Reasoning with Legal Cases seen as Theory Construction

Thesis submitted in accordance with
the requirements of the University of Liverpool
for the degree of Doctor of Philosophy by

Alison Chorley

September 2006

Abstract

The aim of this thesis is to investigate thoroughly the view that reasoning with legal cases can be seen as a process of theory construction. Bench-Capon and Sartor [10] describe a set of theory constructors that can be used to construct a theory to explain a body of case law and these theory constructors were implemented in Java to produce CATE (Case Theory Editor) which can be used by a human user to construct theories.

CATE was used to reproduce a series of examples described in [10] to show that the theory constructors can be implemented and used in a practical system to produce usable theories to explain bodies of case law.

We then investigated how the construction of theories could be automated. We drew on methods from Case-Based Reasoning and implemented AGATHA (ArGument Agent for THeory Automation). AGATHA models a dialogue game where two agents represent the plaintiff and defendant lawyers. A series of dialogue moves based on these Case Based Reasoning approaches represent how lawyers might argue in a court to decide a case and as a side effect the theory is constructed and refined. The program shows that the method is able to construct a set of plausible theories for a given body of case law. The computational problem with this method is that AGATHA creates the entire search space for the problem and as the size of the case background that the lawyer agent can use increases, the search space rapidly becomes very large, posing computational problems and difficulties in interpreting the output.

Ethel was then implemented to analyse the theories according to several criteria and AGATHA can use this evaluation with two search heuristics to guide the lawyer agents through the search space.

The first heuristic implemented in AGATHA is a co-operative heuristic based on A* search, where the two agents are co-operating to produce the "best" theory possible and it does not matter who the winner is. Legal argumentation is, however, an

adversarial process where the two lawyers each want to win but they also want to prevent the other lawyer from winning. To model this behaviour a second heuristic was implemented based on the $\alpha\beta$ pruning heuristic.

In AGATHA both the lawyer agents have access to the same cases to build their theories. To explore what would happen if each lawyer agent had access to a different set of cases the final program, ROSALIND (AGATHA's Daughter), was implemented. ROSALIND uses the adversarial heuristic but as the two agents do not know what cases the other agent is using, each agent must base their next move on what the other agent would do if it had their cases.

Both heuristics were shown capable of pruning the search space, to make the approach computationally possible, while providing theories of a quality comparable to the best of the other existing approaches.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Three ways of reasoning

The importance of cases in legal reasoning has been recognised throughout the development of AI and Law. Even approaches which took formalisation of legislation as their starting point, e.g. [37], rapidly came to realise that crucial questions of the interpretation and application of terms found in the legislation could be answered only by reference to cases (e.g. [4]). Cases, despite some differences in the ways in which they are used, are of considerable importance in Civil Law jurisdictions as well as Common Law jurisdictions [25]. Given this centrality of cases, a good understanding of their contribution and use is vital.

Despite the recognition of the importance of cases, there has been less agreement on the way in which cases should be represented and used within AI and Law systems. We may distinguish approaches which have used cases as a knowledge source, (e.g. [5]) on a par with other sources such as statutes and commentaries, and those which have placed importance on the structure and manipulation of cases as entities in their own right, as in, for example, the various systems originating in HYPO [2], [1], [38], [15]. In the first approach cases will be represented only implicitly, whereas in the second they must be represented explicitly. Both these approaches capture aspects of the truth. Given a body of case law, lawyers experienced in the field will be able to give rule like advice: for example we may, with confidence, on the basis of case law, say that injury during a standard commute to work will not be considered as arising out of, or in the course of, employment and so not attract Industrial Injury compensation. On the other hand, when it comes to forming an argument in the context of a particular legal case, precedents will be explicitly deployed, in the manner of the HYPO like systems. Both approaches have their strong points: on the first approach we can examine the knowledge that the system will apply: such verifiability may be essential, for example, if we are to trust the operation of a system in administrative law. Moreover we can examine the knowledge to critique the law, identifying areas where we are dissatisfied with it and perhaps propose amendments to legislation accordingly. On the other hand, such systems involve potentially subjective interpretation to extract rules from the case, and do not provide very satisfactory models of legal reasoning. Also they fix the theory, whereas in practice, the interpretation of cases is, at least potentially, continually open to reconsideration (e.g. [23]). The second approach means that each new situation is thought through afresh on its particular merits, rather than being decided mechanically.

A middle way, which attempts to include both aspects, is to introduce the notion of theory construction. On this view, there is always a body of knowledge representing the current legal thinking, in the form of a theory, but these theories are always, at least in principle, constructed afresh when a new case appears: thus the theory will be subject to modification in the light of the context provided by difficult cases. We therefore attain the benefits of both approaches: the theory provides the knowledge for inspection (and criticism and modification), and the process of construction can reflect the practice of legal argument.

I have taken the approach of Bench-Capon and Sartor described in [10] and implemented it. Using this implementation I have conducted a series of experiments to explore aspects of the theory and its potential for automation and these experiments evaluating the approach and various design choices are the contribution of my work rather than the implementation itself. The implementation is useful because it enables the experiments to be carried out.

1.2 Contribution

The aim of this research is to thoroughly investigate the view that reasoning with legal cases can be seen as theory construction, and the approach of theory construction using the Theory Constructors defined by Bench-Capon and Sartor in [10]. The first step in this research was to implement the Theory Constructors given in [10] and to test them by hand to see if they can be used to explain the cases in the domain. To do this, CATE (CAse Theory Editor) was implemented in Java to provide a graphical user interface to enable a user to construct theories using this approach. Bench-Capon and Sartor left several design choices open in their theoretical account in [10] and these will be investigated using CATE.

The first stage in testing the approach of [10] was to ensure that it could be used to produce theories that perform as expected. In [10] the authors produced a series of theories as an example of how they expected the Theory Constructors to perform. These theories were reproduced using CATE and executed to give a decision for the cases included in the theories. The second stage was to answer a set of research questions including: is there a methodology we can use in constructing theories, can values be used to determine the relative importance of factors, and how do we compare sets of factors. To answer these questions, several experiments were performed using CATE. From these investigations we can see that the approach of [10] can be realised to produce usable theories to explain bodies of case law.

The next step in the research was to implement the extensions to the Theory Constructors to deal with dimensions instead of factors to investigate the significance of these two ways of representing cases. A dimension is a range of points and a point representing a certain fact in a case may appear anywhere on the dimension and may be moved along the dimension to suggest how the case could be strengthened or weakened. A factor represents a specific fact from the case and is either present in the case or not and cannot be used to suggest to strengthen or weaken a case. The extended Theory Constructors were implemented in CATE and similar research questions were asked.

The next phase in this research was to automate the process of theory construction. I drew on methods from Case-Based Reasoning including HYPO [2] and CATO [1] and implemented AGATHA (ArGument Agent for THeory Automation). AGATHA models a dialogue game where two agents represent the plaintiff and defendant lawyers. A series of dialogue moves represent how lawyers might argue in a court to decide a case and as a side effect the theory is constructed and refined. The result is a theory and a dialogue which explains how it was constructed. The program shows that the method is able to construct a set of plausible theories for a given body of case law. The computational problem with this method is that AGATHA creates the entire Theory space for the problem and as the size of the case background that the lawyer agent can use increases, the Theory space rapidly becomes very large, posing computational problems, and difficulties in interpreting the output.

This leads to the third program that was implemented, ETHEL (Evaluation of THEories in Law). ETHEL analyses the constructed theories and evaluated how "good" each theory is according to several criteria. AGATHA can now use this evaluation with two search heuristics to guide the lawyer agents into making "good" moves and create the "best" theories possible.

The first heuristic implemented in AGATHA is a co-operative heuristic based on A* search, where the two agents are co-operating to produce the "best" theory possible and it does not matter who the winner is. Legal argumentation is, however, an adversarial process where the two lawyers each want to win but they also want to prevent the other lawyer from winning. To model this behaviour a second heuristic was implemented based on the $\alpha\beta$ pruning heuristic.

In AGATHA both the lawyer agents have access to the same cases to build their theories. To explore what would happen if each lawyer agent had access to a different set of cases the final program, ROSALIND (AGATHA's Daughter), was implemented. ROSALIND uses the adversarial heuristic but as the two agents do not know what cases the other agent is using, each agent must base their next move on what the other agent would do if it had their cases.

Both heuristics were shown capable of pruning the search space, to make the approach computationally possible, while providing theories of a quality comparable to the best of the other existing approaches.

The dialogues produced by AGATHA and ROSALIND may not seem like natural arguments made in a legal court but they do produce "good" theories. Moreover, while the dialogues may not seem natural to lawyers, they remain explicable in terms of a legal conceptualisation.

1.3 Overview of Thesis

In Chapter 2 I describe the background material that has preceded and influenced this work and in the following chapter I give a description of the two law domains that are used throughout this work in the examples. In Chapter 4 I describe the implementation of CATE, a program that takes the Theory Constructors described in [10] and shows how theories can be created and evaluated. I describe the extension to CATE to handle the dimensions from [2] in the manner described in [10] in Chapter 5. I also show how factors could be placed on dimensions.

I then describe the implementation of a set of programs to automate the construction of theories. In Chapter 6 I describe AGATHA, a program that models a dialogue between two agents that creates and modifies a series of theories as each agent makes a move in the dialogue. ETHEL, a program to assess the quality of the theories produced by AGATHA is described in Chapter 7. In Chapters 8 and 9 I describe two search heuristics which are used to limit the number of theories produced by AGATHA and to provide a way of moving through the search space intelligently. Because AGATHA allows the two agents to use the same background information available to them another program, ROSALIND, was implemented to explore situations when the agents have access to different background information and is described in Section 9.5. Chapter 10 briefly describes and compares the dialogues produced by AGATHA and then gives some discussion and some concluding remarks.

Chapter 2

Background

Many different methods have been employed to try and describe the domain of law. It was thought that the Legal domain was logical and so rules could be produced to describe and explain legislation and expert knowledge. The problem is there are many exceptions to the rules and these also have to be modelled. For example, in [27] the author describes the rule that "vehicles are prohibited in the park" but what is a vehicle? Is a bicycle a vehicle and so prohibited? If the rule is to protect people enjoying the park from cars, is the ambulance rushing to someone's aid also prohibited? If rules are to be used, then all the exceptions to the rules must also be listed. There may also be conflicts between rules that have to be resolved.

A different method uses knowledge from cases that have been decided in a court of law, or precedent cases. Early systems simply retrieved precedent cases on how well they matched a new case, but later systems tried to reason on why a new case should be decided in a certain way. These systems are also limited in the fact that if there is a conflict as to how a case should be decided they cannot resolve the conflict and it is left to a human to decide.

A third method uses reasoning with cases and rules derived from legislation or expert knowledge. In Theory-based reasoning, precedent cases are used to explain a new case but conflicts are resolved by using rules from expert knowledge.

Rule-based approaches tend to be most popular in Europe, whereas case-based approaches mostly originate in the US.

2.1 Rule based Methods

Rule-based approaches can use rules to express "expert knowledge", for example Smith in [39], or to represent legislation, for example in the British Nationality Act program described in [37]. Creating rules from legislation means that the rules stay linked to the legislation which means that when the legislation changes (which it will because the legal domain is always changing, even if only slightly) it is fairly simple to change the corresponding rules. The problem with creating systems which just use rules derived from legislation is that more expertise of the legal area may be needed to understand the system and the systems may be complicated to maintain. When creating rules from expert knowledge, the rules describe the knowledge of the expert and may be very understandable compared to the rules from legislation, however the rules obtained have no connection with the legislation and if anything changes it is very difficult to find the correct rules to change and may require an expert to completely rewrite the rules. A third way is to combine both approaches and obtain rules from legislation, rules from expertise and rules as to how they relate to each other. This way means that it is easy to add, change or delete rules and still maintain the relationship between the legislation and the expert knowledge.

Legislation can be interpreted by different people in different ways depending on how the legislation is to be used. In [4], Bench-Capon describes how legislation can be formalised as rules in a Knowledge Based System but the Knowledge Based System will also need to be supplemented with rules derived from expert knowledge. In [5], Bench-Capon showed that there are several ways to create expert systems that can be used in the legal domain. The systems could use rules derived from legislation, rules derived from expert knowledge or a combination of both.

In the above description of an expert system the rules are conflict free, but in practice there will usually be conflict between the rules even when formalising legislation. This is because legal reasoning often operates on defeasible and inconsistent information. Some of these conflicts can be resolved using the general principles of *specificity* (the most specific rule is preferred), *superiority* (the rules produced from the laws of a superior hierarchy are preferred over the rules produced from the laws of an inferior hierarchy) and *temporally* (the rules produced from the most recent laws are preferred over the rules produced from the older laws).

In [30], Prakken and Sartor analysed legal reasoning with precedent cases (from case-based reasoning) in the setting of a formally defined dialogue game. They showed how factors can be linked to rules and rule preferences, and compared their dialogue game to various case-based reasoning systems, including HYPO [2], CATO [1], CABARET [38] and Branting's work [13]. In their system, cases are represented as argument structures consisting of two conflicting rules and a priority statement stating which rule is preferred. They define case-based reasoning moves from HYPO as strategies for introducing information into a dispute. They represent the precedent cases as multi-step so they are able to represent the idea of citing portions of precedents from Branting's work. A precedent case may only match a portion of the new case and a different precedent case may match a different portion. The Factor Hierarchy from CATO is represented by a set of rules with rule priorities and using these rule priorities they could use the moves of emphasising and downplaying distinctions from CATO. Instead of restricting their system to using the most-on-point cases (cases with the highest number of matches with the new case and decided for the wanted side) for solving conflicting precedents they can use many other priorities including favouring the superior, most recent and most specific rules.

2.2 Case Based Approaches

Initially Case-based Reasoning systems were simple retrieval systems (for example Kowalski in [22]) that retrieved precedent cases based on the factors present in a new case. In the 1980's Ashley and Rissland worked on a system called HYPO [2] which created arguments using precedent cases to explain case outcomes. Ashley then worked with Aleven to produce CATO [1] for teaching law students how to argue effectively in law. Ashley then worked with Brüninghaus to produce IBP [15] which predicts the outcome of cases. In a different direction Rissland developed CABARET [38]with Skalak to combine rule based and case based approaches. they then went on to develop BankXX [33] with Friedman which uses heuristic search.

2.2.1 HYPO

HYPO [2] is a system for reasoning about precedent cases and uses adversarial Case Based Reasoning. HYPO functions in the domain of US Trade Secret Misappropriation Law and the main focus is creating arguments not making decisions. Justifying a conclusion about a problem involves drawing an analogy to a similar past case and arguing that the problem should be decided in the same way.

HYPO consists of a Case Knowledge Base, which is a structured database containing a small number of actual and hypothetical legal cases and a set of Dimensions, which are HYPO's principle index to cases in the case knowledge base. They represent the stereotypical facts of legal cases and their structure allows HYPO to determine whether a dimension applies to a case and also the magnitude of the dimension.
The Process of Reasoning/Arguing in HYPO comprises several tasks or moves:

1. Drawing factual analogies to past cases. This means stating the similarities between the case under consideration and the precedent cases.

2. Distinguishing precedent cases. HYPO states the dimensions that were present in the precedent cases but not in the case under consideration and vice versa and shows how these dimensions were important in deciding the precedent cases and as they are so important, their lack in the new case means that it cannot be decided in the same way as the precedent case.

3. Citing precedent cases as counter examples. HYPO states cases with a different outcome that have the same (an *as-on-point* case) or more similarities (a *more-on-point* case) with the case under consideration than a previously stated precedent case.

4. Posing made-up cases or hypotheticals. HYPO suggests how a case could be strengthened or weakened along a dimension to make it more or less similar to the case under consideration.

5. Evaluating the strength of case-citing arguments.

HYPO's reasoning process consists of:

1. *Analyse the current fact situation dimensionally.* HYPO finds all the applicable dimensions and near-miss dimensions by checking the prerequisites of the dimension and comparing them to the new case.

2. *Retrieve all the relevant precedent cases from the Case Knowledge Base.* Using the applicable and near-miss dimensions HYPO retrieves all of the precedent

cases with the same dimensions. The cases may be decided for either party and will have one or more dimensions in common with the new case.

3. *Position the new case with respect to the retrieved precedent cases.* HYPO then creates two claim lattices using the dimensions and the precedent cases which contain these dimensions. The regular claim lattice only contains the applicable dimensions whereas the extended claim lattice also contains the near-miss dimensions. Figure 2.1 shows an example claim lattice. Node 0 is the new case and nodes 1 and 2 contain the precedent cases which are most-on-point. The precedent cases in node 1 match different dimensions with the new case than the precedent cases in node 2. The precedent cases in node 1 are more-on-point than the precedent cases in nodes 3 and 4 because the cases in node 1 will match more dimensions with the new case. This means that the branches in the claim lattices represent different ways of arguing about the new case.

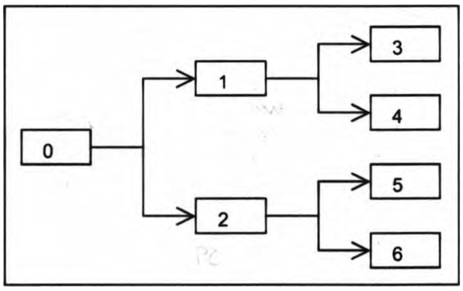

Figure 2.1: Example Claim Lattice.

4. *Compare cases and select the best precedent cases.* HYPO retrieves the most-on-point precedent cases for each party (both Plaintiff and Defendant) and then finds the best precedent cases for those parties. The best cases are those that have at least one dimension that favours the party that won the case. Choosing a more-on-point precedent case means that the opponent has less opportunity to counter the case.

5. *Generate 3-ply Arguments for the current fact situation citing precedents.* Side 1 analogises the best case to the new case, then side 2 can respond by distinguishing or countering and then side 1 responds to create the 3-ply argument. Either the plaintiff or the defendant can start the 3-ply argument.

6. *Hypothetically modify the current fact situation.* HYPO can create hypothetical variations of the case under consideration to show how the case can be strengthened or weakened. There are five heuristics that HYPO can use to modify the case:

   (a) Make a near-miss dimension apply.
   (b) Strengthen or weaken a case along an applicable dimension.
   (c) Move the case along a related dimension.
   (d) Make the case extreme along a dimension.
   (e) Make a case into a near-miss given a target.

7. *Generate 3-ply Arguments for selected hypotheticals.*

8. *Explain by illustrating arguments and comparing arguments for the new case and selected hypothetical cases.*

HYPO would be a good tool for lawyers to quickly find cases and obtain 3-ply arguments to see how their opponent may argue (in an ideal world). Although central to the original conception, moves 6 to 7 were less than fully developed in HYPO and have received little subsequent attention.

2.2.2 CATO

CATO [1] provides an instructional environment for students to work through a curriculum of case analysis, theory-testing and argumentation tasks.

CATO can generate examples to illustrate ways of using cases in arguments. In comparison a textbook can have only a limited number of examples. By dynamically generating the examples, the students can study as many as needed. CATO also makes the underlying structure of the arguments visible and this helps to guide the argument. CATO also makes students' tasks more manageable. This is because CATO reduces some of the distracting complexity because it uses cases pre-analysed in terms of factors. Compared to full text retrieval, CATO makes it easier to zero in on relevant cases. CATO aims to suppress the complexity long enough for the students to find patterns and structures in the arguments and can try the process of formulating, testing and revising a legal hypothesis, but not to the extent that it becomes pointless.

CATO differs from HYPO because HYPO used dimensions to represent the cases whereas CATO uses factors. A dimension is a range of points and a point for a certain case may be anywhere on the dimension range. A factor is a specific point and it is either present in the case or not.

For example, in HYPO there is a dimension called *Security-Measures-Adopted* which ranges through several possibilities from taking *minimal measures* to using *employee nondisclosure agreements*. CATO has two factors to represent this dimension, *F6-Security-Measures* and *F19-No-Security-Measures*. These two factors will, of course, not appear in the same case because you cannot both be taking security measures and not taking security measures. The dimension could represent more precisely how strict were the security measures you were taking.

Providing a list of factors to represent a case is a more conservative way of stating generalisations about a domain than stating necessary and sufficient conditions. The factors are used to represent factual strengths and weaknesses of cases and are a stereotypical collection of facts that influence the outcome of a case. The presence of a factor makes a case stronger or weaker for a side, but there is not an authoritative weighting scheme that could be used to decide whether the pro-plaintiff factors outweigh the pro-defendant factors, or vice versa, instead arguments are made comparing and contrasting the problem to past cases.

CATO's background consists of 147 Trade Secret Misappropriation Law cases, indexed by factors. For each case the database contains a list of factors and a squib. A squib is a short explanation of the case.

CATO arranges the factors in The Factor Hierarchy consisting of 26 base-level factors, 11 intermediate factors, 5 high-level factors (issues) and 50 links. The factors used to represent cases are linked to the intermediate legal concerns, which are in turn linked to legal issues. The upper two layers are referred to as high-level factors. The links indicate the level of support and can be strong (thick) or weak (thin) and this allows the blocking of weak paths from factors to high level factors by strong paths.

CATO uses the Factor Hierarchy:

- To identify issues in a problem case

- In the discussion of an issue, to focus on the strengths and weaknesses (factors) that are related.

