&\quad + l(\tau_{23-1}) + l(\tau_{34}) \\
&= 120.0
\end{aligned}$$

Because this number is positive and very large, the element is very typical for the class.

5.4 Sub-classes

Sub-classes are part of the class. A sub-class is defined by a progenitor, which can be the same as the progenitor of the class or a progenitor that includes the progenitor of the class, and a set of transformations. These transformations can be the transformations defining the class (features and noise transformations) or transformations composed by the combinations of several transformations of the class. The weights of these transformations can vary slightly from those of the transformations of the class. The anti-features of the class are also part of the anti-features of the sub-class.

In this section, two sub-classes of the class Γ, presented earlier in this chapter, are detailed. The first sub-class groups the persons who are open-minded. The second sub-class represents highly productive persons.

5.4.1 First example of a sub-class

One of the sub-classes of the class Γ is the sub-class Γ_1 representing people who are open-minded. A person belonging to this sub-class will be more likely to be casual. She or he will enjoy collaboration with other professors. She or he will be flexible when supervising graduate students.

The progenitor $\bar{\boldsymbol{\kappa}}_1$ of this sub-class is presented in Figure 5.6. $\bar{\boldsymbol{\kappa}}_1$ is the same progenitor as $\bar{\boldsymbol{\kappa}}$ with the transformations τ_2 and τ_9 applied.

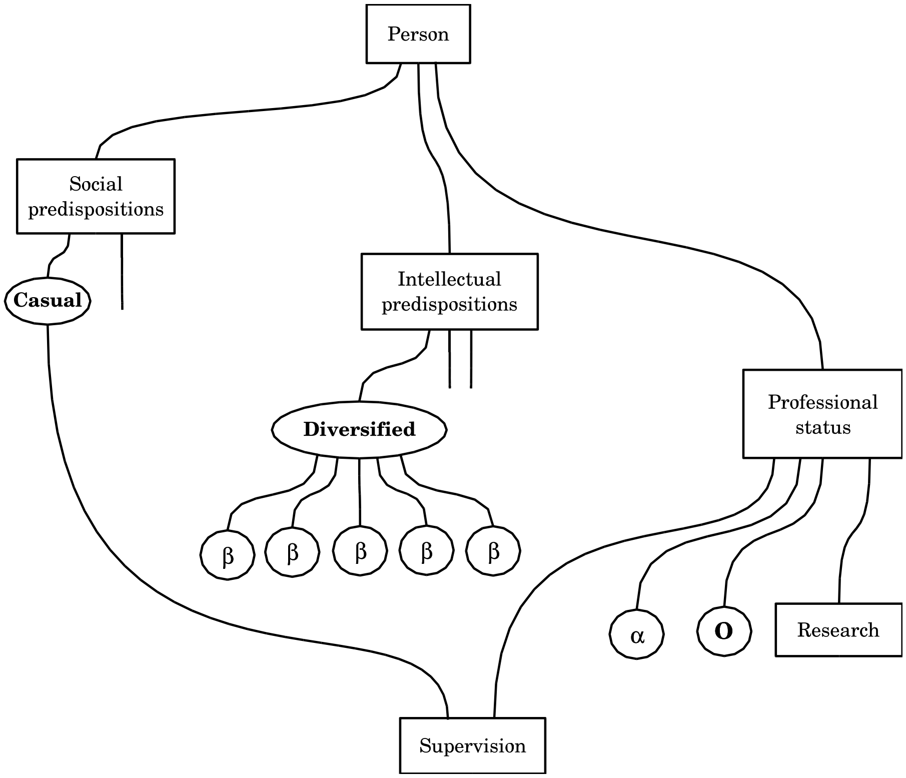

Figure 5.6: Progenitor $\bar{\boldsymbol{\kappa}}_1$ of class Γ_1.

The set of transformations that define the sub-class Γ_1 is as follows:

$$\begin{aligned}\Theta_1 \quad = \quad \{&\tau_3, \tau_4, \tau_5, \tau_7, \tau_{10}, \tau_{11}, \tau_{12}, \tau_{13}, \tau_{14}, \tau_{15}, \tau_{16}, \tau_{17}, \tau_{19}, \tau_{20}, \tau_{21}, \tau_{22}, \tau_{23},\\ &\tau_{24}, \tau_{25}, \tau_{26}, \tau_{27}, \tau_{28}, \tau_{29}, \tau_{30}, \tau_{31}, \tau_{32}, \tau_{33}, \tau_{34}\}\end{aligned}$$

The weights associated with the transformations of this sub-class are summarized in Table 5.2.