- To give reasons why strengths matter to an issue being discussed
- To find strengths that are closely related to weaknesses and, hence, may compensate for those weaknesses

An issue is a point of contention. CATO raises an issue when a problem or case presents evidence related to it in the form of base-level factors, regardless of which side the factors favour. It starts with the applicable factors from the cases and traces paths upwards, collecting all legal issues and keeping track of which factor relates to which issue.

CATO uses the issues to organise its arguments. It can generate Issue-based Arguments for any problem represented in terms of factors, using any (small) set of cases selected by the student. CATO addresses each issue in turn, arguing for a favourable conclusion with respect to each, following a common strategy, to emphasise strengths first, then downplay weaknesses.

There are four basic argument moves to employ cases. When cases have one or more factors that are relevant to the issue, CATO uses them:

- To *Emphasise strengths related to the issue* - CATO states reasons why they matter (in terms of more abstract factors) and cites cases in which those strengths led to favourable outcome.
- To *Downplay weaknesses* - Points to strengths that are closely related in the Factor Hierarchy and therefore may compensate for the weakness. Also cites cases that had favourable outcome in spite of the fact that the same weaknesses were present.
- To *Distinguish* the opponents case
- As *counter examples* to cases cited by the opponent and may be either more-on-point or as-on-point counter examples.

CATO's issue based arguments serve as models. They are supported by cites to multiple cases and are organised by issue. CATO's example is meant to communicate rhetorical structure: How does one organise an argument by issues, addressing strengths and weaknesses for each issue.

CATO's domain model provides an account of how attorneys argue with cases. The CATO instructional environment is designed to help students learn basic skills of making arguments with cases through practice of two main tasks of theory testing and written argumentation.

First CATO presents dynamically generated argumentation examples. Second, it reifies argument structure in a number of ways including making the underlying structure of the arguments visible, as it is not visible in a more traditional setting. Finally, it makes students' tasks more manageable.

The analysis of CATO in [1] has been used as the basis of the experimental data presented in the rest of this thesis.

2.2.3 Developments of above

The following sections describe how the ideas in HYPO and CATO have been or can be extended.

2.2.3.1 CABERET

In [38], Skalak and Rissland describe the program CABARET. It is in the domain of Home Office Deduction. Taxpayers may legitimately deduct on a US Federal Income Tax Return expenses relating to an office maintained at the taxpayer's residence.

CABARET has a domain-independent architecture which heuristically combines rule-based and case-based reasoning. It uses a control strategy incorporating top-down and bottom-up processing to generate skeletal arguments.

There are two routes to the ideal case

- Specification-Driven (Top-down) - CABARET Specifies the ideal case, independent of the cases present in the case base. CATO uses top-down processing
- Case-Driven (Bottom-up) - Defines arguments according to the cases that actually exist in the case base. HYPO uses bottom-up processing

CABARET specifies the ideal case for a particular argument strategy-move combination. The cases then found determine which of the strategic moves can be used to implement the argument strategies given the limitations of the current case base.

CABARET generates argument skeletons that report the processing of the system to suggest possible attack. A complete argument could be based on such a skeleton. A correspondence can be seen between portions of the skeleton and the argument strategies, moves and primitives and the control heuristics that include them.

HYPO, CATO and CABARET identify but do not resolve conflicting arguments, the next system described, IBP, provides a means of adjudicating between conflicting arguments.

2.2.3.2 IBP - Issue Based Prediction

In [3], [15], [14] and [16] Brüninghaus and Ashley have taken CATO and adapted it for prediction. IBP is an algorithm that combines reasoning with an abstract domain model and case-based reasoning techniques to predict the outcome of case-based legal arguments.

IBP identifies the issues raised in a case, and then uses a kind of scientific evidential reasoning with cases to resolve conflicting evidence when the issue related factors favour both sides. It outputs a prediction and provides an explanation in an argument-like outline of its reasoning. IBP can predict which party will win and also whether it was a trade secret using its domain model.

IBP was developed and implemented for US Trade Secret Misappropriation Law. It employs CATO's factor models for representing and reasoning with cases.

The Uniform Trade Secret Act and the Restatement of Torts have been translated into a high level logical structure of the domain. This domain model captures relations between the 5 major issues of the domain model and there are 5 to 7 factors associated with each issue (factors may be associated with several issues). See Figure 2.2 for the domain model (reproduced from [3]).

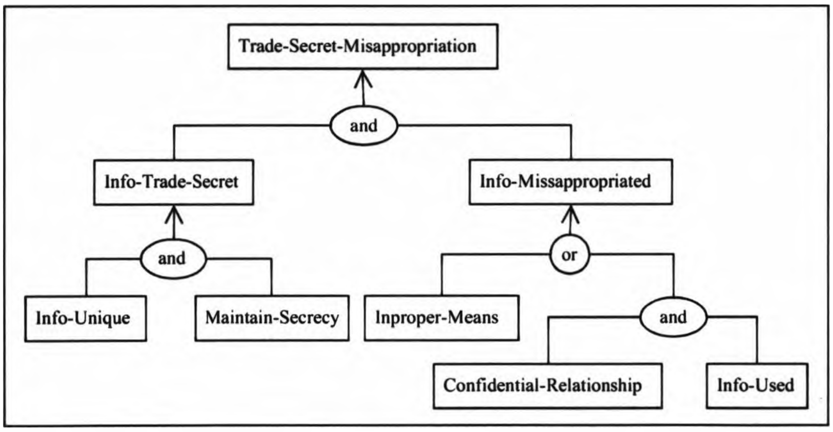

Figure 2.2: IBP's domain model. Reproduced from [3]

To predict the outcome for a case, IBP does the following:

1. Identify issues in a case
2. Determine which party is favoured for each issue. If all the issue related factors favour one party the issue is decided for that party. If there are conflicting factors then IBP uses three case based reasoning functions to resolve the conflicting evidence: Theory-Testing, Explain-Away, and Broaden-Query.
3. Combine the analysis of the issues.

IBP also separates the factors into 3 groups depending on their characteristics.

- KO factors - In almost all the cases where these apply, the case is won by the side it favours.
- Weak Factors - If the issue is represented by an isolated weak factor then IBP does not allow the issue to be discussed.
- Normal Factors - all the rest.

IBP's domain model is a different representation of CATO's factor hierarchy. It separates arguments by issue so that those in conflict can be identified. IBP has proved very successful in experiments ([3], [15], [14] and [16]). The experiments reported will be used as benchmark for later investigations.

2.2.3.3 BankXX

In [33] Rissland et. al. described the implementation of their program BankXX and in [34] they described the evaluation of BankXX. BankXX uses resource constrained heuristic search to search through multiple types of legal knowledge available to create arguments.

The knowledge base, the case domain graph, is a semantic network with legal cases and legal theories represented by nodes with links between them. There are five types of legal case nodes which represent various perspectives useful for human reasoners, and one type of legal theory node.

The argument that BankXX creates contains *pieces* which are filled by BankXX as the execution progresses. The pieces are filled with cases and legal theories which BankXX finds by using a set of neighbour methods to traverse the case domain graph.

When BankXX creates an argument for a new case, it first analyses the new case and creates a claim lattice in the same manner as HYPO in [2] and then randomly chooses one of the most-on-point cases as a starting point and places it onto the *Open* list. The open list contains all the nodes that have been harvested by BankXX during the search. BankXX then enters the execution cycle consisting of three steps:

1. Evaluate the remaining nodes on the *open* list using one of three Heuristic Evaluation Functions and pick the best node. Remove it from the *open* list and move it to the *closed* list. The closed list contains all the nodes which have been used by BankXX in the execution cycle.

2. Apply the predicates for each argument piece and add the current node to each argument piece it satisfies. There may be fill limits on the pieces.

3. Using a set of neighbour methods, BankXX generates all the nieghbours of the current node and adds them to the open list.

BankXX is described here as it is an important work which combines heuristics and argumentation. Surprisingly it has received little subsequent attention. Both our model of argumentation and use of heuristics differ from BankXX: our model of argumentation is derived from the argumentation model of HYPO and our use of heuristics looks to traditional methods of pruning the search space.

2.2.3.4 GREBE

In [12] Branting describes his program GREBE, which uses general legal rules and specific explanations of precedent cases to evaluate legal predicates in new cases. GREBE assesses similarity by attempting to find a pattern of relations in the new case that corresponds to the facts of the precedent case. Each new case is analysed by comparing it to the exemplars it most closely resembles. This depends critically on an accurate assessment of similarity.

The explanation of a past case is a collection of reasoning steps that relate the facts of the case to the solution of a problem in the case. GREBE is queried as to whether a certain conclusion applies to a case. It tries to find the conclusion in the case description and if unable to, then it attempts to construct an explanation of the conclusion.

In [13] Branting describes reasoning with portions of precedent cases. Matching in case-based reasoning can be improved by comparing new cases to portions of precedent cases. The focus of the paper is how to determine the relevant similarities and differences between cases. A new case may match portions of the facts of several precedent cases more strongly than it matches the entire set of facts of any single precedent case.

The strength of the explanations is improved by permitting precedent case constituents from different cases to be combined in a single explanation and by having the flexibility to either apply case-based reasoning to evaluate a predicate or instead reformulating the predicate, depending on which leads to a better match.

We mention GREBE here because our method can lead to the theory being modified by cases seemingly unrelated to the starting case. This can be seen as incorporating portions of precedent cases, and the surrounding case features are ignored

2.2.4 Purposes - Berman and Hafner

Berman and Hafner [11] added a teleological component to case-based reasoners. They introduced the notion of "purposes" or "values". A purpose represents a reason for the factor. Each factor indicates the legal purposes which it advances and each legal purpose in turn specifies whether it favours the plaintiff or defendant. This means that they argue using factors but justify the decision with the purposes it advances.

Teleological or policy decisions affect the decision of the judge in legal cases. Teleological knowledge allows a legal argument system to go beyond factual similarities to include broader jurisprudential concepts. This can make the choice between competing arguments less arbitrary because there can now be a choice between purposes and this can make legal arguments more realistic.

2.2.4.1 Development of Purposes

In the special issue of Artificial Intelligence and Law 2002 in memory of Donald Berman, several authors developed the idea of purposes initially proposed in [11].

In [6], Bench-Capon revisits the notion of purposes or as he terms them "values". The term "value" as used here is not meant as a numerical value or weight but rather as social or legal concepts that are desired. Factors in a case are thought of as promoting certain social values, eg. promoting Less Litigation or promoting Social Equality. The preferences over the values promoted by the factors in the cases provide further orderings over the rules, thus enabling decisions to be explained and new decisions to be made. However, the ordering of the rules may not be complete and so there may be several coherent and competing theories.

In [29], Prakken states that a case should be decided in a certain way because that advances certain values. He uses values to explain rule preferences and hence case decisions. He then illustrates the expressiveness of the system of Prakken and Sartor [30] by applying it to a new class of examples and proposes a formalisation methodology for this class.

In [36], Sartor models legal reasoning as dialectical theory construction directed by teleology. The paper describes theory constructors, factors and values, and cases are evidence to be explained through theories. There is then a dialectical exchange of competing theories. Usually the argumentation process is viewed as consisting in the exchange of arguments, the victory goes to the party proposing the strongest argument. In this paper and [30], argumentation is viewed as being the process through which parties exchange theories which are alternative comprehensive accounts of a controversial domain. The victory goes to the party which succeeds in providing the most coherent theory.

When just using factors, the parties have to resort to arbitrary preferences and the parties fail to provide a theory more coherent than the other. They need to include values and value preferences to determine and explain rule preferences and so avoid arbitrary preferences.

This strand of work reached its culmination in [10], the exploration of the ideas of which are the main contribution of this thesis.

2.3 Theory based Approaches

Work on purposes lead the revival of the notion of theory construction originally proposed by McCarty [26]. This is a way of using both rule-based approaches and case-based approaches. McCarty proposed that two Supreme court Justices, Pitney and Brandeis, were constructing theories to explain how the case of *Eisner v. Macomber* should be decided. Justice Pitney put forth a theory of how he felt the case should have been decided and then Justice Brandeis disputed certain facts and modified the theory to show how he thought the case should be decided. They used rules obtained from the Constitution and precedent cases to construct their theories. They also used hypothetical cases to show how changing certain facts could strengthen their position. McCarty said:

> "The task for a lawyer or a judge in an "hard case" is to construct a theory of the disputed rules that produces the desired legal result, and then to persudae the relevant audience that this theory is preferable to any theories offered by an opponent" ([26], p285).

This leads to the work by Bench-Capon and Sartor, where they describe a model which can be used to construct theories in [8], [9] and [10].

To give an explanation of the role of theories we can consider the ways in which people can disagree in a given case. Suppose I have a case: I may immediately say that it should be found for one of the parties. If the position is accepted there is no need for arguing. But if my intuition is not shared, I will have to give reasons for my view. This will involve citing features of the case which we believe are reasons for deciding for the plaintiff (for example). These reasons are normally called *factors* in AI and Law. Thus I describe the case using terms which tend to support a decision for my view. The person disagreeing with me may now describe the case using factors of his own, which will this time be reasons to decide for the defendant. Such descriptions do not come "written on" the cases: they involve a degree of interpretation. At this point I suppose that the factors used to describe the case have been agreed upon by both parties. I now have a case with a number of reasons to decide it one way and a number of reasons to decide it in the other way. The question now is how do I justify my position in the face of this?

At this point I must ascend a level and introduce *precedent cases*. Precedent cases represent past situations where these competing factors were weighed against one another, and a view of their relative importance taken. On the assumption that new cases should be decided in the same way as past cases, if a past case can be found with the same factors as in the current case, then I can justify my choice using this precedent. If no past cases exactly match or subsume the current case, I argue about the importance of the differences. It is at this level that HYPO-like systems operate: but while they identify the differences, they do not justify acceptance or rejection of the significance of these differences.

To justify these preferences I must ascend a further level. At this level I ask why a factor is a reason for deciding for a given party. I argue that this is because deciding for that party where that factor is present tends to promote or defend some value (legal values eg. Less Litigation or social values eg. Social Equality) that I want to be promoted or defended. This follows the use of *purposes* in Berman and Hafner [11]. The conflict is thus finally stated in terms of competing values/purposes rather than competing cases or competing factors. At this point the solution may be apparent: my set of factors may relate to values which subsume the opponent's values, or be accepted by the opponent as having priority. Beyond this I can only argue about which values should be promoted or defended, and so move beyond positive law, into the realms of politics and general morality. Disagreement is still possible, but no longer a purely legal matter. Laws apply to a community, and this community is held to have common priorities amongst values, and one role of the judge is to articulate these values. Communities can change their values, but to disagree with the decision is to commit to effecting such a change, which is beyond the scope of precedent-based legal argument.

The picture is roughly as follows: factors provide a way of describing cases. A factor can be seen as grounding a defeasible rule. Preferences between factors are expressed in past decisions, which thus indicate priorities between these rules. From these priorities we can abduce certain preferences between values. Thus the body of case law as a whole can be seen as revealing an ordering on values. Figure 2.3 shows the process graphically. As you move up the diagram you are engaged in theory construction and as you move down the diagram you are using the theory.

In [10], the elements of a theory depend on a store of background knowledge consisting of:

- a set of factors - which are an abstraction away from the facts of the case and strengthen the case for one of the parties and are represented as ⟨*factorName, outcome, value*⟩
- a set of cases described using factors which are represented as ⟨*caseName, factorNames, outcome*⟩

When the theory is constructed it consists of:

- a set of cases. This includes the new case to be decided and all the background cases that have been included,
- a set of factors which have been included,

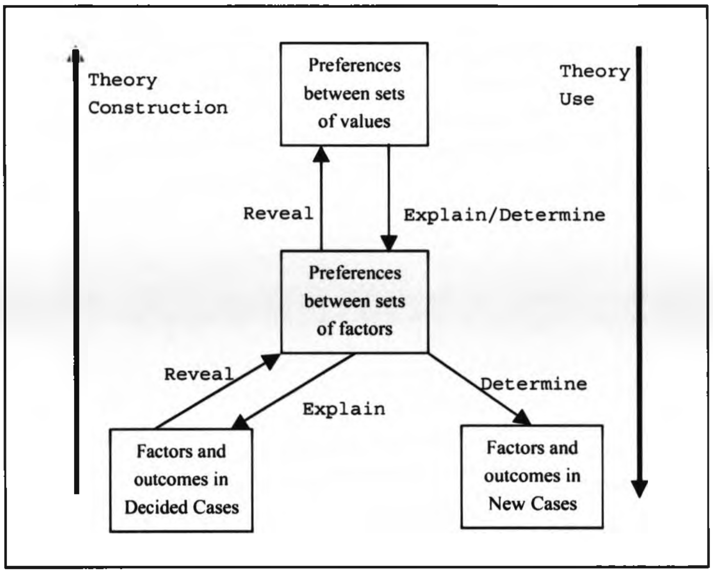

Figure 2.3: Graphical representation of Theory Construction.

- a set of rules. This consists of the primitive rules which correspond to the factors present in the theory and all the rules which have been constructed from the primitive rules,
- a set of rule preferences. The rule preferences can be obtained from precedent cases, from the value preferences or from the arbitrary reason of "because I say so",
- a set of value preferences which correspond to the rule preferences.