Transformation	Typicality
τ_3	10.0
τ_4	0.0
τ_5	20.0
τ_6	50.0
τ_7	100.0
τ_{10}	0.0
τ_{11}	30.0
τ_{12}	25.0
τ_{13}	35.0
τ_{14}	25.0
τ_{15}	5.0
τ_{16}	0.0
τ_{17}	30.0
τ_{19}	25.0
τ_{20}	30.0
τ_{21}	0.0
τ_{22}	0.0
τ_{23}	0.0
τ_{24}	30.0
τ_{25}	0.0
τ_{26}	0.0
τ_{27}	0.0
τ_{28}	0.0
τ_{29}	0.0
τ_{30}	0.0
τ_{31}	30.0
τ_{32}	20.0
τ_{33}	30.0
τ_{34}	0.0

Table 5.2: Typicalities of the transformations for the sub-class Γ_1.

The example shown in Figure 5.7 represents an element $\bar{\gamma}_1$ of the subclass Γ_1

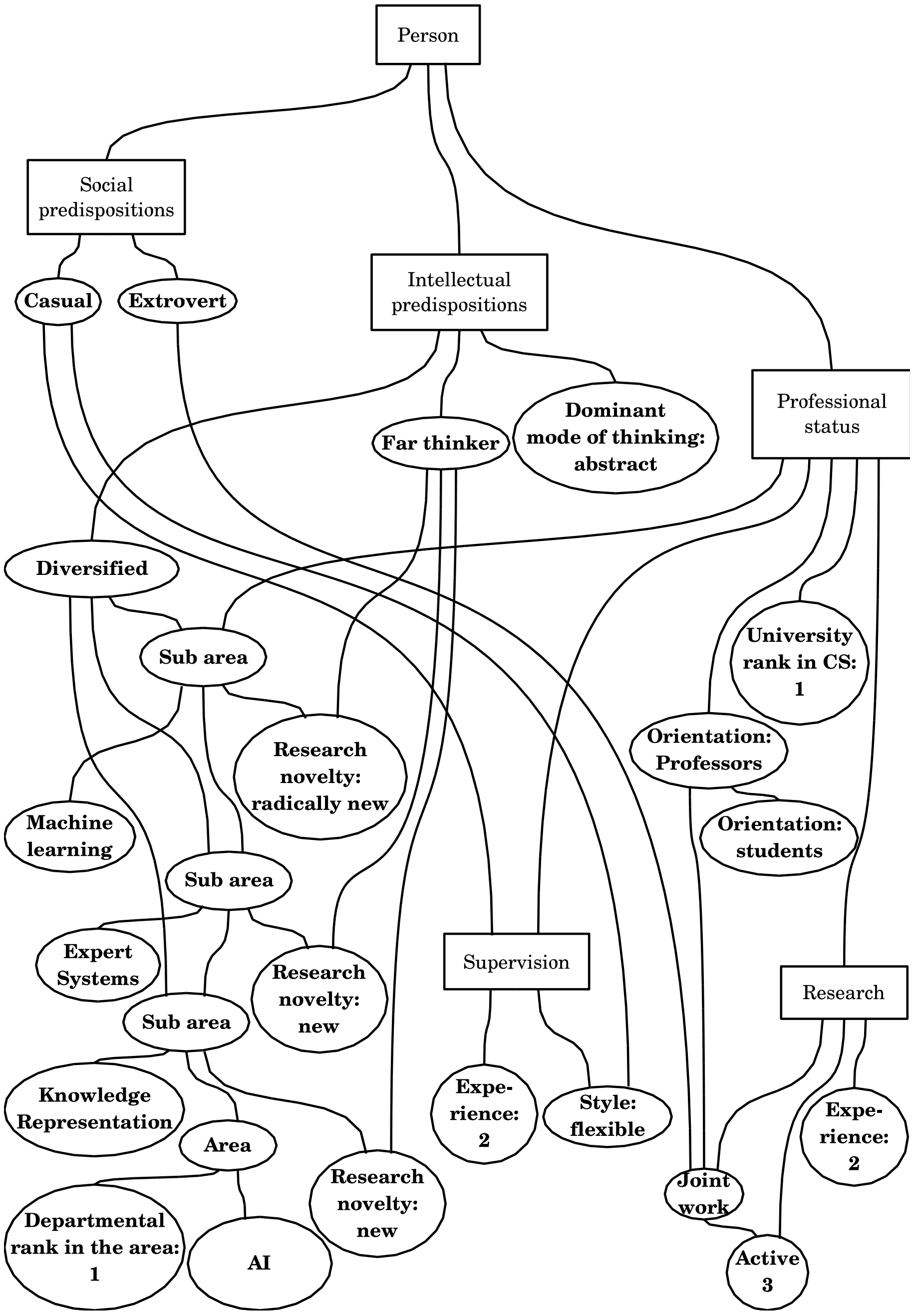

Figure 5.7: Element $\bar{\boldsymbol{\gamma}}_1$ of sub-class Γ_1.

constructed as follows:

$$\begin{aligned}\bar{\boldsymbol{\gamma}}_1 \quad = \quad & \bar{\boldsymbol{\kappa}}_1 \lhd \tau_3 \lhd \tau_6 \lhd \tau_{10-2} \lhd \tau_{12} \lhd \tau_{11} \lhd \tau_{15} \lhd \tau_{16-2} \lhd \tau_{20} \lhd \tau_{22} \lhd \tau_{26-\text{Machine learning}} \\ & \lhd \tau_{33} \lhd \tau_{22} \lhd \tau_{26-\text{Expert systems}} \lhd \tau_{32} \lhd \tau_{22} \lhd \tau_{26-\text{Knowledge representation}} \lhd \tau_{32} \\ & \lhd \tau_{25} \lhd \tau_{23-1} \lhd \tau_{34-1}\end{aligned}$$

Its typicality $\nu_1(\bar{\gamma}_1)$ in the subclass Γ_1 is computed with the method used previously and the weights found in Table 5.2. Its value is as follows:

$$\nu_1(\bar{\gamma}_1) = 190.0$$

5.4.2 Second example of a sub-class

Another sub-class of the class Γ is the sub-class Γ_2 representing people who are very productive. A person belonging to this sub-class will be a pragmatist and a concrete thinker. Her or his orientation will tend to be towards industry. She or he will do joint work with other professors to obtain more results. She or he will have numerous graduate students for the same reason.

The progenitor $\bar{\boldsymbol{\kappa}}_2$ of this sub-class is presented in Figure 5.8. $\bar{\boldsymbol{\kappa}}_2$ is the same progenitor as $\bar{\boldsymbol{\kappa}}$ with the transformations τ_5 and τ_{13} applied.

The set of transformations that define the sub-class Γ_2 is as follows:

$$\begin{aligned}\Theta_2 &= \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_7, \tau_8, \tau_9\tau_{10}, \tau_{11}, \tau_{14}, \tau_{15}, \tau_{16}, \tau_{17}, \tau_{18}, \tau_{21}, \tau_{22}, \tau_{23},\\ &\quad \tau_{24}, \tau_{26}, \tau_{28}, \tau_{29}, \tau_{30}, \tau_{31}, \tau_{34}\}\end{aligned}$$

The weights associated with the transformations of this sub-class are summarized in Table 5.3.

The example shown in Figure 5.9 represents the element $\bar{\gamma}_2$ of the sub-class Γ_2

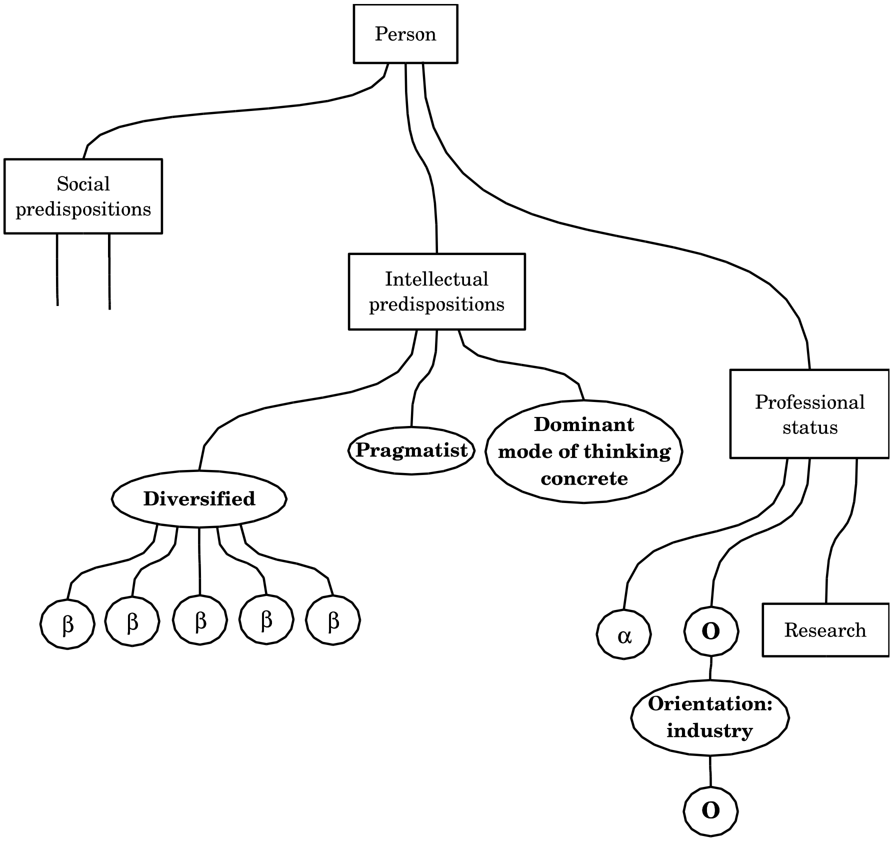

Figure 5.8: Progenitor $\bar{\boldsymbol{\kappa}}_2$ of sub-class Γ_2.