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, P>
Theory Factors :
  pLiv
  pNposs
Theory Rules :
  <{pLiv}, P>
  <{pNposs}, D>
Theory Rule Preferences :
  pref(<{pLiv}, P>, <{pNposs}, D>
     <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valPref({MProd}, {LLit})
```

The theory above shows an example theory constructed using the Theory Constructors. Two cases have been included in the theory. Two factors have been included along with their associated primitive rules. *pLiv* is a plaintiff factor and *pNposs* is a defendant factor. There is a rule preference which prefers *pLiv* over *pNposs* and it is supported by the background case of *Keeble*. Finally there is a value preference of *MProd* over *LLit*. This is because *pLiv* promotes the social value of *More Productivity* and *pNposs* promotes the social value of *Less Litigation*. This means that in *Keeble* the court preferred protecting Productivity over reducing the amount of Litigation.

In [10] Bench-Capon and Sartor describe a number of *Theory Constructors* which can be used to add elements to theories as defined above. The theory constructors are:

- *Include Case*. This is used to add the new case and any cases from the case background.
- *Include Factor*. This is used to add the factors to be considered into the theory. This constructor also adds the corresponding primitive rules to the set of rules.
- *Factor Merging*. This constructor merges the primitive rules to produce complex rules that are tailored to particular cases.
- *Rule Broadening*. This constructor removes some of the antecedents from a rule to make it applicable to a case.
- *Preferences From Case*. This constructor allows us to state a preference between two rules which is supported by a case present in the case background.
- *Rule Preference from Value Preference*. This is used to include a rule preference which is supported by a value preference but not by a particular case.
- *Arbitrary Rule Preference*. This constructor allows a rule preference to be included which has no support but needs to be included in the theory.
- *Arbitrary Value Preference*. This also allows a preference to be included with no support.

These theories are used to explain a case if I have a rule which allows me to conclude the outcome of the case on the basis of factors present in the case and this rule is not defeated by any other rule in the theory.

Different theories can be evaluated using several criteria. These include the *explanatory power* of the theory, which is the number of cases included in the theory that can be explained, and the *simplicity* of the theory, which means the fewer factor to explain the same cases. The theories should be *consistent* and free from internal contradictions and contain few *arbitrary preferences*

Bench-Capon and Sartor [10] also describe how the basic model can be extended to use dimensions as described in [2], or to use multi-step arguments like CATO.

The purpose of this thesis to explore these theoretical proposals of Bench-Capon and Sartor in [10] empirically. First we will explore the feasibility of their approach on the adequacy of their theory constructors. Then we will consider how we can automate the approach, developing theories by employing typical argument moves drawn for case-Based Reasoning approaches, and constraining the search space with standard search heuristics.

The ideas and systems outlined in this chapter have all contributed in some way to the work in this thesis. HYPO and CATO bring dimensions and factors and argument moves which can be used in dialogue games. IBP associates factors with issues to predict the outcome of new cases and provides a valuable benchmark. CABARET combines rule based and case based approaches and BankXX uses heuristic search to create arguments. Branting shows that matching to portions of many precedent cases may give better explanations than matching to a single case. Finally, values form an integral part in Theory based reasoning and allow conflicts to be resolved.

Chapter 3

Domain Analysis

In this chapter I describe the two domains used in the experiments described in later chapters. The first domain consists of cases involving the pursuit of wild animals and has been much discussed in the literature and was used to develop and illustrate the approach of Bench-Capon and Sartor [10]. The use of this small domain will enable me to confirm that the implemented tools reproduced the theoretically predicted behaviour in [10]. The second domain is drawn from the area of US Trade Secret Misappropriation Law and is the most sustained line of work in Case Based Reasoning methods in AI and Law. The domain is taken directly from the work of Ashley [2], Aleven [1] and Brüninghaus [3] and permits explicit comparison with these landmark systems.

Domain

3.1 Background Description

The starting point for theory construction is the background of *cases* and the *factors* with which to describe them, which are represented as two files: the *factor background* and the *case background.* Factors, originally used in [1], are particular patterns of facts which may be present in a case and which if present, will provide a prima facie reason for deciding for one or other of the parties to a case. Factors are additionally linked to *values* as in [10] and Section 2.3: this account takes a consequentialist view of legal theory, so that we view decisions as justified by the purposes (or values) they effect. Here a *value* associated with a factor is some desired purpose, which will be promoted by deciding for the party favoured by the factor when the factor is present. The factor background thus consists of a set of 3-tuples of the form <*factor-name,* *outcome-favoured, value-promoted*>.

These factors are used to describe the cases which form the case background. Each case will contain a set of factors from the factor background, and each case will have an outcome. The case background thus comprises a set of 3-tuples of the form <*case-name, factors-present, outcome*>.

These definitions are exactly as described in Bench-Capon and Sartor [10].

3.2 Wild Animal Domain

3.2.1 Cases

The Wild Animal domain consists of three cases involving the pursuit of wild animals. These cases were introduced by Berman and Hafner in [11] and they have been much discussed subsequently. In all of these cases, the plaintiff was chasing wild animals, and the defendant interrupted the chase, preventing the plaintiff from capturing those animals. The issue to be decided is whether the plaintiff has a legal remedy (a right to be compensated for the loss of the game) against the defendant or not.

In the first case, *Pierson v Post*, the plaintiff was hunting a fox on open land in the traditional manner using horse and hound when the defendant killed and carried off the fox. In this case the plaintiff was held to have no right to the fox because he had gained no possession of it.

In the second case, *Keeble v Hickeringill*, the plaintiff owned a pond and made his living by luring wild ducks there with decoys, shooting them, and selling them for food. Out of malice the defendant used guns to scare the ducks away from the pond. Here the plaintiff won.

In the third case, *Young v Hitchens*, both parties were commercial fisherman. While the plaintiff was closing his nets, the defendant sped into the gap, spread his own net and caught the fish. In this case the defendant won. The cases are interesting because many people intuitively feel that the final case should have been decided for the plaintiff rather than the defendant, and the challenge is to come up with a convincing rationale of the actual decision.

3.2.2 Factors

The analysis given here follows that originally presented in Sartor [36]. First we identify four factors:

1. whether the plaintiff did not have possession of the animal,
2. whether the plaintiff owned the land on which the chase was taking place,
3. whether the plaintiff was engaged in earning his living and
4. whether the defendant was engaged in earning his living.

We abbreviate these factors to *pNposs* (plaintiff had no possession), *pLand* (it was the plaintiff's land), *pLiv* (plaintiff earning his living) and *dLiv* (defendant earning his living).

3.2.3 Values

We can now identify the values associated with these factors. By requiring the plaintiff to be actually in possession of the animal we give a clear line following which will tend to reduce litigation in this area, since the satisfaction of any lesser requirement may be open to question. The first factor thus promotes this value, which we abbreviate as *LLit* (less litigation). The second factor promotes respect for property rights, offering more security of these rights (*MSec*). The final two factors will protect economic activity, which should enable more production (*MProd*), to the general benefit of society as a whole.

Our initial starting point thus comprises a factor background which is described in Table 3.1 and a case background which is described in Table 3.2.

Table 3.1: Wild Animal Factor Descriptions

<pNposs, D, LLit>
<pLand, P, MSec>
<pLiv, P, MProd>
<dLiv, D, MProd>

For the Pierson case, the plaintiff did not have possession of the fox because he was still pursuing it so the *pNposs* factor applies. Both the plaintiff and defendant were on open land so the *pLand* factor does not apply. Finally, neither plaintiff or defendant were earning their living. This means that the Pierson case is described by only one factor as shown in Table 3.2.

For the Keeble case, the plaintiff does not have possession of the ducks so the *pNposs* factor applies.[1] The plaintiff was on his own land so the *pLand* factor applies. The plaintiff was engaged in earning his living but the defendant was not so the *pLiv* factor applies but the *dLiv* does not.

Finally for the Young case, the plaintiff does not have possession of the fish because he has not closed his nets and landed the fish on his boat so the *pNposs* factor applies. The plaintiff and defendant are on the open sea so the *pLand* factor does not apply. Finally both the plaintiff and defendant are earning their living so both *pLiv* and *dLiv* apply.

Table 3.2: Wild Animal Case Descriptions

<Pierson, {pNposs}, D>
<Keeble, {pLiv, pLand, pNposs}, P>
<Young, {pLiv, pNposs, dLiv}, D>

3.3 US Trade Secrets Misappropriation Law Domain

For the second domain I draw on the domain used by HYPO [2], CATO [1] and IBP [15]. Where these diverge, I follow the domain used in CATO. CATO uses 26 base level factors, each associated with either the plaintiff (p) or the defendant (d), as given in appendix A, and we will take these as the starting point for our background. CATO, however, does not make use of values, and so we need to identify a set of values and associate them with the factors. These values were first described in [17] and used in [21].

3.3.1 Factors

I based the factor background on the CATO system as described in [1]. This work provides 26 factors, each identified as being pro-plaintiff or pro-defendant. In this paper we use the identifiers for the factors used in [1]. The complete list of factors is given in Table 3.3.

[1]It is sometimes argued that the plaintiff did have possession of the ducks, in virtue of owning the land they frequented (e.g.[7]). Here, however, I follow the more usual interpretation of Sartor [36]).

Table 3.3: Factors in CATO (NB: There is not F9 in [1])

Pro Plaintiff Factors	*Pro Defendant Factors*
F2 Bribe Employee	F1 Disclosure in Negotiations
F4 Agreed not to disclose	F3 Employee Sole Developer
F6 Security Measures	F5 Agreement not specific
F7 Brought Tools	F10 Secrets Disclosed Outsiders
F8 Competitive Advantage	F11 Vertical Knowledge
F12 Outsider Disclosures Restricted	F16 Info Reverse Engineerable
F13 Noncompetition Agreement	F17 Info Independently Generated
F14 Restricted Material Used	F19 No Security Measures
F15 Unique Product	F20 Info Known to Competitors
F18 Identical Products	F23 Waiver of Confidentiality
F21 Knew Info Confidential	F24 Info Obtainable Elsewhere
F22 Invasive Techniques	F25 Info Reverse Engineered
F26 Deception	F27 Disclosure in Public Forum

3.3.2 Values

The final element needed for the factor background is values. CATO, however, does not use this notion, and so I need to supply this element of the background ourselves. So what values seem to underlie the factors? Values relate to behaviour that the law wishes to encourage or discourage. The motive for encouraging or discouraging behaviour is to promote some socially desirable end. For example, marking *F1-Disclosure in Negotiation* as an important consideration would promote the social end that people act with reasonable care for their own interests: if one has a secret one has a certain responsibility to keep it to oneself. I therefore examined the factors to identify patterns of behaviour which they encouraged or discouraged.

First a number of factors relate to confidentiality agreements. Clearly if all trade secret disputes were governed by a specific agreement, the task of deciding them would be much eased. I would therefore expect the law to encourage such agreements to be made. My first value then is *Confidentiality Agreement (CA)*: the side favoured will depend on the nature of the agreement. This value secures five factors:

- F4 Agreed not to disclose (p)
- F5 Agreement not specific (d)
- F13 Noncompetition Agreement (p)
- F21 Knew Info Confidential (p)

- F23 Waiver of Confidentiality (d).

Next it seems that the law does not wish to condone lax behaviour, so that it wishes people with secrets to take reasonable measures to protect them. This gives the second value *Reasonable Efforts (RE)*. Making such efforts are encouraged if having made them favours the plaintiff, and having failed to make them favours the defendant. Six factors share this value.

- F1 Disclosure in Negotiations (d)
- F6 Security Measures (p)
- F10 Secrets Disclosed Outsiders (d)
- F12 Outsider Disclosures Restricted (p)
- F19 No Security Measures (d)
- F27 Disclosure in Public Forum (d).

Third the law wishes to encourage competition by legitimate means. Therefore if a person can develop the product using *Legitimate Means (LM)*, this should tell in their favour. This covers eight factors. Note that one of them is pro-plaintiff; the uniqueness of a product creates a presupposition that it cannot be developed by legitimate means, and so places an extra burden of proof on the defendant.

- F3 Employee Sole Developer (d)
- F11 Vertical Knowledge (d)
- F15 Unique Product (p)
- F16 Info Reverse Engineerable (d)
- F17 Info Independently Generated (d)
- F20 Info Known to Competitors (d)
- F24 Info Obtainable Elsewhere (d)
- F25 Info Reverse Engineered (d)

The reverse of this is that illegal or immoral means should be discouraged. Five factors relate to this value, *Questionable Means (QM)*, which always favours the plaintiff:

- F2 Bribe Employee (p)
- F7 Brought Tools (p)
- F14 Restricted Material Used (p)
- F22 Invasive Techniques (p)
- F26 Deception (p)

The final two factors are intended to show that the secret had *Material Worth (MW)*. The law would naturally attempt to discourage litigation about secrets of no worth, and so will favour the plaintiff if his secret had demonstrable value. Two factors, both of which favour the plaintiff, are used here:

- F8 Competitive Advantage (p)
- F18 Identical Products (p)

I have now assigned the factors to five values. Conveniently the distribution is reasonably equal, with only *Material Worth* represented by substantially fewer factors.

The complete factor Descriptions used in the Factor Background are shown in Table 3.4.

Table 3.4: Factor Background.

Pro Plaintiff Factors	*Pro Defendant Factors*
<F2, P, QM>	<F1, D, RE>
<F4, P, CA>	<F3, D, LM>
<F6, P, RE>	<F5, D, CA>
<F7, P, QM>	<F10, D, RE>
<F8, P, MW>	<F11, D, LM>
<F12, P, RE>	<F16, D, LM>
<F13, P, CA>	<F17, D, LM>
<F14, P, QM>	<F19, D, RE>
<F15, P, LM>	<F20, D, LM>
<F18, P, MW>	<F23, D, CA>
<F21, P, CA>	<F24, D, LM>
<F22, P, QM>	<F25, D, LM>
<F26, P, QM>	<F27, D, RE>

An alternative method of assigning factors to values would be to use the Factor Hierarchy defined by Aleven in CATO ([1]). In this hierarchy all the base level factors are linked to high level Legal Issues also called Abstract Factors. There are several

Abstract Factors with a similar meaning to my values, for instance, *F114-Confidential-Relationship* in CATO and the value of *Confidentiality Agreement*. A second method would be to use the "Logical Model" of Brüninghaus described in IBP ([15], [14] and [16]). In this model the base level factors are linked to high level Legal Issues which have been found using the Uniform Trade Secret Act and the Restatement of Torts. These Legal Issues are also very similar to my meaning of values. In both of these alternatives some of the factors are linked to more that one Abstract Factor or Issue, but in my version each factor only promotes a single value. This aspect of a factor being linked to several Legal Issues or promoting several values is explored further in Chapter 5 when I look at complex dimensions.

3.3.3 Cases

I must now select a set of cases. Unfortunately the majority of cases from Ashley's group are not made available. However, a number are described in [1] and [15] and I choose a selection of these, although excluding some of the cases flagged as problematic by Aleven. My initial selection, used in [17] included seven cases found for the plaintiff and seven cases found for the defendant. I continued to use these cases, which I call *Group 1*, as the initial data set with which to construct our theories. In order to test whether the theories generalised to cases not used in their construction I added another group of cases (*Group 2*) also taken from published work ([3], [14]). In Table 3.5 I show the case, the factors present, split according to whether they favour the plaintiff or the defendant, and the outcome of the case.

Table 3.5: Factor Based Cases Used in Future Experiments.

Case	*Pro-P factors*	*Pro-D factors*	*Outcome*
Group 1			
Arco		F10, F16, F20	D
Boeing	F4, F6, F12, F14, F21	F1, F10	P
Bryce	F4, F6, F18, F21	F1	P
College Watercolour	F15, F26	F1	P
Den-Tal-Ez	F4, F6, F21, F26	F1	P
Ecologix	F21	F1, F19, F23	D
Emery	F18, F21	F10	P
Ferranti	F2	F17, F19, F20, F27	D
Robinson	F18, F26	F1, F10, F19	D
Sandlin		F1, F10, F16, F19, F27	D
Sheets	F18	F19, F27	D
Space Aero	F8, F15, F18	F1, F19	P
Televation	F6, F12, F15, F18, F21	F10, F16	P
Yokana	F7	F10, F16, F27	D
Group 2			
CMI	F4, F6	F10, F16, F17, F20, F27	D
Digital Development	F6, F8, F15, F18, F21	F1	P
FMC	F4, F6, F7, F12	F10, F11	P
Forrest	F6, F15, F21	F1	P
Goldberg	F21	F1, F10, F27	P
KG	F6, F14, F15, F18, F21	F16, F25	P
Laser	F6, F12, F21	F1, F10	P
Lewis	F8, F21	F1	P
MBL	F4, F6, F13	F5, F10, F20	D
Mason	F6, F15, F21	F1, F16	P
MineralDeposits	F18	F1, F16, F25	P
National Instrument	F18, F21	F1	P
National Rejectors	F7, F15, F18	F10, F16, F19, F27	D
Reinforced	F4, F6, F8, F15, F21	F1	P
Scientology	F4, F6, F12	F10, F11, F20	D
Technicon	F6, F12, F14, F21	F10, F16, F25	P
Trandes	F4, F6, F12	F1, F10	P
Valco-Cincinnati	F6, F12, F15, F21	F1, F10	P

3.4 Experiments

For many of the experiments described in this research the case of *Mason versus Jack Daniels* is used as the case requiring a decision. The Mason case is described as follows [1].

> "In 1980, a restaurant owner named Mason developed a combination of Jack Daniel's whiskey, Triple Sec, sweet and sour mix, and 7-Up to ease a sore throat. He promoted the drink, dubbed "Lynchburg Lemonade" for his restaurant, "Tony Mason's, Huntsville," served it in Mason jars and sold t-shirts. Mason told the recipe only to his bartenders and instructed them not to reveal the recipe to others. The drink was only mixed out of customer's view. Despite its extreme popularity (the drink comprised about one third of the sales of alcoholic drinks), no other establishment had duplicated the drink, but experts claimed it could easily be duplicated. In 1982, Randle, a sales representative of the distillery, visited Mason's restaurant and drank Lynchburg Lemonade. Mason disclosed part of the recipe to Randle in exchange, Mason claimed, for a promise that Mason and his band would be used in a sales promotion. Randle recalled having been under the impression that Mason's recipe was a "secret formula". Randle informed his superior of the recipe and the drink's popularity. A year later, the Distillery began using the recipe to promote the drink in a national sales campaign. Mason did not participate in the promotion or receive other compensation."

This is an interesting case because it is quite finely balanced, and has been discussed extensively by Aleven.[2]

[2] it is also parodied in a Simpsons episode in which Homer invents a cocktail called "The Flaming Mo" which makes Mo's tavern (briefly) world famous.

Chapter 4

Factor Based Theory Construction

The first step in this research is to ensure that the theoretical descriptions of the Theory Constructors defined by Bench-Capon and Sartor in [10] can be realised as a practical system. To do this the program CATE (CAse Theory Editor) had to be constructed to implement the Theory Constructor definitions to ensure that the factor and case background supplied the requisite information and the Theory Constructors could be used to construct theories. In [10] the authors left several design choices especially with respect to comparison of factors and values and these choices will be explored using CATE in later sections.

4.1 CATE Theory Construction Tool

4.1.1 Purpose of CATE

CATE is a tool developed to provide support to the process of understanding a legal domain through theory construction. CATE is intended to be useful both to lawyers exploring their understanding of a set of cases, and to knowledge engineers desirous of building an automated system. By providing a means rapidly to develop and execute theories, the tasks of exploring alternatives and refining initial intuitions is greatly eased.

CATE is designed to embody the set of Theory Constructors given in [10] and described in section 2.3 and has been implemented in Java.

Because CATE starts from a case and factor background, it requires that a domain has already been analysed to identify factors with which to describe the cases, and to provide a set of case descriptions in terms of these factors. CATE is not restricted to any particular domain and so can be used with any domain for which the analysis to supply the requisite background is available. The case and factor backgrounds are loaded into CATE from previously prepared files.

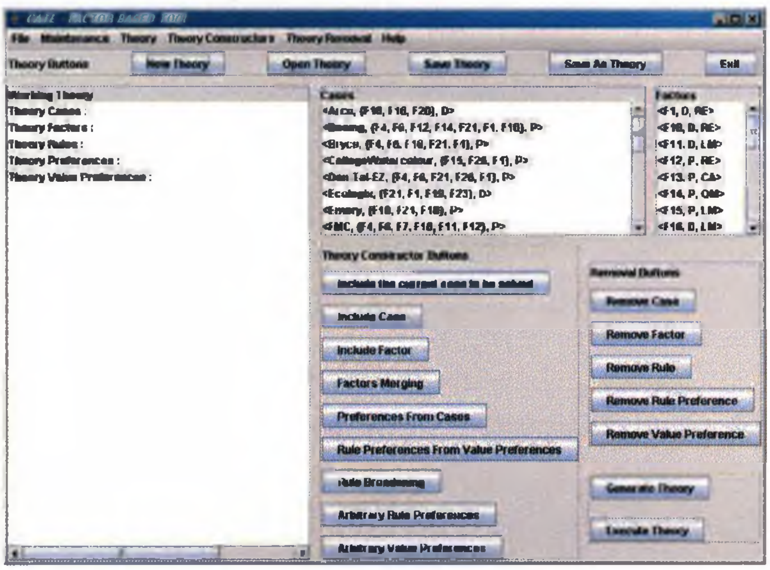

Figure 4.1: CATE Program.

Figure 4.1 gives a screen shot of CATE. The three panels show the theory which is being constructed, the case descriptions present in the domain and the factor descriptions present in the domain.

The *Theory Buttons* operate in the usual way to other applications. The *New Theory* button starts a new theory, the *Open Theory* button asks the user to choose a theory to open, the *Save Theory* button saves the theory to a previously declared name, the *Save As Theory* button saves the theory with a new name (or just with a name if the theory has not been saved before) and the *Exit* button closes CATE after checking if the theory has been saved.

The *Theory Constructor* buttons invoke the various Theory Constructors described in section 2.3, each of which will invoke a dialogue box appropriate to the particular constructor.

- *Include Current Case to be Decided*: This is not a Theory Constructor defined by Bench-Capon and Sartor but it is included here because it allows a case that is not in the case background (or a case that is modified) to be included into the theory.
- *Include Case*: This adds a case from the case background to the theory. There is the option to add all the cases at once.
- *Include Factor*: This adds a factor from the factor background to the theory. Additionally the factor adds a rule expressing that the factor is a (defeasible) reason to decide for the party it favours. This also has the option to add all the factors at once.
- *Factors Merging*: Given a rule in the theory, the antecedent may be strengthened to give a new rule. Antecedents may be strengthened only by the addition of another factor favouring the same party to the dispute.
- *Rule Broadening*: Given a rule, the antecedent may be weakened to give a new rule by omitting one of the factors from the antecedent.
- *Rule Preference from Cases*: Given a case in the theory to which two rules, each favouring a different party are applicable, a preference may be inferred for the party which won the case and hence be added to the theory. Moreover, from this rule preference it may be inferred that the set of values promoted by following the preferred rule are preferred to those promoted by following the other rule, and this is added to the theory. CATE checks that the case does support this rule preference and if it does, then the rule preference and associated value preference are included into theory with the rule preference being labelled with its supporting case.
- *Rule Preference from Value Preference*: Given a value preference in the theory and two rules corresponding to the related sets of values, I can deduce that the rule relating to the preferred value is preferred to the other rule, and include this. The rule preference is labelled with $<|$*From Value Preference*$|>$ to distinguish it from any other rule preference.

- *Arbitrary Rule Preference*: Using this constructor a preference is added to the theory, even though no case can be found to justify it. This rule preference is labelled with <|*Arbitrary Rule Preference*|>.

- *Arbitrary Value Preference*: Using this constructor a preference is added to the theory, even though no rule preference can be found to justify it.

CATE also provides some checking on the legality of use of the constructors: when a user specifies preferences over rules or values, CATE checks that the resulting theory is consistent by comparing the existing rule and value preferences and the new rule and value preferences. If adding the preference would make the theory inconsistent, which means that the new rule or value preference is the complete opposite of an existing rule or value preference, then a warning is issued and the preference is not added. If a user still wishes to include the preference, then they must first remove an existing preference causing the conflict.

The *Removal* buttons enable the user to modify the theory by removing items from the theory. The result of this can be complicated if many linked things have to be removed.

- *Remove Case*: Allows the user to choose a Case to remove. If the Case is providing support to a Rule Preference the user is asked if they want to remove it or replace it with an Arbitrary Rule Preference. If the user decides to remove the Rule Preference then the corresponding Value Preference is also removed unless it is also related to another Rule Preference.

- *Remove Factor*: Allows the user to remove a Factor. It will also remove the primitive Rule, any Rule containing the Factor and any Rule Preference containing it along with the corresponding Value Preference.

- *Remove Rule*: Allows the user to remove a complex Rule from the theory but not a primitive Rule which has to be removed with the *Remove Factor* button. Any Rule Preferences containing the Rule are also removed along with the Value Preference.

- *Remove Rule Preference*: Allows for the removal of a Rule Preference and corresponding Value Preference.

- *Remove Value Preference*: This allows the user to remove an arbitrary Value Preference but not one which is related to a Rule Preference still in the theory.

The *Generate Theory* and *Execute Theory* buttons take the constructed theory and translate it into Prolog code which can then be executed. Section 4.2 will give an explanation of how the Prolog code is generated and executed.

The *Generate Theory* button simply takes the theory and translates it into Prolog code. The *Execute Theory* button takes the Prolog Code and executes it. For a new theory for which the Prolog code has not yet been generated the *Execute Theory* button will also generate the Prolog code and then execute it, but if there is already Prolog code then it will simply execute the existing code. This means that if the user wishes to modify the Prolog code manually they can then execute this modified code. However if the user modifies the theory, new Prolog code has to be generated using the *Generate Theory* button and then execute the new code.

4.2 Constructing and Executing a Theory

The theory is created by selecting buttons to include items in the theory. To build a simple theory using CATE, the user must first select some cases to include in the theory by selecting the *Include Case* button and choosing the desired cases. Next some factors to be used in the theory are selected by selecting the *Include Factor* button and choosing the factors. Next the user can add rule preferences by selecting the *Preference From Cases*, *Rule Preference From Value Preference* or *Arbitrary Rule Preference* buttons. The theory can now be executed.

The code is generated from the theory in the following way. CATE first takes the value preferences and translates them into rule preferences, by substituting the factors related to the values for the values. It then adds these to the original rule preferences and can start ordering the rules. The standard execution of Prolog assigns priority to its clauses in accordance with the order in which they appear in the program: thus the rule preferences are enforced in the program by ordering the rules themselves.

The rules forming the rule preferences are divided into three groups. The left group, which contains only those rules which are always most preferred, and so only appear on the left side of the rule preferences. The right group, which contains only those rules which are always least preferred and so only appear on the right side of the preferences.

Finally the middle group, which contains the rules which appear on both sides of rule preferences. Figure 4.2 shows the process of how CATE sorts the rules. The four rules forming the three rule preferences are sorted into the correct groups. Rule A only appears on the left hand side of the preferences and so is placed in the left group. Rule D only appears on the right hand side of the preferences and so is placed in the right group. Rules B and C each appear on the left hand side of one rule preference and the right hand side of a different rule preference so these are placed in the middle group. The left and right groups are sorted alphanumerically and saved. The process is recursively applied to the middle group so that it is now sorted into the three groups and this process continues until there are no rules in the middle group.

Rule Preferences

1) A > B
2) B > C
3) C > D

First Pass

LeftGroup	Middle Group	Right Group
A	B	D
	C	

Second Pass

Left Group	Middle Group	Right Group
A		C
B		D

Finished List

A
B
C
D

Figure 4.2: Three rule preferences and how the rules are sorted into a finished list.

Each time the process repeats the new left rules are placed below the old left rules and the new right rules are placed above the old right rules. When the process is complete the right rules are placed below the left rules giving a complete ordering of the rules with the most preferred rules at the top of the list and the least preferred rules at the bottom. Any rules which do not appear in a rule preference and hence are not in the list are placed at the bottom of the sorted list.

Note that the theory determines only a partial order on the rules. Each time I have, for example, a left group, I know that these rules have a higher priority than the rules in the other two groups. I do not, however, have any information as to the priority of the rules *within* the left group. The theory thus determines a family of programs, each consistent with the theory, but differing as to the ordering of rules with indistinguishable priorities. Initially CATE simply uses an alphanumeric ordering within groups. This means that the program may need refinement within the constraints, adjusting the order of rules within a group to produce the desired behaviour (or to select the most appropriate theory). This refined program will be an alternative expression of the theory, and so the theory itself will need no modification.

CATE then takes the sorted list of rules and translates them into Prolog Clauses. This is a straightforward matter of mapping from the rule syntax into Prolog syntax. For example, the rule *((F1) → D)* is transformed to *outcome(X, d) :- factor(X, f1)*. The more complex rule *((F1, F10, F19) → D)* is transformed to *outcome(X, d) :- factor(X, f1), factor(X, f10), factor(X, f19)*. It then includes the cases and their factors, which will supply the facts for the execution of the program and saves the complete program, rules and facts.

To execute the theory, CATE takes the Prolog code and for each case searches through the sorted clauses to find the first clause to fire for the case. This clause gives an outcome to the case. These case outcomes can then be used to evaluate how the theory performs with respect to the actual decisions for the cases, so as to see the extent to which the theory does indeed explain the selected cases.

4.3 Wild Animal Study

In [10] Bench-Capon and Sartor illustrated the use of the Theory Constructors with a small example using the Wild Animal Domain described in Section 3.2. This simple domain will illustrate the operation of CATE and reconstruction of the theories given in [10]. In that paper they constructed four theories, and I will now use CATE to replicate their generation, and generate the corresponding code.

Theory 1 is constructed for the defendant. First the *Include Case* Constructor is used to include the *Pierson* case and the defendant based *Young* case. Then the *Include*

Factor Constructor is used to include the *pNposs* factor in the theory. The theory produced by CATE is this:

```
Theory Cases :
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pNposs
Theory Rules :
  <{pNposs}, D>
Theory Rule Preferences :
Theory Value Preferences :
```

This theory can be used to generate Prolog code which is executed to produce the outcome for the cases according to the theory. We show the cases, the outcomes and the decisive rule.

```
pierson | d |  outcome(X, d) :- factor(X, pnposs).
young   | d |  outcome(X, d) :- factor(X, pnposs).
```

Both cases have been decided for the defendant as there was only a defendant factor present in the theory. This theory adopts a very straightforward approach: considering only a single factor, possession being seen as the whole of the law. Theory 2 is next constructed for the plaintiff. It extends Theory 1 by including the *Keeble* case using the *Include Case* Constructor and then the *Include Factor* Constructor is used to include the *pLiv* factor. This gives two conflicting rules applying to *Keeble*. However, it is known how this conflict was resolved: *Keeble* was found for the plaintiff. This case can therefore be used to give a rule preference which prefers *pLiv* factor over the *pNposs* factor. This rule preference also generates a value preference which is also included in the theory.

```
Theory Cases :
  <Pierson, {pNposs}, D>
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, P>
Theory Factors :
  pLiv
  pNposs
Theory Rules :
  <{pLiv}, P>
  <{pNposs}, D>
Theory Rule Preferences :
  pref(<{pLiv}, P>, <{pNposs}, D>
     <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valPref({MProd}, {LLit})
```

The theory can be executed to produce the following outcomes for the cases. *Young* and *Keeble* have been decided for the plaintiff because the plaintiff factor is preferred over the defendant factor.

```
pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pliv).
young   | p | outcome(X, p) :- factor(X, pliv).
```

Theory 3 is constructed for the defendant and adds *pLand* to Theory 2. This factor is merged with *pLiv* to produce a rule with *(pLand, pLiv)* as antecedent. The preference in *Keeble* can now be explained in terms of this rule, giving the rule preference of *(pLand, pLiv)* over *pNposs* instead of *pLiv* over *pNposs* as in Theory 2. However, this will not explain how *Young* should be decided. To do this an arbitrary rule preference of *pNposs* preferred over *pLiv* has to be included (because the defendant is constructing the theory).

```
Theory Cases :
  <Pierson, {pNposs}, D>
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLand
  pLiv
  pNposs
Theory Rules :
  <{pLand, pLiv}, P>
  <{pLand}, P>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Rule Preferences :
  pref(<{pLand, pLiv}, P>, <{pNposs},D>)
     <|<Keeble, {pLiv, pLand, pNposs}, P>|>
  pref(<{pNposs}, D>, <{pLiv}, P>)
     <|<Arbitrary rule Preference>|>
Theory Value Preferences :
  valPref({LLit}, {MProd})
  valPref({MProd, MSec}, {LLit})
```

This theory decides *Young* in the way wanted but has resorted to an arbitrary rule preference, which is not desirable.

```
pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pland),
              factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, pnposs).
```

An alternative to Theory 3 is Theory 4 which is constructed by including *dLiv* instead of *pLand* and merging it with the *pNposs* factor to give a rule with antecedent *(dLiv, pNposs)*. The value preference of *(LLit, MProd)* over *MProd* is added (which seems justifiable as the preferred value is a superset of the less preferred value) and from this the rule preference of *(dLiv, pNposs)* over *pLiv* is derived.

```
Theory Cases :
  <Pierson, {pNposs}, D>
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  dLiv
  pLiv
  pNposs
Theory Rules :
  <{dLiv, pNposs}, D>
  <{dLiv}, D>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Rule Preferences :
  pref(<{dLiv, pNposs}, D>, <{pLiv}, P>)
     <|<From Value Preference>|>
  pref(<{pLiv}, P>, <{pNposs},D>)
     <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valPref({LLit, MProd}, {MProd})
  valPref({MProd}, {LLit})
```

When executed, this theory gives the required decision for *Young*.

```
pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, dliv),
              factor(X, pnposs).
```

Table 4.1: The rules, rule preferences and value preferences for the four theories.

Theory	Rules	Rule Preference	Value Preference
1	(1) pNposs→D		
2	(1) pNposs→D (2) pLiv→P	(2)>(1)	MProd>LLit
3	(1) pNposs→D (2) pLiv→P (3) pLand→P (4) (pLiv,pLand)→P	(4)>(1) (1)>(2)	(MProd,MSec)>LLit LLit>MProd
4	(1) pNposs→D (2) pLiv→P (3) dLiv→D (4) (dLiv,pNposs)→D	(2)>(1) (4)>(2)	MProd>LLit (LLit,MProd)>MProd

The rules, rule preferences and value preferences are reproduced for theory in table 4.1. Theory 1 has one rule and no preferences, whereas Theory 2 has two rules and a preference of the second rule over the first. Theories 3 and 4 both contain four rules but each has different rule preferences.

All of this faithfully reproduces the example of [10]. This example illustrates the benefits of using CATE to assist in coming to an understanding of the domain. By incrementally constructing the theory I can develop a broad understanding and then refine it to accommodate cases not yet explained. At any point I have a clear statement of the theory and can check its implications by executing it. This ensures that I can recognise when additional preferences are required to complete the theory; and that the theory has the desired effects, and if it does not, the reasons for the undesired effects are identified. Adding preferences is constrained by the need to keep the theory consistent. The ability to experiment with different theories also helps to identify false moves: in [11] it is said that students are often misled into incorporating the fact about the ownership of land in *Keeble*, which leaves them unable to explain *Young*. This move corresponds to Theory 3 above, in which the need for an arbitrary preference shows the deficiencies of the theory, allowing me to explore the more acceptable alternative of Theory 4.

This small example shows that CATE does indeed support theory construction as described in [10]. I will now describe a number of research questions, raised but not answered by [10]. These will then be explored by modelling of the larger domain of Trade Secret Misappropriation Law with CATE.

4.4 Research Questions

While Bench-Capon and Sartor [10] lay out a general approach to modelling reasoning with legal cases, they leave a number of points of detail open. The experiments described in this section and those following are intended to cast light on how these issues should be resolved. The focus, of these experiments will not be on the process of theory construction, but rather on how theories are applied once constructed. The standpoint therefore is that of a knowledge engineer attempting to construct a theory of the domain, rather than a lawyer reasoning from the facts of a particular case. To carry out the experiments I will, however, need to construct some theories. There are several issues I need to consider in constructing these theories. First there is the question of how the rules of the theory should be extracted from the past cases.

Our first question is therefore:

Q1: How should we select cases and extract rules for inclusion in the theory?

Once cases have been chosen and rules identified, a similar question arises with respect to factors. Within the general approach it still remains possible to include or exclude the factors that appear in the cases and the domain analysis generally. This gives rise to the second question:

Q2: Should we be inclusive or exclusive with regard to factors?

In particular, it is part of the philosophy of [10] that preferences between factors determine preferences between values, which can then in turn determine preferences between other factors relating to those values. If this is so, I should expect to be able to include additional factors pertaining to these values without major revisions to the preferences of theory. Thus I may pose a third question:

Q3: Is there evidence to suggest that values can be used to determine the relative importance of factors?

A key role of the theory is to explain preferences between rules in terms of a comparison between the sets of values promoted by the factors contained in the rules. The fourth question therefore is:

Q4: How should sets of values be compared?

A related question concerns what should be done when a case contains two factors relating to the same value, but favouring different sides. In such cases I must prefer a factor rather than a value. Our fifth question is

Q5: Is it possible to use a general principle to pre-order factors within a value?

In [10] it is assumed that factors promote values to an equal degree: if a rule contains a factor then following that rule is held to promote the value associated with the factor. It is not impossible, however, that different factors will promote different values to different degrees, and that this possibility needs to be considered. This leads to the sixth question:

Q6: Is there evidence to suggest that factors promote values to different degrees?

Again, [10] does not allow for accumulation of factors: if following a rule promotes a value, then it is not relevant whether this is the result of the presence of one factor or several. Perhaps, however, a value is promoted to a greater extent if several factors promoting that value are present. The seventh question is thus:

Q7: Is there evidence to suggest that values and/or factors have a cumulative effect?

I see these questions as central to the effective modelling of reasoning with cases in law in the manner of [10]. While I acknowledge that the experiments on a single domain with a limited number of cases described here cannot produce definitive answers. My hope is that the experiments may help to move the debate forward, and indicate which lines may be worth pursuing.

4.5 Building a Theory

I am exploring these questions in the context of US Trade Secret Misappropriation Law as described in Chapter 3. The background used was described in section 3.3.

4.5.1 Need for a method

Having established the background, I can proceed to construct theories. Often theory construction is directed towards a particular, as yet undecided, case. Here, however, I am trying to come to a theory which will explain as many of the available cases as possible, and so I need to choose the cases. For guidance I need some principles for how I will construct the theory. To explore a range of possibilities, I decided to use three approaches.

For each approach I restricted the theory to four cases, selected from the *Group 1* cases in Table 3.5 to construct the theory, two won by the plaintiff and two won by the defendant. The other cases in *Group 1* will then be used to assess the theory, and if necessary to refine it. Once the theory has been refined, it will then be tested against the *Group 2* cases to see if it generalises so as to classify new cases correctly. The relative success of the three methods will help me to answer Q1 of the research questions.

4.5.2 The "Safe" Method

Using the safe method I will say no more than I am strictly justified in saying from a consideration of the cases. I will not attempt to generalise beyond them, nor impose any preconceptions as to how the domain should be. In this method I am willing to include as many factors as possible, but will produce rules which do not go beyond the minimum that I am entitled to infer. This latter effect is given by using the method of Prakken and Sartor for producing rules from cases given in [30], whereby the conjunction of all the pro-plaintiff factors present gives one rule, the conjunction of all the pro-defendant factors gives another, and the priority is determined by the decision.

For this method I selected *Emery* and *College Watercolour* as plaintiff cases and *Robinson* and *Sheets* as defendant cases, the idea being that these would generate the most powerful rules since they involve the fewest factors. Representing the rules from these four cases in the manner of [30] yields the rule and value preferences shown in Figure 4.3.

```
Theory Preferences :
  pref(<{F1, F10, F19}, D>, <{F18, F26}, P>)
    <|<Robinson, {F1, F10, F18, F19, F26}, D>|>
  pref(<{F15, F26}, P>, <{F1}, D>)
    <|<CollegeWatercolour, {F1, F15, F26}, P>|>
  pref(<{F18, F21}, P>, <{F10}, D>)
    <|<Emery, {F10, F18, F21}, P>|>
  pref(<{F19, F27}, D>, <{F18}, P>)
    <|<Sheets, {F18, F19, F27}, D>|>
Theory Value Preferences :
  valpref({MW, CA}, {RE})
  valpref({RE}, {MW, QM})
  valpref({LM, QM}, {RE})
  valpref({RE}, {MW})
```

Figure 4.3: Rule and Value Preferences from the "Safe" Theory.

Table 4.2: The rules, rule preferences and value preferences for the "Safe" theory.

Rules	Rule Preference	Value Preference
(1) F1→D	(9)>(13)	RE> (MW, QM)
(2) F10→D	(11)>(1)	(LM, QM)>RE
(3) F15→P	(12)>(2)	(MW, CA)>RE
(4) F18→P	(10)>(4)	RE>MW
(5) F19→D		
(6) F21→P		
(7) F26→P		
(8) F27→D		
(9) F1, F10, F19→D		
(10) F19, F27→D		
(11) F15, F26→P		
(12) F18, F21→P		
(13) F18, F26→P		

Table 4.2 shows all the rules included in the theory and the four rule preferences and four value preferences. There are several primitive rules which are not used in the rule preferences but they are included because they have been merged to form complex rules.

4.5.3 The "Simple" Method

The second method was intended to produce the simplest theory. Here I want to use the fewest possible number of factors, and am willing to make assumptions which enable me to produce rules not strictly justified by the cases. Hence I selected a small set of factors which cover all the cases, and choose cases to establish priorities between them. The motivation here is similar to the automatic induction of decision trees, which strives to produce the smallest tree capable of classifying the instances on the data available.

For this method I must first select the factors. What I need is a set of factors such that at least one pro-plaintiff factor occurs in every case decided for the plaintiff, and at least one pro-defendant factor occurs in every case decided for the defendant using all the cases from Group 1. *F21-Knew-Info-Confidential* occurs in 6 of the 7 pro-plaintiff cases, so I chose this together with *F15-Unique-Product* to handle *Space Aero*. For the pro-defendant factors, *F19-No-Security-Measures*, *F20-Info-Known-To-Competitors* and *F27-Disclosure-in-Public-Forum* will cover all defendant cases. Now I only need to express preferences where there is both a pro-plaintiff and a pro-defendant factor in the same case. In only two cases do is there a conflict to resolve: *Space Aero* and *Ecologix*, so I express preferences according to the outcomes of these two cases. For this approach, nothing is to be gained by including additional cases, so only these two are used in this theory. I thus get the following rule and value preferences shown in Figure 4.4. Table 4.3 shows the three rules in the theory, the two rule preferences and the value preferences. Because it is a simple theory only primitive rules are used and only a very small number of them.

```
Theory Preferences :
  pref(<{F15}, P>, <F19}, D>)
    <|<SpaceAero, {F1, F8, F15, F18, F19}, P>|>
  pref(<{F19},D>, <F21}, P>)
    <|<Ecologix, {F1, F19, F21, F23}, D>|>
Theory Value Preferences :
  valpref({LM}, {RE})
  valpref({RE}, {CA})
```

Figure 4.4: Rule and Value Preferences from the "Simple" Theory.

Table 4.3: The rules, rule preferences and value preferences for the "Simple" theory.

Rules	Rule Preference	Value Preference
(1) F15→P	(1)>(2)	LM>RE
(2) F19→D	(2)>(3)	RE>CA
(3) F21→P		

4.5.4 The "Value Driven" Method

The third approach will be value driven, thus embodying some pre-determined assumptions about how I believe the domain operates. Here I will first reflect on the values and produce a ranking. I then choose factors to represent these values, and cases to establish the desired value order.

For this approach I must first decide on a value order. I do not distinguish between *Questionable Means* and *Material Worth*, since these both always favour the same side (the plaintiff). I might suppose that the most highly rated value is *Confidentiality Agreement*, since if all the dealings were regulated by properly drafted agreements, there would be no problems to decide. I rate *Legitimate Means* next: in the absence of a specific agreement, the right to enterprise must be protected. I rate *Reasonable Efforts* third, since people must take some steps to protect themselves. This leaves *Questionable Means* and *Material Value* at the bottom. Is *Material Worth* so unimportant, when surely it a *sine qua non* for an action? Well, it is of little importance here, since while if it is not present the action seems pointless, it does not really cast much light on whether the defendant behaved incorrectly. It does not, in fact, appear in every case. I assume that this is because it was accepted by both sides, and so is made explicit only if the matter is raised in an effort to discredit the action. Arguably also, the presence of pro-plaintiff *CA* and *RE* factors implies *MW*.

In order to establish this order on values I need four cases. In choosing representative factors I should have an eye mainly to coverage. First I need a case where *CA* > *LM*, *Televation*, and *F21-Knew-Info-Confidential* and *F16-info-Reverse-Engineerable* can play this role. For *LM* > *RE* I chose *Space Aero* and use factors *F15-Unique-Product* and *F19-No-Security-Measures*. I now need *RE* > *QM*, for which I can have *Robinson* with factors *F19-No-Security-Measures* and *F26-Deception*. Finally for *RE* > *MW* I chose *Sheets* with *F19-No-Security-Measures* and *F18-Identical-Products*. This yields the third theory shown in Figure 4.5. Table 4.4 shows the six rules and the four value preferences. This theory also only uses primitive rules.