constructed as follows:

$$\begin{aligned}
\bar{\boldsymbol{\gamma}}_2 \quad = \quad & \bar{\boldsymbol{\kappa}}_2 \triangleleft \tau_1 \triangleleft \tau_4 \triangleleft \tau_8 \triangleleft \tau_{10-3} \triangleleft \tau_{11} \triangleleft \tau_{14} \triangleleft \tau_{16-3} \triangleleft \tau_{18-3} \triangleleft \tau_{24} \triangleleft \tau_{22} \\
& \triangleleft \tau_{26-\text{Software reliability}} \triangleleft \tau_{31} \triangleleft \tau_{22} \triangleleft \tau_{26-\text{Software evolution}} \triangleleft \tau_{31} \triangleleft \tau_{22} \\
& \triangleleft \tau_{26-\text{Parallel programming}} \triangleleft \tau_{31} \triangleleft \tau_{30} \triangleleft \tau_{23-2} \triangleleft \tau_{34-2}
\end{aligned}$$

Its typicality $\nu_2(\bar{\boldsymbol{\gamma}}_2)$ in the class Γ_2 is as follows:

$$\nu_2(\bar{\boldsymbol{\gamma}}_2) = 255.0$$

Table 5.4 summarizes the typicality of the elements $\bar{\boldsymbol{\gamma}}$, $\bar{\boldsymbol{\gamma}}_1$ and $\bar{\boldsymbol{\gamma}}_2$ computed in the class Γ and in the sub-classes Γ_1 and Γ_2.

Transformation	Typicality
τ_1	0.0
τ_2	0.0
τ_3	0.0
τ_4	0.0
τ_7	100.0
τ_8	0.0
τ_9	0.0
τ_{10-1}	0.0
τ_{10-2}	10.0
τ_{10-3}	20.0
τ_{11}	10.0
τ_{14}	35.0
τ_{15}	5.0
τ_{16-1}	0.0
τ_{16-2}	10.0
τ_{16-3}	20.0
τ_{17}	40.0
τ_{18-2}	10.0
τ_{18-3}	30.0
τ_{21}	0.0
τ_{22}	0.0
τ_{23}	0.0
τ_{24}	50.0
τ_{26}	0.0
τ_{28}	0.0
τ_{29}	0.0
τ_{30}	0.0
τ_{31}	30.0
τ_{34}	0.0

Table 5.3: Typicalities of the transformations for the class Γ_2.