```
Theory Preferences :
  pref(<{F15}, P>, <F19}, D>)
    <|<SpaceAero, {F1, F8, F15, F18, F19}, P>|>
  pref(<{F19}, D>, <{F18}, P>)
    <|<Sheets, {F18, F19, F27}, D>|>
  pref(<{F19}, D>, <{F26}, P>)
    <|<Robinson, {F1, F10, F18, F19, F26}, D>|>
  pref(<{F21}, P>, <{F16}, D>)
    <|<Televation, {F6, F10, F12, F15, F16, F18, F21}, P>|>
Theory Value Preferences :
  valpref({CA}, {LM})
  valpref({LM}, {RE})
  valpref({RE}, {QM})
  valpref({RE}, {MW})
```

Figure 4.5: Rule and Value Preferences from the "Value Driven" theory.

Table 4.4: The rules, rule preferences and value preferences for the "Value Driven" theory.

Rules	Rule Preference	Value Preference
(1) F15→P	(1)>(4)	LM>RE
(2) F16→D	(4)>(3)	RE>MW
(3) F18→P	(4)>(6)	RE>QM
(4) F19→D	(5)>(2)	CAS>LM
(5) F21→P		
(6) F26→P		

Note that I have produced three rather different theories. The value preferences are considerably different - Theory 2 recognises only three rather than five values - and Theory 1 places more stress on *RE* than does Theory 3.

4.5.5 Theory completion

In constructing the theory, typically not all the rule preferences determined by the value preferences will be explicitly included. Since it is part of the philosophy of [10] that the impact of factors can be derived from the values to which they relate, I can complete the theory by adding the remaining rule preferences entailed by the value preferences of the theory. Now, when the theory is executed, the value preferences are used to create these additional rule preferences which are then used to rank all the factors present in the theory. In this way the user does not need to explicitly identify and add all the rule preferences entailed by the value preferences, because is can be done for them.

For example, for the "Value Driven" Theory given above in Figure 4.5, the value preference of *CA>LM* ranks all the *Confidentiality Agreement* factors used above all the *Legitimate Means* factors used.

4.6 Methods of Factor Comparison

The initial method of code generation caused the first rule to match for each case to fire and give the case the outcome that the rule promotes. Thus I are embodying the assumption that the most preferred rule will govern the case by itself. This assumption will be relaxed in later experiments.

Even so, there are a number of ways in which I can form the rules from the theory. In the experiments I considered four different methods.

The first two methods distinguish between whether I choose to consider the absence of factors as well as their presence. The intention here is to explore Q4 of our research questions.

The last two methods address the issue of whether it is possible to order factors within a value (to explore Q5) in advance according to some general principle. I suggest two candidate possibilities.

4.6.1 Best factor

This method of code generation simply sorts the rules present in the theory according to the value order and rewrites the rules as Prolog clauses. Effectively this method weights a set of values according to its most highly weighted value, unless a comparison between particular set of values has been treated explicitly. Where a case contains the same value as the most highly rated for both the pro-plaintiff and pro-defendant sets, the outcome is determined by preferences between factors related to the same value, which makes the ordering of factors within a value significant. Much of the program refinement process, if this method is used, may be seen as tuning this ordering of factors with common values.

4.6.2 Best non-shared factor

A second possibility, following [28], is to discount values found in both the pro-plaintiff and pro-defendant sets, giving the set importance according to its most significant value not in the other set. This cancellation method creates rules of the form *f1 and not f2 and not f3... and not fn* where f2..fn are the factors which relate to the same value as f1 but favour the opposite side. For example, factors A, B and C promote some value V1 and A is a pro-plaintiff factors and B and C are pro-defendant factors. Table 4.5 shows the Prolog rule constructed from the primitive rules. The first Prolog rule will only fire if factor A is present in the case but factors B and C are absent. The second Prolog rule will fire if factor B is present in the case but factor A is absent. Because factors B and C are both pro-defendant factors, factor C is not included in the second Prolog rule.

Table 4.5: Example of best non-shared rules

Theory Rule	*Prolog Rule*
A→P	(X,p):-factor(X,a) and not factor(X,b) and not factor(X,c).
B→D	(X,d):-factor(X,b) and not factor(X,a).
C→D	(X,d):-factor(X,c) and not factor(X,a).

4.6.3 Exceptions

The exception method creates more rule preferences by ordering the factors within each value. The idea here is that more importance should be placed on exceptions than defaults. Whether a factor is considered an exception is determined by examining the factors that relate to each value, and considering whether pro-plaintiff or pro-defendant factors predominate. The factors promoting the less common outcome are taken to be exceptions and these are preferred to the factors promoting the more normal outcome for the value. Again using factors A, B and C, because A is outnumbered it is considered to the exception factor. CATE therefore adds two extra rule preferences into the theory: preferring A over B and A over C. The theory is now executed with these extra rule preferences.

4.6.4 CATO

The fourth method takes information from the CATO system. The CATO method also sorts the factors within each value. The factor hierarchy from [1] contains thin and thick arcs from the factors to the abstract factors and these are used to represent whether the factor is strong or weak. Strong factors are those with only strong/thick arcs from the factor to the abstract factor whereas weak factors have at least one weak arc. The strong factors within each value are preferred to the weak factors and so CATE adds extra rule preferences to the theory with all the combinations of a strong factor being preferred to a weak factor.

4.6.5 Experiments

In the experiments, all of these methods were applied to the theories produced by the three methods of theory construction. This was intended to cast light on Q1, by determining whether there were significant differences in the performance of the theories produced by the different methods.

Additionally, in each case, the theory was augmented by including all the factors as well as those used in the initial theory construction. The constructed theories will typically contain only a subset of the factors available from the background. Since the impact of factors can be derived from the values to which they relate, the theory can be extended by adding the remaining factors available from the background and ranking them according to the value preferences of the theory. The priority assigned to the rules containing these additional factors is determined solely by the preferences between their values in the theory. If the performance does not degrade, this can be taken as an acceptable way of establishing priorities between factors not explicitly considered, providing an answer to Q3.

4.7 Factor Comparison Results

The main results of these initial experiments are depicted in Tables 4.6 and 4.7 (for more detailed results see appendix B). *Chosen* is for the code using only the factors explicitly included and *All* is for the code which includes all the factors available from the background. The cells show the number of cases not correctly classified by the theory, including abstentions as well as misclassifications. The number in brackets show the failures to classify before refinement, for those cases where refinement was used.

Table 4.6: Results for the 14 Group 1 Cases

	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Best Factor	1 (3)	0 (3)	0	1 (4)	1	0 (1)
Cancellation	0 (1)	0 (1)	0	1	0	0
Exceptions	1 (3)	0	0	1	1	0
CATO	1 (3)	0 (1)	0	1 (3)	1	0 (1)

Table 4.7: Results for the 18 Group 2 Cases

	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Best Factor	3 (11)	4 (11)	5	6 (9)	6	4 (5)
Cancellation	8 (8)	5 (4)	4	6	5	3
Exceptions	3 (11)	5	5	7	6	4
CATO	3 (11)	6 (3)	5	6 (7)	6	4 (4)

From these results we can answer some of our original questions.

Q1: *How should we select cases and extract rules for inclusion in the theory?*

With regard to the Group 1 cases, it can be see that all three methods can produce theories which explain the body of cases reasonably well, and especially the theories can be refined by ordering factors within a value. Note also that for theories 1 and 3, performance improves if more factors are included. This is not true, however, of theory 2, suggesting that this method tends to overfit the data. None of the theories generalise particularly well, failing to give the correct decision in at least three, and typically more of the eighteen Group 2 cases. The cases misclassified varied somewhat for the different theories.

Q2: *Should we be inclusive or exclusive with regard to factors?*
Theories 1 and 3 actually improve their performance on the Group 1 cases if I include all the available factors. For the group 2 cases, abstentions are eliminated, often with the correct decision, except for the overfitting Theory 2. For Theories 1 and 3, I would recommend using all available factors.

Q3: *Is there evidence to suggest that values can be used to determine the relative importance of factors?*

The benefits of including all available factors, and the fact that these new factors are able to make a positive contribution using the value preferences determined by the other factors sharing their values, offers evidence that values are significant in accounting for the importance to be placed on factors.

Q4: *How should sets of values be compared?*
The evidence is by no means clear cut here. Theories 1 and 3 perform better with the best non-shared method on the training set, but this improvement does not generalise to the Group 2 cases. An alternative view is that comparison should use weighted factors, and this will be explored in the next set of experiments.

Q5: *Is it possible to use a general principle to pre-order factors relating to the same value?*

This question is addressed by the exception and CATO methods. The key test, because involving the most factors to order, is when the complete set of factors is used. These methods perform about the same, and perform the same as when no pre-ordering is used. Note, however, that the non pre-ordered results are for the refined theory: before refinement both theories 1 and 2 made more errors. The suggestion here, therefore, is that pre-ordering works to a certain extent, and can reduce the need for refinement - which amounts to a manual ordering of factors within a value. It does not, however, improve the capacity to generalise.

I next performed experiments involving the weighting of factors. In these experiments instead of the decision being determined by a single rule, all applicable rules are able to contribute to the decision. These experiments will address Q6 and Q7 of the original questions in section 4.4.

4.8 Weighting Methods

In order to consider all the values in a set I assigned its members a weight relative to the importance of the value to which they relate. Preferences are now reflected by weights rather than by rule order, allowing for the possibility that several weak factors may collectively outweigh a strong factor. There are two possibilities here, according to whether I count the values represented once, or increase the weight according to how many factors representing each of the values are present (that is, I use "bags" of values rather than "sets"). To explore this possibility I must assign weights to values.

The numeric method assigns weights to each value and hence to each factor within the value. The weight for each value is decided by the value preferences, a value which is preferred being given a larger weight than the value to which it is preferred. The weight is positive for a plaintiff factor and negative for a defendant factor.

For Theory 3 the value preferences are *CA>LM>RE>(MW, QM)*. The least preferred values of *MW* and *QM* are given the weight of 0.1. The next value of *RE* is given the weight of double the previous weight plus 0.1 which is 0.3. The next value of *LM* is again given the weight of double the previous weight plus 0.1 which is 0.7. Finally the value of *CA* is given the weight of double the previous weight plus 0.1 which is 1.5.

Assigning the weights for Theory 3 is easy because the value preference ordering is very simple. Assigning the weights for a more complicated set of value preferences is more difficult because the weights must still ensure that the individual value preferences work. For Theory 1 possible value weights are: (*MW, QM*) = 0.1, *RE* = 0.3 and (*LM, CA*) = 0.7. Looking at the value preferences for Theory 1 there are several complicated ones. For the value preference of *(MW, CA) > RE* I obtain a joint weight of 0.8 for *(MW, CA)* which is larger than the weight of 0.3 for the *RE* value. For the value preference of *RE > (MW, QM)* I obtain a joint weight of 0.2 for (*MW, QM*) which is less than the weight of 0.3 for *RE*.

Instead of relying on the order of the execution of the rules to enforce priorities, I now consider every factor present in order to determine the relative strength of the plaintiff's and defendant's cases. The size of the numbers produced may be an indicator of confidence in the predicted decision, but I lay no stress on this here.

I adopted four methods of accumulating weights.

4.8.1 Value weights

Values are assigned weights in accordance with the value preferences determined by the theory. The weights for each value present as a pro-plaintiff factor are summed as are the weights for each value present as a pro-defendant factor. The latter is then subtracted from the former to give the outcome for the case. A positive weight is pro-plaintiff and a negative weight is pro-defendant. This method considers sets of values. For the weights given for Theory 3 in section 4.8 a case with only pro-plaintiff factors promoting the value of Confidentiality Agreement *(CA)* will be given a weight of 1.5 no matter how many factors are present in the case. If the cases has pro-plaintiff factors and pro-defendant factors promoting *CA* it will have a weight of 0 because the value is cancelled out by its factors.

4.8.2 Factor weights

Factors are assigned weights according to the preferences given to the value they represent. The weights for each pro-plaintiff factor present are summed as are the weights for each pro-defendant factor. The latter is then subtracted from the former to give the outcome for the case. A positive number is pro-plaintiff and a negative number is pro-defendant. This method considers bags of values. For the weights given for Theory 3 each factor in the case that promotes *CA* will be given a weight of 1.5. The weight will be positive for a pro-plaintiff factor and negative for a pro-defendant factor and these weights are summed to give a weight for the case. A case with two pro-plaintiff factors promoting *CA* will have a weight of 3.0 whereas if the case also had a pro-defendant factor the case weight would be 1.5.

4.8.3 Exceptions

Exceptions, as defined in section 4.6.3, are considered strong factors, and the weights given to these factors are a multiple of the weight from the value of the factor. In the following experiments the factors which are exceptions have their weight multiplied by 10. These weights are then used as in the factor weights method. For the factors promoting the value of Confidentiality Agreement there are three pro-plaintiff factors (*F4-Agreed not to Disclose, F13-Non Competition Agreement* and *F21-Knew Information Confidential*) and two pro-defendant factors (*F5-Agreement Not Specific* and

F23-Waiver of Confidentiality). This means that *F5* and *F23* are considered as exception and will be given a weight of 15 (1.5x10) which will be negative because they are pro-defendant factors whereas the normal factors of *F4, F13* and *F21* will be given the weight of only 1.5 (using the weights defined for Theory 3).

4.8.4 IBP

In [15] a predictive program, IBP, based on CATO is described. In IBP factors are of three types: Knock-Out Factors, which are typically sufficient to determine the outcome on their own; Weak Factors, which typically have no significant impact on the outcome, but are important for contextualising the case, and Normal Factors, which have an influence, but not a determinant influence. I used this classification in our method by multiplying (by 10) the weight from the value for the knock-out Factors and dividing (by 10) the weight from the value for the weak Factors. Table 4.8 shows all of the factors sorted into their values and labelled with whether they are knock-out, normal or weak factors. In the *Reasonable Efforts* value pro-plaintiff factors *F6* and *F12* are normal factors and will be given a weight of 0.3. Pro-defendant factors *F1* and *F10* are weak factors and are given a negative weight of 0.03. Finally pro-defendant factors *F19* and *F27* are knock-out factors and are given a negative weight of 3.0.

4.9 Weighting Results

Results are summarised in Tables 4.9 and 4.10, which shows the number of cases, for Groups 1 and 2 respectively, misclassified by the theories. The full results, with the numerical outcome for each case are given in appendix B.

From this I can again see the problems with over fitting in Theory 2. For Theory 1 I can see the benefits of including all factors: this is not, however, a benefit in Theory 3. In comparison with the results of using unweighted factors,I have a clear improvement: suggesting in answer to Q4 that all the elements of a set do need to be considered. The exception method of identifying string factors is outperformed by IBP, indicating that expert knowledge derived from an analysis of cases is superior to the simple principle. No marked differences between counting all the factors relating to a value rather than simply all the values was found.

Some observations on the mistakes relating to particular cases may be made. Only

Table 4.8: Weak, normal and knock-out factors

Confidentiality Agreement	*Type*
F4 Agreed Not To Disclose (p)	Normal
F5 Agreement Not Specific (d)	Normal
F13 Non Competition Agreement (p)	Normal
F21 Knew Info Confidential (p)	Normal
F23 Waiver Of Confidentiality (d)	Normal
Reasonable Efforts	*Type*
F1 Disclosure In Negotiations (d)	Weak
F6 Security Measures (p)	Normal
F10 Secrets Disclosed Outsiders (d)	Weak
F12 Outsider Disclosures Restricted (p)	Normal
F19 No Security Measures (d)	KO
F27 Disclosure In Public Forum (d)	KO
Legitimate Means	*Type*
F3 Employee Sole Developer (d)	Normal
F11 Vertical Knowledge (d)	Normal
F15 Unique Product (p)	Normal
F16 Info Reverse Engineerable (d)	Weak
F17 Info Independently Generated (d)	Normal
F20 Info Known To Competitors (d)	KO
F24 Info Obtainable Elsewhere (d)	Normal
F25 Info Reverse Engineered (d)	Normal
Questionable Means	*Type*
F2 Bribe Employee (p)	Normal
F7 Brought Tools (p)	Normal
F14 Restricted Materials Used (p)	Normal
F22 Invasive Techniques (p)	Normal
F26 Deception (p)	KO
Material Worth	*Type*
F8 Competitive Advantage (p)	KO
F18 Identical Products (p)	Normal

Table 4.9: Results for different methods of weighting factors for the Group 1 cases.

	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Weighted Values	1	0	0	1	1	0
Weighted Factors	1	0	0	1	1	0
Exceptions	1	0	0	1	1	0
IBP	1	1	1	1	1	1

Table 4.10: Results for different methods of weighting factors for the Group 2 cases. * indicates *Mineral Deposits* classified correctly

	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Weighted Values	4	1	5	5	3	3
Weighted Factors	4	3	2*	7	1	3
Exceptions	5	4	2*	5	2	4
IBP	3*	2	1*	2	0*	2

the results marked with a "*" in Tables 4.9 classified *Mineral Deposits* correctly. Our reading of the decision was that the Court placed considerable stress on a specific feature of this case which is not adequately represented by any of the factors used. [24] says "compare *Mineral deposits Ltd. v. Zigan*, 773 P. D2 606 (Colo. App. 1988) (reverse engineering not allowed when product loaned in confidence)", although Aleven's analysis [1] does not have any factors relating to confidentiality for *Mineral Deposits* in our background. We might therefore suggest that this case is not suitable for use with the existing factor background. Another case which is consistently misclassified is *Space Aero*, which is also a mistake in all IBP methods. This may be explained by the presence of two knock-out Factors in that case, one for each side. If we disregard these two cases, when using the complete set of factors Theory 1 performs perfectly for all methods except the exception method. Theory 3, however, misclassifies *Scientology* (albeit with a very small weight) unless IBP is used.

To return to the original questions:

Q6: *Is there evidence to suggest that factors promote values to different degrees?*

The results show the need to use knowledge of the domain to modify the value weights because using the IBP method gives better results than when just using exceptions and better results than when using unmodified value weights.

Q7: *Is there evidence to suggest that factors relating to a particular value have a* *cumulative effect?*

On the basis of these experiments, there appears to be little difference between accumulating weights from values, from all the factors, and from giving differential weights to factors within values. For Q7 and both theories 1 and 3 the same cases are misclassified whichever method is used. With regard to Q6, the experience of [15], however, which reports IBP, which does use differential weights, as significantly outperforming programs that do not, suggests that I should investigate this further. I have no evidence to deny this, and would need to run the experiment on a larger data set then I have available before coming to any firm conclusions.

4.10 Concluding Remarks

This chapter described CATE and how it can be used to construct theories. The example using the Wild Animal domain shows that theories can be constructed incrementally and if the theory does not give the desired result the reason why can be identified and corrected. The second part of the chapter described how to construct generic theories to explain large parts of the case background and used three different methods to construct the theories. Four methods for executing the theory based on factor comparisons were described as well as four methods for applying weights to the factors and accumulating the weights.

The work here has provided some answers as to how to construct theories and has showed that the amount of refinement is reduced if some way of pre-ordering the factors is used.

Although the experiments with weights were inconclusive, the methods used to assign weights were not really very principled. In the next chapter I will consider dimensions and explore whether they offer a better way of determining appropriate weights for the factors.

Chapter 5

Dimension Based Theory Construction

5.1 Explanation of Dimensions

The experiments described in Chapter 4 were all conducted on cases represented using factors. Factors were used in the CATO system [1] and represent a simplification of the original notion of dimensions proposed in HYPO [2]. A discussion of the differences between factors and dimensions can be found in [32], and a case for the importance of dimensions is made in [7].

Factors represent features of cases which can be inferred from the case facts, and which are either present or absent. If present, factors strengthen the case for one of the plaintiff or the defendant. Dimensions allow for a finer grained consideration. On the facts of a case a dimension may be applicable or inapplicable. An applicable dimension represents a range of possibilities, with a direction. One end of the range is a pro-plaintiff extreme, and points along the range represent positions which are increasingly less favourable to the plaintiff and more favourable to the defendant until the other end of the range, the pro-defendant extreme, is reached. The case facts determine the point on the range that applies in that case. Note that dimensions do not themselves favour either party, although a point on a dimension is more or less favourable to a party.

Dimensions may be converted to factors in a number of ways:

1. a dimension may have a cross over point at which it ceases to favour the plaintiff and starts to favour the defendant. This may be mapped into two factors, one pro-plaintiff and one pro-defendant.

2. it may be that one end of the dimension favours, say, the plaintiff, and becomes less favourable as we move away from an extreme, but then becomes inapplicable rather than favouring the defendant. In such cases the dimension maps to a single factor.

3. it may be that we wish to map the dimension into a number of factors of differing strength, and possibly favouring different parties.

Examples of all three can be found in the transition from HYPO to CATO. Examples are:

1. HYPO has a dimension *Security-Measures-Adopted* with a number of positions. CATO has a pro-plaintiff factor *F6-Security-Measures* and a pro-defendant factor *F19-No-Security-Measures*. Only one of *F6-Security-Measures* and *F19-No-Security-Measures* can be present in a CATO case. *F19-No-Security-Measures* represents the pro-defendant extreme and *F6-Security-Measures* embraces all the pro-plaintiff points from minimal measures, access to premises controlled, restrictions on entry by visitors, restrictions on entry by employees, product marked confidential, restrictions on hardcopy release, and employee nondisclosure agreements.

2. HYPO has *Brought-Tools*. CATO has the pro-plaintiff factor *F7-Brought-Tools*, but no corresponding pro-defendant factor.

3. HYPO has *Secrets-Disclosed-Outsiders* in which the number of disclosures represents the points. CATO has a pro-defendant factor *F10-Secrets-Disclosed-Outsiders*, and also the stronger pro-defendant factor *F27-Disclosure-in-Public-Forum*.

Table 5.1 shows the HYPO dimensions and their type. Only two of the dimensions are true dimensions of type 3, all the others are either binary or singular. This lack of "real" dimensions in HYPO explains why I did not simply use the HYPO analysis.

Table 5.1: HYPO Dimensions and their type

Dimension Name	*Type of Dimension*
Competitive-Advantage-Gained	3
Vertical-Knowledge	1
Secrets-Voluntary-Disclosed	1
Disclosures-Subject-To-Restriction	1
Agreement-Supported-By-Consideration	1
Common-Employee-Paid-To-Change-Employers	1
Exists-Express-Noncompetition-Agreement	1
Common-Employee-Transferred-Product-Tools	1
Nondisclosure-Agreement-Re-Defendant-Access	2
Common-Employee-Sole-Developer	2
Nondisclosure-Agreement-Specific	2
Disclosure-In-Negotiations-With-Defendant	2
Security-Measures-Adopted	3

Thus a dimension can be seen as a collection of factors which all relate to a given issue. Case descriptions can either be constructed in terms of the factors present, or in terms of points on the applicable dimensions.

Factors in CATO also have different degrees of importance (represented by thick and thin lines in the Factor Hierarchy [1] which indicate that a factor may exclude some other factors, and in [15] as normal, weak and knock out factors indicating predictive power). If the factors *F10-Secrets-Disclosed-Outsiders* and *F27-Disclosure-in-Public-Forum* are compared, they are both defendant factors but *F10-Secrets-Disclosed-Outsiders* is weaker for the defendant because the plaintiff only disclosed the trade secret to some outsiders, whereas *F27-Disclosure-in-Public-Forum* is stronger for the defendant because if the plaintiff disclosed the trade secret in a public forum then it can hardly be considered a secret anymore. *F27-Disclosure-in-Public-Forum* is a knock-out factor in [15]. As indicated in (3) above, such factors can be thought of as points on the same dimension with *F27-Disclosure-in-Public-Forum* placed at the stronger part of the dimension and *F10-Secrets-Disclosed-Outsiders* placed at the weaker part of the dimension.

5.1.1 Relation to values

Values represent groups of factors which promote the same social value and dimensions are groups of factors with similar features in common. Therefore if the feature the dimension represents is also a social value then it means that all the factors promoting the social value (to different degrees) will all be present on the same dimension. This is not the only way to construct dimensions and in some domains it may be that the factors promoting a particular value may make more sense when they are divided between several dimensions.

This similarity of values to dimensions is the basis of the extension to values described by Bench-Capon and Sartor in [10], and we will follow their account and arrange our factors according to the extent to which they promote the value to which they relate. For example, *F10-Secrets-Disclosed-Outsiders* and *F27-Disclosure-in-Public-Forum* can be seen as points on the social value scale of taking *Reasonable Efforts*, with *F27-Disclosure-in-Public-Forum* being stronger than *F10-Secrets-Disclosed-Outsiders*. In making this move, however, I am departing considerably from the HYPO conception of dimension, and changing the focus from how the facts of a case are represented to a measure of the contribution to an issue made by a factor. One might therefore argue that I should not use the term "dimension". Since, however, the approach is inspired by the notion of dimension, and follows Bench-Capon and Sartor's attempt to accommodate the role of dimensions in their notion of a theory, I will still use the term "dimension" despite the differences with the HYPO conception. What I am doing is providing structure to the values, by identifying the different extents to which the factors promote their values. In [20] I used the term "Structured Values" rather than Dimension, to emphasise the differences with HYPO.

5.1.2 Comparison with IBP model

In Brüninghaus and Ashley's program IBP (Issue-Based Prediction) [15] the factors are grouped in five issues and the factors may be present in more than one issue. These issues are similar to my values although in my values, the factors can only be present in one value. (This limitation will be addressed when I consider complex structures later in this chapter) IBP also has factors of different strengths, knock-out, normal and weak which can be seen as corresponding to points on dimensions. However IBP does not explicitly order the factors. I can therefore see my approach as relating issues and dimensions, and using the dimensions notion to give a finer grained assessment of the importance of factors to particular issues.

5.1.3 As a justification of weights

Each dimension, like each value, can have a different weight because some dimensions might be considered more important than others. Also, because a dimension consists of a range of points of differing strengths, the weight can be varied to give each point a different proportion of the dimension weight. For example, the strong factor of *F27-Disclosure in a Public Forum* would be given a large proportion of the dimension weight and the weak factor of *F10-Disclosure in Negotiation* would be given a small proportion of the dimension weight.

In the experiments I wanted to explore three main issues:

- How should we map factors back into dimensions?
- Can we identify a plausible relationship between values and dimensions?
- Can we use the notion of dimensions to produce a principled means of assigning weights to factors?

The experiments performed with dimensions are described below.

5.2 Conversion of Factors to Dimensions

In this chapter I will use two types of Dimensions. On the Simple Dimensions a factor can only be present as a point on a single Dimension, whereas for the Complex Dimensions each factor can be present as a point on more than one Dimension.

5.3 Simple Dimensions

I began with a notion of Simple Dimensions, where each factor can only be present on a single dimension. These Simple Dimensions were created by using the value groupings described in Section 3.3.2. Figure 5.1 shows the grouping of the factors into the dimensions. (See Table 5.2 for factor names)

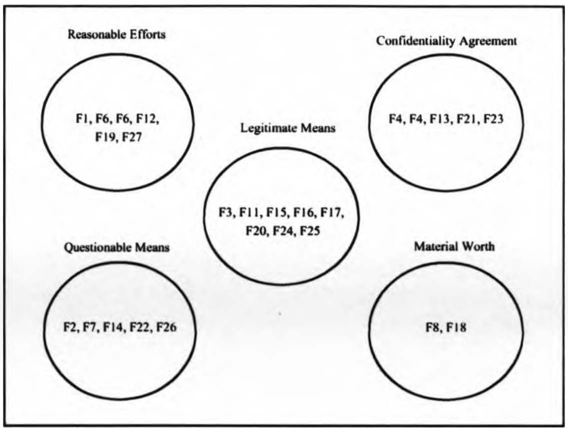

Figure 5.1: Factors divided into simple dimensions and their relationships.

The dimension is arranged with the strong pro-P factors first, running through the weaker pro-P factors, then changing to the weaker pro-D factors and finally the strong pro-D factors. The dimensions do not need to be arranged like this as they can have the defendant factors first; I have, however, chosen a consistent direction for ease of reading. If the dimension only consists of factors which promote one outcome then only this part of the dimension is used. Table 5.2 shows the factors present in each dimension sorted into the correct range for the dimension. The Type column indicates the type of factor as defined in IBP.

This table clearly indicates the consistency of my ordering with the strengths assigned in IBP, although I have made some decisions, eg. making *F6-Security-Measures* stronger for the plaintiff than *F12-Outsider-Disclosures-Restricted.*

Because I want the stronger factors to subsume the weaker factors to avoid double counting, the case descriptions have to reflect this. The weaker factors are removed from the description, leaving only the strongest pro-plaintiff factor and the strongest pro-defendant factor on each dimension. Where there are Plaintiff and Defendant factors from the same dimension in the case description then the strongest Plaintiff factor and strongest Defendant factor remain while the weaker points are removed.

For example, in the Arco case, factors *F16-Info-Reverse-Engineerable* and *F20-Info-Known-To-Competitors* are both on the *Legitimate Means* dimension and because *F20-Info-Known-To-Competitors* is a stronger factor, it subsumes the weaker *F16-Info-Reverse-Engineerable.* Figure 5.2 demonstrates this.

Table 5.2: Simple Dimensions

Confidentiality Agreement	*Type*
F13 Non Competition Agreement (p)	Normal
F4 Agreed Not To Disclose (p)	Normal
F21 Knew Info Confidential (p)	Normal
F5 Agreement Not Specific (d)	Normal
F23 Waiver Of Confidentiality (d)	Normal
Reasonable Efforts	*Type*
F6 Security Measures (p)	Normal
F12 Outsider Disclosures Restricted (p)	Normal
F1 Disclosure In Negotiations (d)	Weak
F10 Secrets Disclosed Outsiders (d)	Weak
F27 Disclosure In Public Forum (d)	KO
F19 No Security Measures (d)	KO
Legitimate Means	*Type*
F15 Unique Product (p)	Normal
F16 Info Reverse Engineerable (d)	Weak
F25 Info Reverse Engineered (d)	Normal
F3 Employee Sole Developer (d)	Normal
F11 Vertical Knowledge (d)	Normal
F24 Info Obtainable Elsewhere (d)	Normal
F17 Info Independently Generated (d)	Normal
F20 Info Known To Competitors (d)	KO
Questionable Means	*Type*
F26 Deception (p)	KO
F22 Invasive Techniques (p)	Normal
F2 Bribe Employee (p)	Normal
F14 Restricted Materials Used (p)	Normal
F7 Brought Tools (p)	Normal
Material Worth	*Type*
F8 Competitive Advantage (p)	KO
F18 Identical Products (p)	Normal