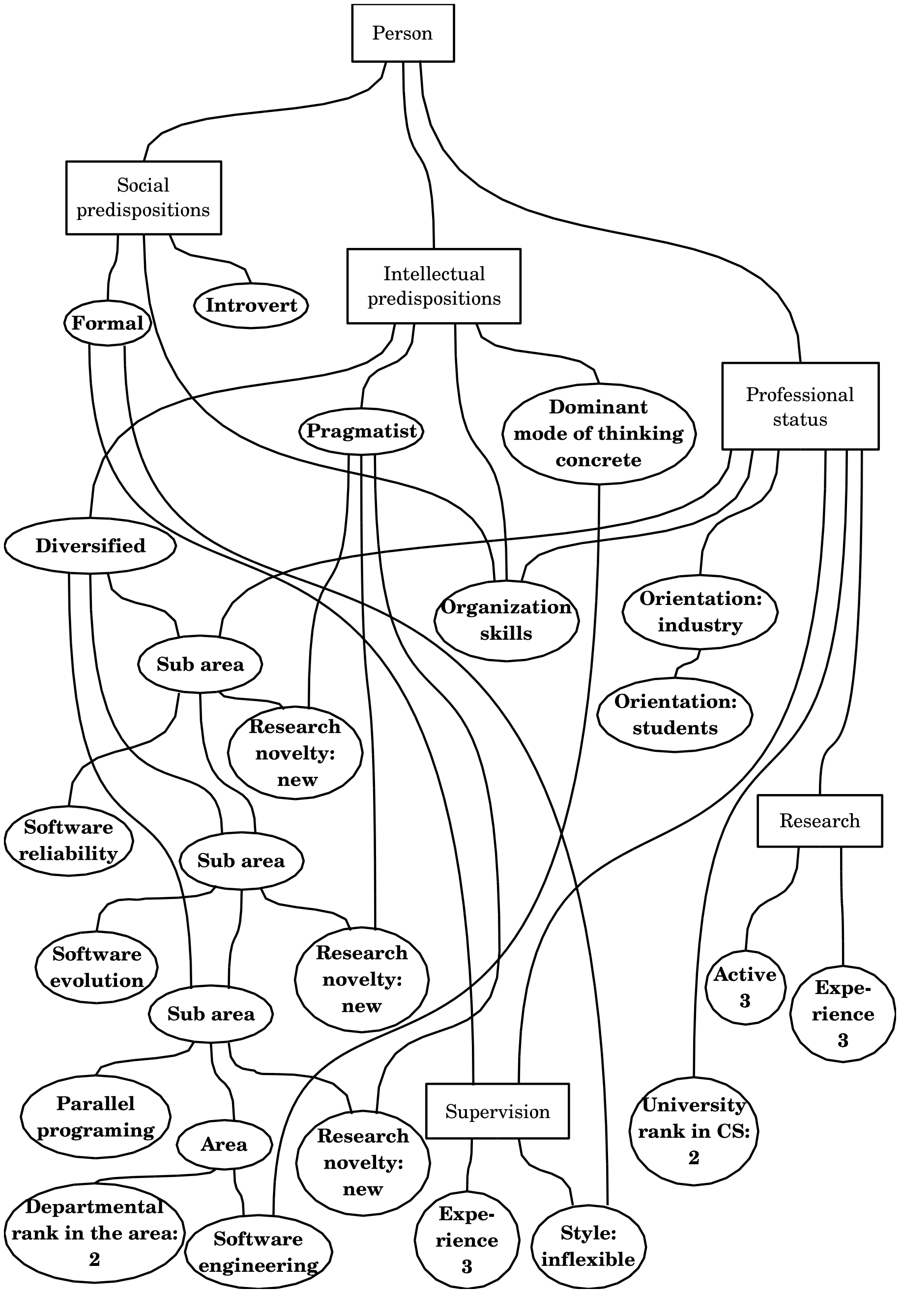

Figure 5.9: Element $\bar{\gamma}_2$ of sub-class Γ_2.

Element	Typicality	Typicality	Typicality
	in class Γ	in sub-class Γ_1	in sub-class Γ_2
$\bar{\gamma}$	120.0	150.0	X
$\bar{\gamma}_1$	167.0	190.0	X
$\bar{\gamma}_2$	230.0	X	255.0

Table 5.4: Typicalities of the different elements in the class and subclasses (X means that the element cannot be generated from the progenitor and the transformations defining the class).

Chapter 6
Homepage retrieval

The user—a graduate student—issues a query: either a full representation, a partial representation, or the description of a class (progenitor and set of transformations). The retrieval system returns the elements of a class or of a set of classes. This chapter explains the matching process and highlights the differences between the model developed in the present study and the Boolean model.

6.1 Matching

Matching is the core of the retrieval process. In the model presented in this thesis, retrieval is based on classes. Classification of the representations of the academic homepages is done before processing the query. Each class contains several elements, and each element can belong to several classes.

When a user issues a query, this query can be expressed in three different forms:

- An element of the universe. The user inputs as a query the full representation of the professional profile of the fictitious professor with whom she or he would like to work.

- A partial representation. The query is a partial representation, *i.e.* a part of a representation described in the previous case.

- The description of a class. The query issued is composed of a progenitor and a set of transformations.

6.1.1 The representation of an element as a query

In this type of retrieval, the user inputs the full representation of the professional profile of a fictitious professor. Two cases arise:

- If the input representation belongs to only one class, the retrieved elements are the elements that belong to this class.
- If the input representation belongs to more than one class, the retrieved elements are the elements of the classes for which the input representation has the highest typicality. The number of classes output depends on the precision and completeness needed by the user.

The existence of a hierarchy in the classification in classes, sub-classes, and so on, increases the precision of the retrieval. Depending on the degree of precision required by the user, the depth of the search can be adapted. Ranking is based on the typicality of each element for the class.

6.1.2 An incomplete representation as a query

In most cases, the user cannot input a full representation, because she or he is unable to specify her or his query exactly or does not want to restrict her or his query too much. Hence, the information retrieval system should be able to process such an incomplete query.

This input representation is considered as an element for which the generative process has been stopped; it is an incomplete element. If the generating process has been achieved, the typicality of the full representation obtained would have been the sum of the weights of all the transformations present in its construction history. However, the transformations applied to the incomplete element are unknown. Hence, a null weight is assigned to each of them. The typicality is therefore the sum of the weights of the transformations of the constructive history of the incomplete element. This typicality is an *assumed* typicality.