```
<Arco, {F10, F16, F20}, D>
            ↓
  <Arco, {F10, F20}, D>
```

Figure 5.2: Modification of the Arco Case.

In the Boeing case, factors *F6-Security-Measures* and *F12-Outsider-Disclosures-Restricted* are Plaintiff factors on the *Reasonable Efforts* dimension and factors *F1-Disclosure-in-Negotiation* and *F10-Secrets-Disclosed-Outsiders* are Defendant factors. *F6-Security-Measures* subsumes *F12-Outsider-Disclosures-Restricted*, *F10-Secrets-Disclosed-Outsiders* subsumes *F1-Disclosure-in-Negotiation* and so both of these factors remain in the case description. Also factors *F4-Agreed-Not-To-Disclose* and *F21-Knew-Info-Confidential* are on the *Confidentiality Agreement* dimension and *F4-Agreed-Not-To-Disclose* subsumes *F21-Knew-Info-Confidential*. *F14-Restricted-Materials-Used* is the only factor present from the *Questionable Means* dimension and so remains. Figure 5.3 demonstrates this.

```
<Boeing, {F1, F4, F6, F10, F12, F14, F21}, P>
                     ↓
      <Boeing, {F4, F6, F10, F14}, P>
```

Figure 5.3: Modification of the Boeing Case.

Table 5.3 shows the new case descriptions with the factors separated into Plaintiff and Defendant factors.

5.4 Complex Dimensions

When I use Complex Dimensions, these will relate factors which may promote several of my original values. Also factors may appear in several complex Dimensions. This means that the complex Dimensions will need to represent new values.

I choose four such new values. *Honouring Agreements* is desirable since if all dealings were regulated by properly drafted agreements, there would be no conflicts for the courts to resolve. A second Dimension can relate to interests in security, to enforce the notion that a person with a secret should take *Reasonable Efforts* to maintain it. Thirdly I want to avoid litigation where possible, and if information is known generally,

Table 5.3: Simple Dimension Based Cases Used in CATE

Case	*Pro-P factors*	*Pro-D factors*	*Outcome*
Group 1			
Arco		F10, F20	D
Boeing	F4, F6, F14	F10	P
Bryce	F4, F6, F18	F1	P
CollegeWatercolour	F15, F26	F1	P
Den-Tal-Ez	F4, F6, F26	F1	P
Ecologix	F21	F19, F23	D
Emery	F18, F21	F10	P
Ferranti	F2	F19, F20	D
Robinson	F18, F26	F19	D
Sandlin		F16, F19	D
Sheets	F18	F19	D
Space Aero	F8, F15	F19	P
Televation	F6, F15, F18, F21	F10, F16	P
Yokana	F7	F16, F27	D
Group 2			
CMI	F4, F6	F20, F27	D
DigitalDevelopment	F6, F8, F15, F21	F1	P
FMC	F4, F6, F7	F10, F11	P
Forrest	F6, F15, F21	F1	P
Goldberg	F21	F27	P
KG	F6, F14, F16, F18, F21	F25	P
Laser	F6, F21	F10	P
Lewis	F8, F21	F1	P
MBL	F6, F13	F5, F10, F20	D
Mason	F6, F15, F21	F1, F16	P
MineralDeposits	F18	F1, F25	P
NationalInstrument	F18, F21	F1	P
NationalRejectors	F7, F15, F18	F16, F19	D
Reinforced	F4, F6, F8, F15	F1	P
Scientology	F4, F6	F10, F20	D
Technicon	F6, F14, F21	F10, F25	P
Trandes	F4, F6	F10	P
Valco-Cincinnati	F6, F15, F21	F10	P

there should be no case to answer and hence *Less Litigation*. Finally I want to promote *Fair Competition*, and this gives rise to two Dimensions, one relating to questionable means taken to obtain the secret, and one to fair methods having been used to develop the product. I thus have five complex Dimensions with which to organise the factors.

With Complex Dimensions, the factors can occur as points on several of these dimensions. Factors were assigned to these dimensions by analysing the factor descriptions from [1]. Many factors seem to have several different characteristics and so can be placed on different dimensions.

F10 SecretsDisclosedOutsiders (d)
Description: Plaintiff disclosed its product information to outsiders. This factor shows that the plaintiff's information was known in the industry or available from sources outside plaintiff's business. Also, it shows that plaintiff showed a lack of interest in maintaining the secrecy of its information.
The factor applies if: Plaintiff disclosed its product information for example to licensees, customers, suppliers, subcontractors, etc.
The factor does not apply if: Plaintiff published the information in a public forum. (In that situation, F27 applies.) all we know is that plaintiff marketed a product from which the information could be ascertained by reverse engineering.

Figure 5.4: Factor Description of *F10-Secrets-Disclosed-Outsiders* from [1].

Figure 5.4 shows the description for *F10-Secrets-Disclosed-Outsiders* as given in [1] and Appendix A. It has a *Security* characteristic, namely that the plaintiff told outsiders and so failed to show concern for his secret, an *Information Known* characteristic because people outside the plaintiff know the information and a *Fair Method* characteristic because the defendant can obtain the information from the outsiders.

Figure 5.5 shows how the factors can be grouped into Complex Dimensions and also how the dimensions overlap and relate to each other.

The dimensions are again arranged with strong pro-P factors first, running through the normal and weak pro-P factors, then changing to the weak pro-D factors, then the normal pro-D factors and finally the strong Pro-D factors. If the dimension only consists of factors which promote one outcome then again only this part of the dimension is used.

The factors for each dimension are arranged into their order of strength with the strong factors at the end of the dimension and the weaker factors towards the centre. The relative ordering of the factors stays the same on all the dimensions, so that if a factor is stronger than another factor on one dimension then it will be stronger on all the

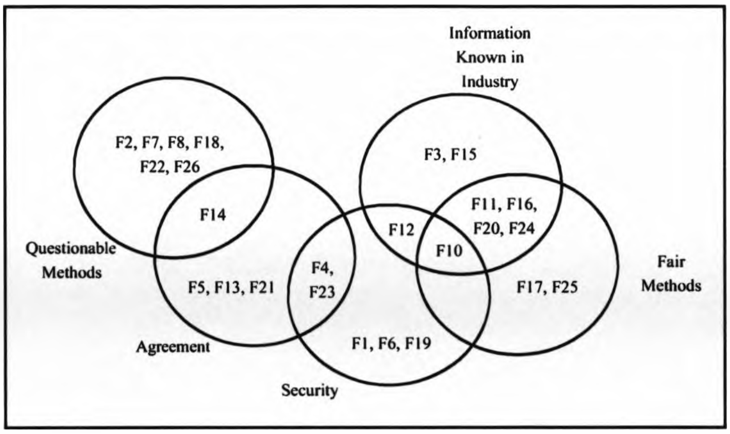

Figure 5.5: Factors divided into complex dimensions and their relationships.

other dimensions on which they both occur. Factors *F11-Vertical-Knowledge* and *F16-Info-Reverse-Engineerable* occur on two dimensions and *F11-Vertical-Knowledge* is stronger than *F16-Info-Reverse-Engineerable*. In the *Information Known in Industry* dimension they are next to each other, but in the *Fair Methods* dimension they are separated by the inclusion of *F25-Info-Reverse-Engineered.* Table 5.4 shows the factors present in each dimension sorted into the correct range for the dimension, and the type in [15]. Again the ordering is consistent with the partial order implied by IBP.

The factor based case descriptions need to be modified again to reflect these Complex Dimensions. First, the factors which are always together on all the dimensions are compared and the weaker factors subsumed by the stronger factors. Second, the remaining factors are transformed into all the possible dimension points, because *F10-Secrets-Disclosed-Outsiders* is on three dimensions it is replaced by three dimension points, namely *F10sec, F10inf* and *F10fme*. The dimension point is labelled with the factor number plus three extra letters to reflect which dimension it is located on. Third, the points from each dimension are compared and the strong factors subsume the weaker factors and remain in the case description.

In the Arco case *F16-Info-Reverse-Engineerable* and *F20-Info-Known-To-Competitors* are always together on all the dimensions in which they feature because they share the same characteristics. Because *F20-Info-Known-To-Competitors* is stronger it subsumes *F16-Info-Reverse-Engineerable*, leaving *F10-Secrets-Disclosed-Outsiders* and *F20-Info-Known-To-Competitors*. Next they are replaced by the relevant

Table 5.4: Complex Dimensions

Interests in Security	*Type*	*Info Known in Industry*	*Type*
F6 Security Measures (p)	Normal	F15 Unique Product (p)	Normal
F4 Agreed Not To Disclose (p)	Normal	F12 Outsider Disclosures Restricted (p)	Normal
F12 Outsider Disclosures Restricted (p)	Normal	F10 Secrets Disclosed Outsiders (d)	Weak
F1 Disclosure In Negotiations (d)	Weak	F16 Info Reverse Engineerable (d)	Weak
F10 Secrets Disclosed Outsiders (d)	Weak	F11 Vertical Knowledge (d)	Normal
F23 Waiver Of Confidentiality (d)	Normal	F24 Info Obtainable Elsewhere (d)	Normal
F27 Disclosure In Public Forum (d)	KO	F3 Employee Sole Developer (d)	Normal
F19 No Security Measures (d)	KO	F20 Info Known To Competitors (d)	KO
		F27 Disclosure In Public Forum (d)	KO
Questionable Methods	*Type*		
F8 Competitive Advantage (p)	KO	*Fair Methods*	*Type*
F26 Deception (p)	KO	F10 Secrets Disclosed Outsiders (d)	Weak
F22 Invasive Techniques (p)	Normal	F16 Info Reverse Engineerable (d)	Weak
F2 Bribe Employee (p)	Normal	F25 Info Reverse Engineered (d)	Normal
F14 Restricted Materials Used (p)	Normal	F11 Vertical Knowledge (d)	Normal
F7 Brought Tools (p)	Normal	F24 Info Obtainable Elsewhere (d)	Normal
F18 Identical Products (p)	Normal	F17 Info Independently Generated (d)	Normal
		F20 Info Known To Competitors (d)	KO
Agreements	*Type*	F27 Disclosure In Public Forum (d)	KO
F13 Non Competition Agreement (p)	Normal		
F4 Agreed Not To Disclose (p)	Normal		
F21 Knew Info Confidential (p)	Normal		
F14 Restricted Materials Used (p)	Normal		
F5 Agreement Not Specific (d)	Normal		
F23 Waiver Of Confidentiality (d)	Normal		

dimension points. *F10-Secrets-Disclosed-Outsiders* is replaced by three points and *F20-Info-Known-To-Competitors* by two points. Finally the stronger *F20-Info-Known-To-Competitors* points subsume the weaker *F10-Secrets-Disclosed-Outsiders* points on both the *Information Known in Industry* and *Fair Methods* dimensions so we are left with the strongest factors from each dimension. Figure 5.6 shows this process.

```
<Arco, {F10, F16, F20}, D>
↓
<Arco, {F10, F20}, D>
↓
<Arco, {F10sec, F10inf, F10fme, F20inf, F20fme), D>
↓
<Arco, {F10sec, F20inf, F20fme}, D>
```

Figure 5.6: Modification of the Arco Case.

In the Boeing case, none of the factors are always together so all the factors are replaced by their relevant dimension points. Now the weaker factors can be subsumed by the stronger factors. Figure 5.7 shows the process for the Boeing case.