In the previous chapter, the typicality of an element was defined as the sum of the weights of the transformations present in its constructive history. This measure is effective because it assigns higher values to incomplete elements that contain in their constructive histories the set of transformations that most strongly determine class membership.

The set retrieved is the class in which the partial representation has the highest typicality. As before, the ranking is based on the typicality of the elements within the class. For example, the user can input a query represented in Figure 6.1. This query is constructed as follows:

$$\bar{\boldsymbol{\kappa}} \lhd \tau_2 \lhd \tau_3 \lhd \tau_6 \lhd \tau_{10-2} \lhd \tau_{12} \lhd \tau_9 \lhd \tau_{15} \lhd \tau_{16-2} \lhd \tau_{20}.$$

The value of the typicality associated with this query in class Γ is 87. This value is relatively large. Depending on the other classes, the system may retrieve some elements of class Γ.

However, as in any retrieval system, when the query of the user is not completely

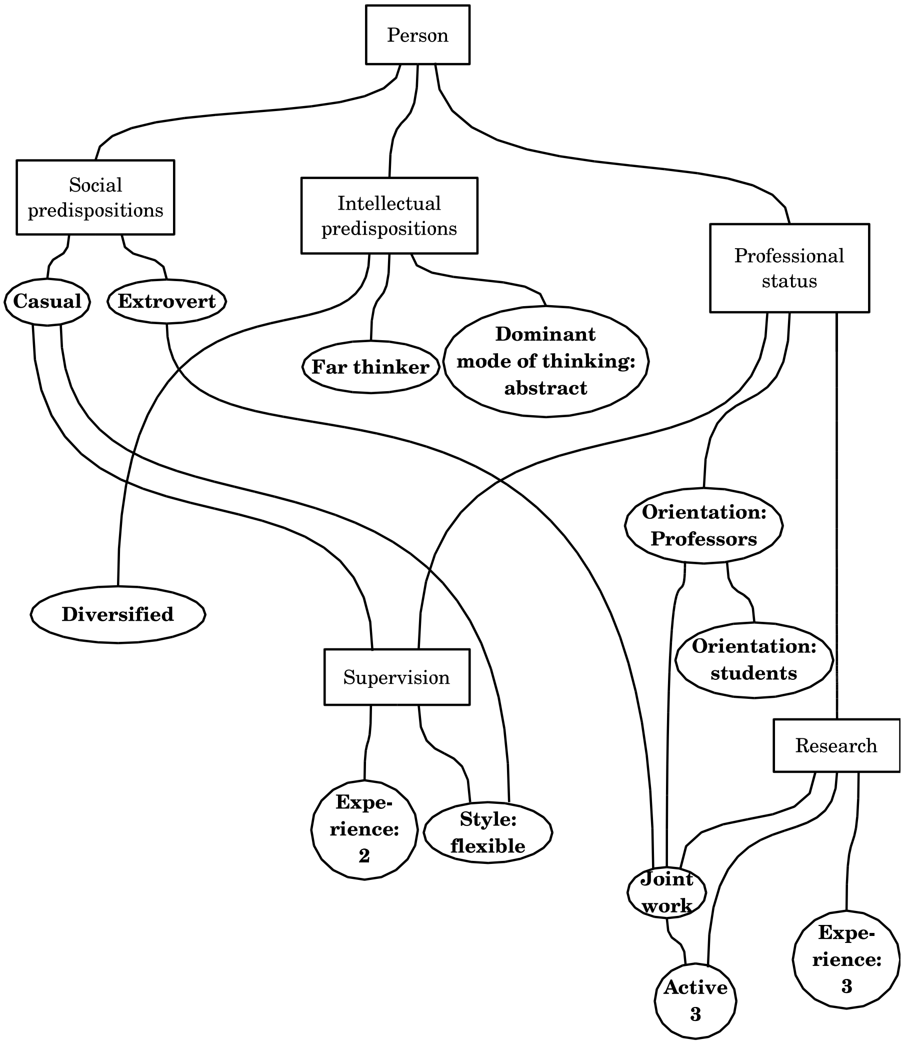

Figure 6.1: Example of a query.

defined, the retrieved class may not correspond to the (unexpressed) expectations of the user.

The ranking method used when the user inputs either a full or a partial representation may be slightly improved. The typicality of the retrieved elements can be computed using larger weights for the transformations that constitute the constructive history of the query. The weight increase can be defined as follows:

- A large multiplicative coefficient can be assigned to the transformations that attach attributes like **Area**. Such attributes are easy to define and cannot mislead the user.

- A small multiplicative coefficient can be assigned to the transformations that attach attributes that are more difficult for the user to define.

This method assigns a higher rank to the elements showing a higher similarity to the input representation whereas the first method assigns a higher rank to the more typical elements of the class.

6.1.3 The description of a class as a query

In some cases, the user inputs her or his query as a fully defined class, *i.e.* a progenitor and a set of weighted transformations. The system tests the membership of each element of the universe to the input class. If the element belongs to the class, it is part of the retrieved set. Ranking is based on typicality.

6.2 Comparison with a Boolean model

In a Boolean model, each document of the corpus is represented by a set of keywords. A query is expressed as a set of keywords linked by logical operators: AND, OR, and NOT. A Boolean model is applied to the corpus defined for this study to draw a comparison between the Boolean model and the one presented in this study.

To keep some consistency between both models, in the Boolean case, the keywords associated with each homepage belong to the set of attributes presented in Chapter 4. The query issued by the user is also a subset of these attributes, linked by the logical operators. The system retrieves the elements that satisfy the Boolean expression of the query. No ranking is done.

There are three main differences between these models:

- In the Boolean model, no ranking is possible. Hence, the first element output by the retrieval system can be the less interesting for the user. However, in the model based on ETS, if the user issues a query which conveys all the information she or he wants, she or he will get the best result first.

- Given a query, expressed as a full representation or a partial representation in the model based on ETS or as a list of keywords linked by logical operators in the Boolean model, the set of elements retrieved will be significantly different.

  In the Boolean model, the elements retrieved by the system will correspond to those that satisfy the Boolean expression of the query. The main disadvantage is that this exact matching may lead to retrieval of too few or too many documents.

  In the ETS model, the elements retrieved will be those that are similar to or contain the input representation, further all the elements that have similar properties. This property improves the completeness and the precision of the information retrieved.

- If no document matches the query of the user, the set of documents retrieved using the Boolean model is empty whereas the set of documents retrieved using the model based on the ETS framework is the class in which the partial or full representation given by the user has the highest typicality. This typicality can even be negative. The documents retrieved are the most similar to the query.

6.3 Discussion

The Boolean model is one of the simplest retrieval models. More elaborate models have been developed; some of them include a method of ranking (Chapter 2).

The novelty of the model based on the ETS framework compared to other models used in information retrieval is the use of the structure of the information contained in the document. This structure expresses the interdependence of attributes. This dependence may be expressed in other formalisms, but the structural information will be treated as the "other" information. For example, the attachment can be expressed in a Boolean model by adding to the list of keywords a list of structural information, but this information can be used only as the keywords are used. This information cannot be treated as structure. The use of the structural information in the ETS model is one of its main advantages. The structural representation of a document expresses its constructive history.

The novelty of the model based on the ETS formalism, therefore, lies also in the presence of a generating process that allows classification of documents into classes of elements having similar properties. Such a model returns elements that can be very different in their representations but these elements have similarities in their professional profile.

For example, if a graduate student wishes to work with a professor who is likely to be successful, she or he may input the element $\bar{\gamma}$ presented in Figure 5.5. The class retrieved is the class Γ. The element $\bar{\gamma_2}$ presented in Figure 5.9 belongs to the class Γ, it is therefore retrieved. The representation of the element $\bar{\gamma_2}$ is totally different from the representation of the element $\bar{\gamma}$. However, the element $\bar{\gamma_2}$ is one of the documents that the user expects. No other model used in information retrieval would have included this element in its output because of its large difference with the query.

Chapter 7
Conclusion and recommendations

The main objectives of this thesis were to develop a model based on the ETS formalism, to apply it for an information retrieval task, and to show its advantages compared to other models.

First, the corpus consisting of the academic web pages of professors in Computer Science in various Canadian universities was studied to find regularities. Based on these regularities and those features of the professional profile of the professor believed to interest a graduate student, a structural ETS representation of the professional profile was developed. This representation was composed of six placeholders and a list of attributes that characterize three main aspects of the professional profile of the professor. Based on this model, classes grouping the professors who show similarities in their personalities can be defined. One representative class and two sub-classes were constructed. The retrieval process was then defined. When a user issues a query (which is a representation), the system returns all the elements of the class in which the input representation has the highest typicality. The ranking is based on the value of the typicality.

Although the model was simple in that it included only a few aspects of the professional profile, it has proven to be effective for an information retrieval task. In addition, the model showed advantages in comparison to models commonly used in information retrieval: it includes the structural information in the representations. The repartitioning of the elements into classes and sub-classes enables the system to retrieve relevant elements that are totally different from the query. This retrieval would not be possible with any classical information retrieval models.