```
<Boeing, {F1, F4, F6, F10, F12, F14, F21}, P>
↓
<Boeing, {F1, F4, F6, F10, F12, F14, F21}, P>
↓
<Boeing, {F1sec, F4agr, F4sec, F6sec, F10sec, F10inf,
F10fme, F12sec, F12inf, F14qme, F14agr, F21agr}, P>
↓
<Boeing, {F4agr, F6sec, F10sec, F10inf, F10fme,
F12inf, F14qme}, P>
```

Figure 5.7: Modification of the Boeing Case.

Tables 5.5 and 5.6 show the new case descriptions with the factors and values separated into Plaintiff and Defendant.

Table 5.5: Group 1 Complex Dimension Based Cases Used in CATE

Case	*Pro-P factors*	*Pro-D factors*	*Outcome*
Arco		F10sec, F20inf, F20fme	D
Boeing	F4agr, F6sec, F12inf, F14qme	F10fme, F10inf, F10sec	P
Bryce	F4agr, F6sec, F18qme	F1sec	P
College Watercolour	F15inf, F26qme	F1sec	P
Den-Tal-Ez	F4agr, F6sec, F26qme	F1sec	P
Ecologix	F21agr	F19sec, F23agr	D
Emery	F18qme, F21agr	F10sec, F10inf, F10fme	P
Ferranti	F2qme	F19sec, F27inf, F27fme	D
Robinson	F26qme	F10inf, F10fme, F19sec	D
Sandlin		F19sec, F27inf, F27fme	D
Sheets	F18qme	F19sec, F27inf, F27fme	D
Space Aero	F8qme, F15inf	F19sec	P
Televation	F6sec, F15inf, F18qme, F21agr	F10sec, F16inf, F16fme	P
Yokana	F7qme	F27sec, F27inf, F27fme	D

5.5 Constructing Dimension theories

Due to the poor results for the simple method (Theory 2) in the experiments based on factors described in Chapter 4, this method was not used in the Dimension experiments leaving only two approaches to be studied.

5.5.1 The "Safe" Method

The first of these is the safe theory described earlier in Section 4.5.2. For the simple dimensions, the same cases can be used as in Section 4.5.2, although the actual theory will be different due to the changes in the case description. The Plaintiff cases used are *Emery* and *College Watercolour* and the Defendant cases are *Robinson* and *Sheets*. Representing the rules from these four cases in the manner of Prakken and Sartor in [30] yields the rule and value preferences shown in Figure 5.8.

For the complex dimensions it is more difficult to choose cases. I wanted the simplest value preferences available and this means not producing rule preferences with the same value preferred over itself because such value preferences are incoherent. This highlights a problem with using the Prakken and Sartor approach with dimensions. The case description no longer expresses a definite rule, as we cannot tell which aspect of

Table 5.6: Group 2 Complex Dimension Based Cases Used in CATE

Case	*Pro-P factors*	*Pro-D factors*	*Outcome*
CMI	F4agr, F6sec	F27fme, F27inf, F27sec	D
Digital Development	F6sec, F8qme, F15inf, F21agr	F1sec	P
FMC	F4agr, F6sec, F7qme, F12inf	F11inf, F11fme, F10sec	P
Forrest	F6sec, F15inf, F21agr	F1sec	P
Goldberg	F21agr	F27fme, F27inf, F27sec	P
KG	F6sec, F14qme, F15inf F21agr	F16inf, F25fme	P
Laser	F6sec, F12inf, F21agr	F10fme, F10inf, F10sec	P
Lewis	F8qme, F21agr	F1sec	P
MBL	F6sec, F13agr	F5agr, F10sec, F20fme, F20inf	D
Mason	F6sec, F15inf, F21agr	F1sec, F16inf, F16fme	P
Mineral Deposits	F18qme	F1sec, F16inf, F25fme	P
National Instrument	F18qme, F21agr	F1sec	P
National Rejectors	F7qme, F15inf	F19sec, F27inf, F27fme	D
Reinforced	F4agr, F6sec, F8qme, F15inf	F1sec	P
Scientology	F4agr, F6sec, F12inf	F10sec, F20inf, F20fme	D
Technicon	F6sec, F12inf, F14qme, F21agr	F10sec, F16inf, F25fme	P
Trandes	F4agr, F6sec, F12inf	F10fme, F10inf, F10sec	P
Valco-Cincinnati	F6sec, F15inf, F21agr	F10fme, F10inf, F10sec	P

```
Theory Preferences :
  pref(<{F15, F26}, P>, <{F1}, D>)
    <|<CollegeWatercolour, {F1, F15, F26}, P>|>
  pref(<{F18, F21}, P>, <{F10}, D>)
    <|<Emery, {F10, F18, F21}, P>|>
  pref(<{F19}, D>, <{F18, F26}, P>)
    <|<Robinson, {F18, F19, F26}, D>|>
  pref(<{F19}, D>, <{F18}, P>)
    <|<Sheets, {F18, F19}, D>|>
Theory Value Preferences :
  valpref({CA, MW}, {RE})
  valpref({LM, QM}, {RE})
  valpref({RE}, {MW, QM})
  valpref({RE}, {MW})
```

Figure 5.8: Rule and Value Preferences for the "Safe" Method using the Simple Dimensions.

the factor was held to be of importance in the decision. This might be resolved by returning to the original text. This means I are limited to using just two Plaintiff cases, *College Watercolour* and *Space Aero*. Using these two cases yields the rule and value preferences shown in Figure 5.9.

```
Theory Preferences :
  pref(<{F15inf, F26qme}, P>, <{F1sec}, D>)
    <|<CollegeWatercolour, {F1sec, F15inf, F26qme}, P>|>
  pref(<{F15inf, F8qme}, P>, <F19sec}, D>)
    <|<SpaceAero, {F8qme, F15inf, F19sec}, P>|>
Theory Value Preferences :
  valpref({FC, LL}, {RE})
```

Figure 5.9: Rule and Value Preferences for the "Safe" Method using the Complex Dimensions.

5.5.2 The "Value Driven" Method

For Simple Dimensions, because they use the same values as were used with factors, the same cases can be used, yielding the same rule and value preferences. *Televation* is used to represent the value preference of CA>LM, *Space Aero* is used to represent the value preference of LM>RE, *Robinson* is used to represent the value preference of RE>QM and finally *Sheets* is used to represent the value preference of RE>MW. The rule and value preferences are given in Figure 5.10.

```
Theory Preferences :
  pref(<{F15}, P>, <{F19}, D>)
    <|<SpaceAero, {F8, F15, F19}, P>|>
  pref(<{F19}, D>, <{F18}, P>)
    <|<Sheets, {F18, F19}, D>|>
  pref(<{F19}, D>, <{F26}, P>)
    <|<Robinson, {F18, F19, F26}, D>|>
  pref(<{F21}, P>, <F16}, D>)
    <|<Televation, {F6, F10, F15, F16, F18, F21), P>|>
Theory Value Preferences :
  valpref({CA}, {LM})
  valpref({LM}, {RE})
  valpref({RE}, {MW})
  valpref({RE}, {QM})
```

Figure 5.10: Rule and Value Preferences for the "Value Driven" Method using the Simple Dimensions.

However, due the change in the values for the complex dimensions, the theory must be changed to reflect this. Let us suppose that the most highly rated value is *Honouring Agreements*, since if all dealings were regulated by properly drafted agreements, there would be no problem to decide. Let us rate the value of *Less Litigation* next; I want to stop frivolous court cases wasting time and money. I rate the value of *Reasonable Efforts* third, since people must take some steps to protect themselves. This leaves the value of *Fair Competition* last.

```
Theory Preferences :
  pref(<{F15inf}, P>, <{F1sec}, D>)
    <|<CollegeWatercolour, {F1sec, F15inf, F26qme}, P>|>
  pref({<F19sec}, D>, <{F26qme}, P>)
    <|<Robinson, {F10fme, F10inf, F19sec, F26qme}, D>|>
  pref({F21agr}, P>, <{F10inf}, D>)
    <|<Emery, {F10fme, F10inf, F10sec, F18qme, F21agr}, P>|>
Theory Value Preferences :
  valpref({HA}, {LL})
  valpref({LL}, {RE})
  valpref({RE}, {FC})
```

Figure 5.11: Rule and Value Preferences for the "Value Driven" Method using the Complex Dimensions.

Because there are only four values, only three cases are needed to represent the value preferences. For the value preference of HA>LL *Emery* is used with *F10inf* and *F21agr*. For the value preference of LL>RE *College Watercolour* is used with

F1sec and *F15inf.* Finally for RE>FC *Robinson* is used with *F19sec* and *F26qme*. These cases yield the rule and value preferences shown in Figure 5.11. Of course, this ordering may not be that of the jurisdictions which tried the cases.

5.6 Methods of Comparison

The four methods for comparing factors described in Chapter 4 are again used in our Dimension experiments. These are Best Factor, Best Non-Shared Factor, Exceptions and CATO.

5.7 Results

The full results of these experiments are described in Appendix C and the main results are depicted in Tables 5.7 and 5.8. The theories here use all the factors available in the background. The comparison is therefore with the "All" columns of Tables 4.6 and 4.7. The tables show the number of cases not correctly classified by the theory, including abstentions as well as misclassifications. The number in brackets show the failures to classify before refinement, for those cases where refinement was used.

Table 5.7: Results for the 14 Group 1 Cases

	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Best Factor	0 (3)	0 (1)	0 (5)	0 (1)
Cancellation	0 (1)	0	0 (1)	0
Exceptions	0	0	0 (4)	0
CATO	0 (1)	0	0 (2)	0 (1)

Table 5.8: Results for the 18 Group 2 Cases

	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Best Factor	5 (12)	5 (5)	4 (11)	13 (13)
Cancellation	5 (4)	3	3 (8)	3
Exceptions	5	4	4 (3)	4
CATO	3 (2)	4	4 (4)	4 (4)

As with factors it is possible, perhaps with some refinement, to produce theories to explain the training set of cases using either method of theory construction and any of the methods of comparison. When using Simple Dimensions, the dimensions method performs worse for Theory 1 than the corresponding methods using factors, except when weights are determined using the information from CATO. This suggests that the subsumption of factors has had some negative effect. Cases represented with Simple Dimensions typically contain fewer factors than when represented using factors directly. Since here weights are not used, it may be that removing the subsumed factor understates the contribution of that dimension. Theory 3 performs as well or better than Theory 1 except for the CATO comparison method, and performs comparably to when using factors rather than dimensions. When using Complex Dimensions, Theory 1 and Theory 3 perform equally well except for the Best Factor comparison method where Theory 3 gets very bad results. This might be explained because there is a mismatch between the finer grained representation and the broad brush comparison technique. For the remaining comparison methods, Complex Dimensions never perform worse than factors, and both Theories obtain better results that when using factors if the cancellation method is used. The improvements are, however, slight.

Remember, however, that we are here using unweighted factors, and the weighting implied by the position within a dimension is an important part of the rationale of using dimensions. We might therefore expect problems to arise where we have the possibility of cases with a point near the middle of a strongly valued dimension and at one extreme of a weakly valued dimension. We therefore conclude, that if no account of weighting is to be taken, there is little to be gained from thinking in terms of dimensions rather than values. We turn to experiments with weights in the next section.

5.8 Weighting Methods

The four methods of weighting factors used in Section 4.8 are again used with an additional method using the dimensions to adjust the weight given to the factor.

For this new method the dimension is divided into 20 slots, 10 plaintiff slots and 10 defendant slots and the factors contained on the dimension can be placed in any of the slots. Where they are placed depends on how strong the factor is. Knockout factors are placed at the ends of the dimension and receive the largest proportion of the weight. Weak factors are placed at the centre of the dimension and receive the smallest proportion of the weight, while normal factors are placed midway and receive intermediate weights. The factors can be moved along the dimension until the best position is found. Gaps are allowed, respecting the possibility that more factors could be introduced if finer grained points of discrimination are thought necessary. A factor in slot 10 receives one tenth of the weight, in slot 9 it receives two tenths and so on until slot 1 where it receives the whole weight. Table 5.9 shows the *Reasonable Efforts* dimension with its factors placed into their slots. *Reasonable Efforts* has a maximum weight of 0.3 to reflect its importance relative to the other dimensions. Here we are able to reflect that *F6-Security-Measures* and *F12-Outsider-Disclosures-Restricted* are normal factors, *F1-Disclosure-in-Negotiation* and *F10-Secrets-Disclosed-Outsiders* weak factors and *F19-No-Security-Measures* and *F27-Disclosure-in-Public-Forum* are knock-out factors, by positioning them in different parts of the range. The factor *F6-Security-Measures* receives eight tenths of the weight and as the weight given to *Reasonable Effort* is 0.3, *F6-Security-Measures* has a weight of 0.24. *F12-Outsider-Disclosures-Restricted* is slightly weaker and only receives a weight of seven tenths or 0.21. Were I to discover a knock-out plaintiff factor for this dimension it could be placed in Slots 1 or 2.

Table 5.9: Dimension weighting for the Reasonable Efforts Dimension

Plaintiff End		*Weight*	*Change Over Point*		*Weight*
Slot 1		0.3	Slot 10	F1	-0.03
Slot 2		0.27	Slot 9	F10	-0.06
Slot 3	F6	0.24	Slot 8		-0.09
Slot 4	F12	0.21	Slot 7		-0.12
Slot 5		0.18	Slot 6		-0.15
Slot 6		0.15	Slot 5		-0.18
Slot 7		0.12	Slot 4		-0.21
Slot 8		0.09	Slot 3		-0.24
Slot 9		0.06	Slot 2	F27	-0.27
Slot 10		0.03	Slot 1	F19	-0.3
Change Over Point			*Defendant End*		

5.9 Results

Results for the different methods of using weights are summarised in Tables 5.10 and 5.11. The full set of results are given in Appendix C. The cells show the number of cases misclassified by the theory using all the cases, separated for groups 1 and 2.

Table 5.10: Results for 14 Group 1 Cases

	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Weighted Values	0	0	0	0
Weighted Factors	0	0	0	0
Exceptions	0	0	1	1
IBP	1	1	0	1
Dimension	0	0	0	0

Table 5.11: Results for the 18 Group 2 Cases

	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Weighted Values	1	3	3	3
Weighted Factors	1	3	3	3
Exceptions	3	4	5	5
IBP	2	2	4	2
Dimension	2	5	2	5

For Simple Dimensions and Theory 1 the Weighted Values and Weighted Factors methods and the Dimension Weights method perform the best for the Group 1 cases. For the Group 2 cases all the methods perform worse with Dimension Weights being outperformed by Weighted Values and Factors. This may be due to the fact that the weights were adjusted for the Group 1 cases and need to moved slightly to correctly decide the Group 2 cases. When using Theory 3, the Dimension Weights method performs worst. With only a single misclassified case for Theory 1 (which is Mineral Deposits and is due to the single plaintiff factor promoting a weak value and being beaten by stronger defendant values), these represent the best performance from any of our experiments. For Complex Dimensions, both Theory 1 and Theory 3 perform equally well, and the Dimension Weights and IBP methods are the best performing comparison methods. For all the different versions the Exceptions method always performs badly and we should therefore discount this as a method of assigning weights.

Note, however, the Dimension Weights method, using the technique described in the previous section is usually one of the best performing methods. This may partly result from the ability to tune the weights more accurately, and clearly to differentiate between normal, weak and knock-out factors, but does suggest that dimensions offer a sensible way to structure factors so as to assign weights.

5.10 Comparison of Simple and Complex Dimensions

When using unweighted factors, better performance can typically be obtained from complex dimensions, but more refinement is required. When using Theory 3 Simple Dimensions for the training set of Group 1 cases, only the Best Factor method needs refining whereas Complex Dimensions need the CATO method refining as well. Apart from the Best Factor method where the Complex Dimensions performs badly, both versions perform equally. As stated above, however, there seems little gain for unweighted factors in moving to dimensions.

For methods using weights, Simple Dimensions and Theory 1 perform best, with all the other versions performing equally well. One theory as to why Simple Dimensions perform better than Complex Dimensions, is that while a factor may have a number of aspects, only one of these is germane to a particular case. Thus although *F27-Disclosure-in-a-Public-Forum* contains elements of disregard for the secret, and fair methods in that the defendant is making use of information in the Public Domain, in the context of a particular case only one of these may matter. And since *F27-Disclosure-in-Public-Forum* is a knock out factor, allowing it to appear on more than one dimension may distort its impact. Alternatively it could be that the loss of the connection to the five issues identified by IBP from the Restatement of Torts is important, supporting the argument of [3] that the intermediate concepts representing these issues are of vital importance for prediction. That the Dimension Weights method provides best results indicates that the position of factors on the dimensions can provide a sensible basis for assigning weights. Moreover the results show that the placing of the factors on the dimension, using information from IBP, seems to be broadly correct.

I wanted to explore three questions with respect to dimensions. Obviously the smallness of the sample size precludes firm conclusions, but I offer the following tentative answers.

- How should we map factors back into dimensions? I have described two methods of performing this mapping. The results suggest that using simple dimensions is enough.

- Can we identify a plausible relationship between values and dimensions? Since Simple Dimensions, which correspond directly to the values identified in Section 3.3.2 perform at least as well as the complex dimensions, the suggestion is that I can use the values to supply the dimensions.

- Can we use the notion of dimensions to produce a principled means of assigning weights to factors? The result that the use of dimensions to determine weights of factors produced the best results whatever the method of theory construction, suggests that this is so.

5.11 Concluding Remarks

In this chapter and the preceding chapter I have described a number of experiments designed to explore the theoretical account of reasoning with legal cases as theory construction described in [10]. I summarise the findings below, always remembering that they must be tentative given the smallness of the size of the sample available.

- I found that including all factors available from the background produced better performance than being selective as to factors. Since I am starting from the analysis of [1], I can expect all factors to be relevant. In a less well analysed or controversial domain it could prove useful to select factors.
- Since I was able to improve performance using additional factors and determining the priority to be given to these new factors by reference to value priorities established using different factors related to that value, suggests that, as was suggested in [10] the importance of factors does relate to their motivating values.
- On comparison of sets of factors I found it best to take all the factors present in the sets into account. Factors appear to have some cumulative effect.
- The improvements obtainable by weighting values and factors, suggest that factors do support the case of the party they favour to different degrees.
- I found that structuring the factors into dimensions corresponding to values provided an effective and principled way to assign weights to factors.

I did not, however, come to any conclusions as to the best method for the construction of theories. This issue is being explored in the following work, in which we will seek ways to automate the construction of theories.

Chapter 6

Automated Theory Construction

This chapter begins the exploration of ways of constructing theories automatically. The idea is to construct theories using a sequence of argument moves of the sort found in systems such as HYPO [2] and CATO [1], which will enable me to reflect the adversarial nature of the domain, and to draw on the wealth of experience with regard to argument construction embodied in these Case-Based approaches. Each move is associated with a set of theory constructors, and thus as the moves are made, the theory is constructed as a side effect. Now, by modelling the process of reasoning as a two player game in which the plaintiff and defendant alternately make argument moves, I can construct a game tree, which will also correspond to the space of possible theories which can be constructed using these moves. This chapter describes the implementation of a program called AGATHA (ArGument Agent for THeory Automation) which models the two player dialogue.

Since the theory search space can become very large when a large number of cases are involved I need a way of limiting the number of theories produced by AGATHA. To do this the theories must first be evaluated to determine how "good" a particular theory is. To accomplish this a second program, ETHEL (Evaluation of THEories in Law), was implemented which takes the theories and evaluates them using various criteria such as Explanatory Power, simplicity and the ability to generalise to new cases. Chapter 7 describes ETHEL and the various parameters used.

Now the relative worth of the theories can be determined, various search heuristics can be used to guide AGATHA through the theory search space. Chapter 8 describes the use of a heuristic based on A* search where both agents are cooperating to try to reach the "best" theory. Chapter 9 describes a different heuristic based on $\alpha\beta$ pruning where the agents are competing with each other to produce the "best" theory but also prevent the other agent from winning with a better theory.

AGATHA only has one case background that both agents can use in creating the theories. The final program, ROSALIND (Daughter of AGATHA), was implemented to explore the results when the two agents have access to different background information and the program and ROSALIND and its results are described in section 9.5.

6.1 Automated Theory Construction

Although legal expert systems developed solely on the basis of rules derived from legislation have had some limited success in favourable domains, it is now generally accepted that they must be supplemented by knowledge derived from case law if they are to make any real contribution to legal problem solving. Even where there are clear rules, problems of interpretation, under-specification and conflict remain. Although cases are used differently in civil and common law jurisdictions this point applies to both styles of legal system [25]. The need to consider cases can be illustrated by considering IBP [15], [3]: although in that system the Restatement of Torts appears to offer some clear rules, the experience of past cases must be drawn upon to apply them to specific cases.

This means that if one is to understand a piece of law, whether with a view to applying it unaided, or to building a decision support system, it is first necessary to come to an understanding of the relevant case law. Bench-Capon and Sartor in [10], suggests that coming to this understanding is best seen as the construction of a theory of the domain, developed from, and intended to explain, the phenomena presented by decisions in precedent cases. Once constructed, the theory can be evaluated according to both internal considerations of coherence and by its effectiveness in accounting for the decisions in the precedent cases. The goal of AGATHA is to derive such a theory automatically on the basis of a set of precedent cases.

When thinking about how to argue a new case on the basis of case law, it seems natural to see the problem in terms of analogising a past case to the problem or by distinguishing an unfavourable case, rather than in terms of the theory constructors proposed by Bench-Capon and Sartor in [10]. To reflect this, I want to drive the process of theory construction in AGATHA using a series of argument moves as found in case based reasoners so as to provide a more transparent rationale for the theories. I therefore draw on the moves of HYPO [2] and CATO [1] for inspiration.

The idea is that AGATHA will use these moves to simulate a dialogue between the plaintiff and the defendant, constructing the theory as a side effect of the dialogue.

6.2 Argument Moves in Case Based Reasoners

The moves of HYPO [2] and CATO [1] as the starting point. HYPO creates 3-ply arguments using these four moves:

1 Analogising a problem to a past case with a favourable outcome.

2 Distinguishing a case with unfavourable outcome.

3 Citing a more-on-point counterexample to a case cited by an opponent.

4 Citing an as-on-point counterexample.

Either party may start the argument by using the first move and *Analogising* the problem case to a past case to create the 1-ply arguments. The opposing party can then use the remaining three moves to *Distinguish* or *Counter* the cited case to create the 2-ply arguments. The original party can then respond by *Distinguishing* or using a *Counter* example to complete the 3-ply argument. When HYPO cites an *as-on-point* case in move 4 it uses a case with as many matching factors as the original case but with a different outcome. A *more-on-point* case has more matching factors than the original case and the different outcome. HYPO tries to use the *most-on-point* cases to prevent the opponent from making a better counter-move.

CATO extended HYPO with four extra moves:

5 Downplaying the significance of a distinction.

6 Emphasising the significance of a distinction.

7 Citing a favourable case to emphasise strengths.

8 Citing a favourable case to argue that weaknesses are not fatal.

Moves 5, 7 and 8 are used in the third ply, and 6 to strengthen a distinction move in the second ply. Again the argument is started by one party using the first move to analogise a past case to the problem case. The opponent can then respond to this move using another move and then the original party can respond to construct 3-ply arguments.

As I am targeting the basic theory of Bench-Capon and Sartor in [10] using factors rather than the extension using dimensions designed to allow downplaying and emphasising distinctions, moves 5 and 6 of CATO cannot be used. In any event I would argue that these concern theory evaluation rather than theory construction. Also I will not adopt moves 7 and 8 from CATO at this stage. Arguably these moves also relate to evaluation as they strengthen rather than develop the theory. I therefore base AGATHA on the moves found in HYPO, although I am using the factor based representation of cases used in CATO rather than dimensions as found in HYPO: again this is because I am using the basic theory of [10], rather than the extended notion which incorporates dimensions.

6.3 AGATHA Theory Moves

AGATHA models the four moves described in HYPO although with some differences to fully utilise the Theory Constructors. The *distinguish* move is expressed as three distinct moves, depending on whether it is the citation of a case, an arbitrary rule preference or a value preference which is advanced to support the opposing view. The *counter example* moves have been merged, since AGATHA only uses the most-on-point cases available. The difference between moves 3 and 4 relates more to evaluation than construction.

The five moves available in AGATHA are: *Analogise Case, Distinguish with Case, Distinguish with Arbitrary Preference, Distinguish Problem*, and *Counter with Case*.

The moves will be described using the cases from the Wild Animal Domain which is described in section 3.2 and was used in Chapter 4. Figure 6.1 shows the diagramatic representation of the similarities and differences between the cases of *Young, Keeble* and *Pierson*.

1. *Analogise Case*. This move cites a precedent case which has the outcome the party making the move desires. The factors which are present in both the prob-

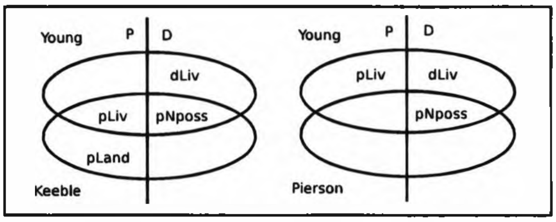

Figure 6.1: Diagramatic representation of the similarities and differences between the cases of Young, Keeble and Pierson.

lem case and the case being cited are sorted into the factors which support that outcome and those factors which support the opposite outcome. A rule preference is made with the supporting factors preferred over the contrary factors. This move follows the method of extracting rules from cases proposed by Prakken and Sartor in [30].

For the example of the plaintiff *analogising* the *Keeble* case to the problem case of *Young* the factors present in both cases are *pNposs* and *pLiv* as shown in Figure 6.1. Because the plaintiff player is making the move the plaintiff rule of *(pLiv→P)* is preferred over the defendant rule of *(PNposs→D)* to give a rule preference of *(pLiv→P)>(PNposs→D)*.

The first move made has to be *Analogise Case*. *Analogise Case* can follow the *Distinguish with Arbitrary Preference* move but if using it introduces inconsistencies within the theory, the rule and value preferences that were introduced by the *Distinguish with Arbitrary Preference* move are removed from the theory and then the *Analogise Case* move can introduce new rule and value preferences. It cannot follow the other three moves.

2. *Distinguish with Case*. This move distinguishes a case already cited in the debate and cites a new case which has the different outcome. To distinguish the previously cited case, AGATHA takes all the factors not used in the *Analogise Case* move which support the outcome and adds them to the factors used in the rule preference from the cited case. So, for example, if the previously cited case was a plaintiff case, AGATHA takes the unused plaintiff factors from that case and adds them to the used plaintiff factors. This creates a larger rule containing all the plaintiff factors from the case which is then preferred over the original

defendant factors. This gives a more complex rule which can be used to decide the previously cited case but cannot be used to decide the problem case because this case does not contain all the factors contained in the new rule preference. AGATHA then cites a precedent case with a different outcome from the previously cited case, to give a theory supporting the other side.

If the plaintiff player had analogised the *Keeble* case, the defendant would want to distinguish *Keeble* and analogise a defendant case to support the defendant theory. The defendant does this by including the extra plaintiff factor of *pland*, which is present in *Keeble* but not in *Young*, into the rule preference. This gives a rule preference of *(pLiv,pLand→P)>(pNposs→D)* which can be used to explain the decision in the case of *Keeble* but cannot be used to explain *Young*.

Distinguish with Case can follow the *Analogise Case, Distinguish with Case* and *Counter with Case* moves because these all cite a new case. It cannot follow the *Distinguish with Arbitrary Preference* and *Distinguish Problem* as these do not cite a case.

3. *Distinguish with Arbitrary Preference*. This move distinguishes the previously cited case in the same way as for the *Distinguish with Case* move, but instead of analogising a new case, AGATHA makes an arbitrary preference using the factors from the problem case that are included in the theory and only these factors. If, for example, AGATHA is making a plaintiff move, the arbitrary preference has the plaintiff factors preferred over the defendant factors, otherwise, for a defendant move, the defendant factors are preferred over the plaintiff factors. The preference is arbitrary because there is no support for the preference; it just depends on what the party making the move needs to assume to make their case.

   It can follow the *Analogise Case, Distinguish with Case* and *Counter with Case* moves because they all cite a new case. It cannot follow the *Distinguish with Arbitrary Preference* and *Distinguish Problem* as these do not cite a new case.

4. *Distinguish Problem*. This move distinguishes the problem case instead of the previously cited case. If, for example, AGATHA is making a plaintiff move, it takes all the plaintiff factors from the problem case and conjoins them as the antecedent into a single rule with plaintiff as consequent. The defendant factors from the problem case are similarly conjoined as the antecedent of a single rule

with defendant as consequent. Next the value sets comprising the values associated with the factors in the two rules are created and a value preference is created with the value set corresponding to the plaintiff factors being preferred over the value set from the defendant factors. Finally a rule preference is created using this value preference.

If the plaintiff had analogised the *Keeble* case, the defendant would want to distinguish *Young* and not *Keeble*. The defendant does this by merging the extra defendant factor of *dLiv* from *Young* with the defendant factor of *pNposs* to create the complex rule of *(dLiv,pNposs)→D*. The defendant wants to create a rule preference of *((dLiv,pNposs)→D)>((pLiv)→P)* so it creates the corresponding value preference (recall that when a rule preference is included in the theory that the corresponding value preference is also included in the theory) and includes this in the theory if some conditions are met. The corresponding rule preference can then be included in the theory.

It can follow the *Analogise Case, Distinguish with Case* and *Counter with Case* moves because they all cite a new case. It cannot follow the *Distinguish with Arbitrary Preference* and *Distinguish Problem* as these do not cite a new case.

5. *Counter with Case.* This move counters the previously cited case by finding a case which is as-on-point or more-on-point as the previous case but was decided for the other side. For an as-on-point counter move, the new case must have the same factors matching the problem case as the previously cited case. The original rule and value preferences which are supported by the previously cited case are replaced with new preferences which are opposite to the original preferences and are supported by the new case. For a more-on-point counter move, the new case must have the same factors matching the problem case as the previously cited case and extra factors which match the problem case but are not present in the previously cited case. The original rule and value preferences supported by the previously cited case are replaced by new preferences which are supported by the new case.

   If the defendant had analogised the *Pierson* case then there is one rule *(pNposs→D)* included in the theory (as there is only one rule there cannot be a rule preference). The plaintiff can *counter* with the *Keeble* case because it has the defendant factor of *pNposs* but also the plaintiff factor of *pLiv* and so is more-on-point than *Pierson*. The plaintiff rule of *(pLiv→P)* is included into the theory and if *Pierson* had provided a rule preference then this would be removed before the new rule preference of *(pLiv→P)>(pNposs→D)* was included.

It can follow the *Analogise Case, Distinguish with Case* and *Counter with Case* moves because they all cite a new case. It cannot follow the *Distinguish with Arbitrary Preference* and *Distinguish Problem* as these do not cite a case.

The moves can only be made once to a given theory apart from *Distinguish with Arbitrary Preference* which can be made more than once, and *Counter with Case* and *Distinguish with Case* which can be made once for each of the cases which are in the case background. Note also that AGATHA may extend beyond the third ply if moves are available to do so.

6.4 Theory Moves and Constructors

The argument moves used in AGATHA use the theory constructors described in Bench-Capon and Sartor in [10] and in chapters 2 and 4 to create the underlying theory. When a move is made, a number of theory constructors are applied to extend the current theory. For example, the *Analogise Case* move uses the *Include Case* constructor to include the cited case into the theory, the *Include Factor* constructor to include all the matching factors with the problem case and the *Merge Factors* constructor to merge the plaintiff and the defendant factors so that they can serve as antecedents of a complex plaintiff and complex defendant rule. Finally it uses the *Preferences from Case* constructor to include the rule preference which is used to explain the decision for the cited case. Table 6.1 shows the Theory Constructors which are used in each move.

Table 6.1: Table of Theory Constructors associated with each move.

Move	Theory Constructors
Analogise Case	Include Case Include Factors Merge Factors Preferences From Case
Distinguish With Case	Include Case Include Factors Merge Factors Preferences From Case Remove Rule Preference
Distinguish with Arbitrary Preference	Include Factors Merge Factors Remove Rule Preference Preferences From Case Arbitrary Preference
Distinguish Problem	Include Factors Merge Factors Value Preferences Rule Preference From Value Preference
Counter with Case	Include Case Include Factors Merge Factors Preferences From Case Remove Rule Preference

6.5 AGATHA Program

AGATHA models adversarial dialogue between two agents with each agent taking turns to make a move to produce a theory. As described above, AGATHA has five moves that it can use according to certain preconditions and it applies all possible moves at each point until no more moves can be made.

AGATHA's interface, shown in Figure 6.2, contains buttons along the top of the interface that, when selected, allow the user to open an existing project, start a new project or modify the currently open project. The user can also choose to execute all the theories in the project to produce a table of the theories and the corresponding outcomes for the problem case.

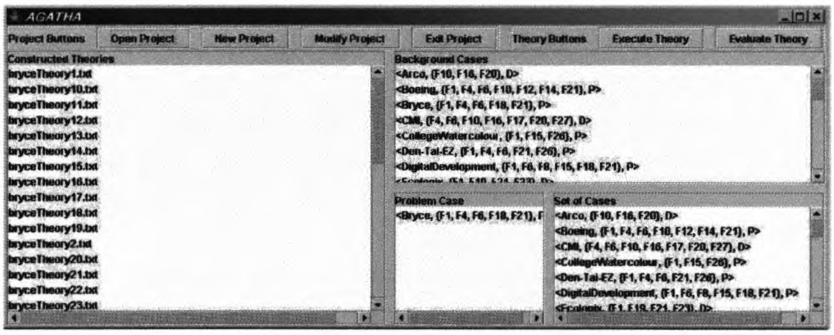

Figure 6.2: Screen shot of AGATHA.

When AGATHA is first started it prompts the user to create a new project or to open an existing project. If the user chooses to start a new project they are asked to choose the problem case which is to be decided and then they are asked to choose the set of cases to be used to explain the decision for the problem case. If the user chooses to open an existing project they can then modify the project by replacing the problem case or by adding or removing cases from the set of precedent cases. When the project has been given a name, AGATHA runs the theory constructor to create all the theories that can be produced in this context.

AGATHA starts theory construction by creating the initial theory (Theory 0) which only contains the problem case. AGATHA then takes this initial theory and applies all the moves which are applicable to it and numbers all the theories sequentially. AGATHA takes the first theory produced (Theory 1) and applies all the applicable moves to it and repeats this with the next theory (Theory 2) until there are no more theories left. For Theory 0, AGATHA can apply both plaintiff and defendant moves but for all the subsequent theories the players must alternate, so a plaintiff move must follow a defendant move, and vice versa.

AGATHA checks which moves can be made by checking the preconditions for each move against the theory at that point in the game tree and, if the preconditions match, it applies the move. Each move that can be applied produces a new theory. When alternative moves are available, new branches are added to the tree of theories being created.

As each move is applied to the theory, the resulting theories are examined and only those which give the same outcome for the problem case as the party making the move are retained. If the move made does not give the desired outcome, the theory is discarded because, even though the move could be applied, it does not help the party making the move, and so does not represent a sensible move.

The effect of this is to give a breadth first construction of the tree of theories.

6.6 Experiments With AGATHA

6.6.1 Wild Animal Example

This illustrative example uses the widely discussed wild animal cases described in section 3.2 and used in [10] and Chapter 4 to illustrate the use of the theory constructors. This small example allows an exhaustive walk through of the operation of AGATHA.

The cases used are *Keeble, Pierson* and *Young* which are described in section 3.2 and section 6.4. *Young* is taken as the problem case with *Pierson* and *Keeble* as the set of cases that AGATHA can use to create the theories. *Keeble* is a plaintiff case and has two factor matches with the problem case. *Pierson* is a defendant case and has one factor matching with the problem case.

Using all the moves defined in AGATHA, AGATHA creates ten theories which are shown in Figure 6.3. Figure 6.3 also shows how the theories relate to each other. The rules, rule preferences and value preferences for the subsequent theories are shown in Table 6.2. Section D.1 in Appendix D shows the theories in full as well as the results obtained for all the Wild Animal cases when the theories are executed.

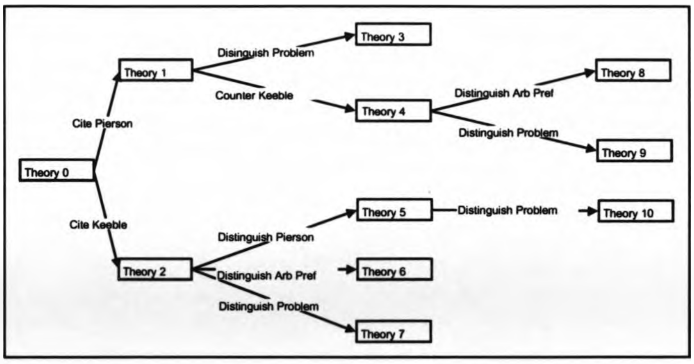

Figure 6.3: Theories produced.