Based on the experience gained from the present study, the following recommendations for further research are made:

- The representation of the traits of personality of the professor was built without significant knowledge of Psychology. It was based on the study of the characteristics of the corpus and an analysis of the information interesting for a student. This representation is limited, as not all the aspects of the personality are taken into account. Therefore, an advanced psychological investigation of the personality of professors and of the needs of the graduate students would improve the reliability of this representation.

- The construction of the ETS representation of the homepages has been done manually. A system should be developed to index the webpages automatically. This issue is not related directly to the proposed ETS representation, but some investigations need to be done to achieve it.

- The class and the sub-classes presented in this thesis have not been selected automatically. To automate this creation of classes, the study of the learning process of the ETS model would be recommended.

- The presented model has been used on few web pages (compared with the thousands existing on the WWW). Large-scale tests of a prototype model will be useful to gain a better knowledge of the effectiveness of this model.

- The major drawback of this model is that it is difficult for a user to use. To perform a search, the user has to define a partial or full representation in a form he is not used to (structural). To improve the ease of utilization of this model, the user should input a query in a more classical form. A pre-processing stage would transform this query into an ETS representation.

- A user should be asked to test this model and compare it with Google (one of the most widely used search engines). Even if the two methods are different, the user should be able to express which one gives the results she or he feels to be the best.

- The model has been applied to one domain, i.e. academic webpages. It would be interesting to transfer it to similar domains.

Bibliography

[1] R. Baeza-Yates and G. Navarro. Integrating contents and structure in text retrieval. *ACM SIGMOD Record*, pages 67–79, 1996.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[3] R. K. Belew. *Finding out about: a cognitive perspective on search engine technology and the WWW.* Cambridge University Press, 2000.

[4] T. Berners-Lee, R. Cailliau, A. Luotonen, H. F. Nielsen, and A. Secret. The world wild web. *Communication of the ACM*, pages 76–82, 1994.

[5] H. Bunke and A. Sanfeliu. *Syntactic and Structural Pattern Recognition - theory and applications.* World Scientific Publishing Co. Pte. Ltd., 1990.

[6] F. Burkowski. An algebra for hierarchically organized text-dominated databases. *Information Processing and Management*, pages 333–348, 1992.

[7] F. Burkowski. Retrieval activities in a database consisting of heterogeneous collections of structured text. *Proc. of the 15th Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 112–125, 1992.

[8] K. S. Fu. *Syntactic Pattern Recognition and Applications.* Prentice-Hall, 1982.

[9] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. Information retrieval using a singular value decomposition model for latent semantic structure. *Proc. of the 11th Annual international ACM SIGIR Conference on research and development in Information Retrieval*, pages 465–480, 1988.

[10] L. Goldfarb and O. Golubitsky. What is a structural measurement process? Technical Report TR00-147, Faculty of Computer Science, University of New Brunswick, Canada, October 2001.

[11] L. Goldfarb, O. Golubitsky, and D. Korkin. What is a structural representation? Technical Report TR00-137, Faculty of Computer Science, University of New Brunswick, Canada, December 2000.

[12] L. Goldfarb, O. Golubitsky, and D. Korkin. What is a structural representation in chemistry : Towards a unified framework for cadd? Technical Report TR00-138, Faculty of Computer Science, University of New Brunswick, Canada, December 2000.

[13] V. Gudivada, V. Raghavan, W. Grosky, and R. Kasanagottu. Information retrieval on the world wild web. *IEEE Internet Computing*, pages 58–68, 1997.

[14] D. Korkin and L. Goldfarb. Multiple genome rearrangement: A general approach via the evolutionary genome graph. *ISBM*, 2002.

[15] L. Miclet. *Structural methods in pattern recognition.* Springer-Verlag, 1986.

[16] C. N. Mooers. Information retrieval viewed as temporal signalling. *Proceedings of the International Conference of Mathematicians, Cambridge, Massachusets August 30-September 6, 1950*, pages 572–573, 1952.

[17] G. Navarro and R. Baeza-Yates. A language for queries on structure and contents of textual databases. *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 93–101, 1995.

[18] G. Navarro and R. Baeza-Yates. Proximal nodes: A model to query document databases by content and structure. *ACM Transactions on Office and Information Systems*, pages 401–435, 1997.

[19] S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, pages 129–146, 1976.

[20] G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, pages 202–215, 1983.

[21] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[22] R. Wilkinson and P. Hingston. Using the cosine measure in a neural network for document retrieval. *Proc. of the ACM SIGIR Conference on research and development in Information Retrieval*, pages 202–210, 1991.

[23] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized vector space model in information retrieval. *Proc. 8th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, 1985.