```
Theory Cases :
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
Theory Rules :
Theory Preferences :
Theory Value Preferences :
```

Figure 6.4: Theory 0

From Theory 0 (Figure 6.4) only the *Analogise Case* move can be made. First the defendant move is made by analogising *Pierson* to the problem case to produce Theory 1 (Table 6.2). Then the plaintiff move is made by analogising *Keeble* to the problem case to produce Theory 2.

From Theory 1 the *Distinguish with Case* and *Distinguish with Arbitrary Preference* moves cannot be made because there are no extra factors that can be used to distinguish *Pierson*. *Distinguish Problem* can be made to distinguish *Young* and produce Theory 3. *Counter with Case* can be made because *Keeble* is more-on-point than *Pierson* and produces Theory 4. Although, as discussed below, these theories contain the same rules and preferences, the justification of the rules and preferences and the moves available may be different for each theory.

From Theory 2 *Distinguish with Case* can be used to distinguish *Keeble* and cite *Pierson* to produce Theory 5, *Distinguish with Arbitrary Preference* produces Theory 6 and *Distinguish Problem* produces Theory 7. *Counter with Case* cannot be used because *Pierson* is less-on-point than *Keeble*.

From Theory 3 there are no moves that can be made so this line of the dialogue stops.

Table 6.2: The rules, rule preferences and value preferences for the theories.

Theory	Rules	Rule Preference	Value Preference
1	(1) pNposs→D		
2,3, 4	(1) pNposs→D (2) pLiv→P	(2)>(1)	MProd>LLit
5	(1) pNposs→D (2) pLiv→P (3) pLand→P (4) (pLiv,pLand)→P	(4)>(1)	(MProd,MSec)>LLit
6, 8	(1) pNposs→D (2) pLiv→P (3) pLand→P (4) (pLiv,pLand)→P	(4)>(1) (1)>(2)	(MProd,MSec)>LLit LLit>MProd
7,9	(1) pNposs→D (2) pLiv→P (3) dLiv→D (4) (dLiv,pNposs)→D	(2)>(1) (4)>(2)	MProd>LLit (LLit,MProd)>MProd
10	(1) pNposs→D (2) pLiv→P (3) pLand→P (4) (pLiv,pLand)→P	(4)>(1) (2)>(1)	(MProd,MSec)>LLit MProd>LLit

From Theory 4 *Distinguish with Case* and *Counter with Case* cannot be used because there are no more defendant cases to be cited. *Distinguish with Arbitrary Preference* produces Theory 8 and *Distinguish Problem* produces Theory 9.

From Theory 5 *Distinguish with Case* and *Distinguish with Arbitrary Preference* cannot be used because *Pierson* has no more factors that could be used to distinguish it. *Distinguish Problem* produces Theory 10. Note that the alternative way of distinguishing the problem, by preferring *MSec* to *LLit* cannot be used because *pLand* is not present in *Young*, and so this would not produce a pro-plaintiff theory for *Young*. *Counter with Case* cannot be used as there are no more defendant cases that can be used.

From Theory 6 the only potential move is *Analogise Case*, but this move cannot be used because there are no remaining plaintiff cases.

From Theory 7 there are no moves that can be used.

From Theory 8 the only potential move *Analogise Case*, but again this move cannot be used because there are no remaining plaintiff cases.

From Theory 9 there are no moves that can be used.

From Theory 10 there are no moves that can be used. The tree is therefore complete.

From an analysis of the preference sections of the theories, it can be seen that several theories have identical preferences, even though these preferences may have different labels and have been produced using different moves. There are three groups of identical theories and two theories which are different from all the others.

The first group of theories contains Theories 2, 3 and 4. Theory 2 and Theory 4 are identical because Pierson only has one factor and so cannot contribute a rule preference so, for Theory 4 when *Counter with Case* is used, Keeble contributes the same rule preference as *Analogise Case* for Theory 2. Theory 3 is a plaintiff theory and so takes the defendant *pNposs* factor from Theory 1 and adds the plaintiff factor from the Young case description and creates a rule preference of *(pLiv→P > pNposs→D)* which is the same rule preference which Keeble contributes.

The second group contains Theories 6 and 8. These are identical because their preceding theories are also identical (Theory 2 proceeds Theory 6 and Theory 4 precedes Theory 8) and they are produced by making the same move.

The third and final group contains Theories 7 and 9 and they are identical due to the same reasoning as for the second group.

Theory 5 is a distinct theory. To create Theory 5 from Theory 2, Keeble is distinguished and Pierson is cited but Pierson only has a single factor which is already present in the theory and so does not contribute a rule preference. A pro-defendant outcome is produced, however, because the rule preference of *({pLiv, pLand}→P)* over *(pNposs→D)*, is not applicable to Young, since *pLand* is not present, which allows *(pNposs→D)* to fire and give an outcome for the defendant.

This example is also described in section 4.3 where the theories are produced by hand. In section 4.3 four theories are produced which correspond to theories 1, 4, 8 and 9. This is because only one branch of the theory space is followed. First *Pierson* is cited to produce Theory 1 for both examples. Then the opponent uses the *Counter* move with *Keeble* to give Theory 2 which AGATHA calls Theory 4. Finally the *Distinguish with Arbitrary Preference* move is used to construct Theory 3 and the *Distinguish Problem* move constructs Theory 4. AGATHA calls these Theories 8 and 9 respectively.

These four theories were used to describe a possible process of theory construction using the theory constructors described in [10]. AGATHA also constructs these theories, but also provides alternative theories which may be better (or worse) than those produced by hand.

6.6.2 US Trade Secrets Misappropriation Law Example

I have also used AGATHA on the other test domain, US Trade Secrets Misappropriation Law, as modelled in [1] and described in section 3.3. This is a larger domain than the wild animals, containing 32 cases, 26 factors and 5 values.

For this experiment I used the case of *Mason versus Jack Daniels* as the problem case. The *Mason* case is described in [1] and was quoted in section 3.4. The number of cases in the case background is increased to explore what happens to the size of the theory search space. Table 6.3 shows the cases which are used in the three backgrounds. Background 1 uses 4 cases, Background 2 has the 4 cases from Background 1 and an extra 2 two cases and Background 3 extends Background 2 by another 2 cases. All of these background cases have two factors matching with *Mason*.

Table 6.3: Original Backgrounds

	Plaintiff Cases	*Defendant Cases*
Background 1	Goldberg National Instrument	Ecologix Sandlin
Background 2	Space Aero	National Rejectors
Background 3	Trandes	CMI

Running AGATHA on the case of *Mason versus Jack Daniels* produces the results shown in Table 6.4. With the limited set of background cases of two Plaintiff cases and two Defendant cases, AGATHA produces a tree of depth 7 with 106 nodes. Adding a further two cases gives rise to a theory space with a maximum depth of 8 and 653 nodes. Again adding another two cases produces a tree of depth 11 and 2855 nodes. Even when using the first background with only four cases there are too many theories produced to analyse in a meaningful way.

As the domain becomes larger, the game tree, and hence the theory space, becomes very much larger. This is entirely to be expected, and as this is what invariably happens to a game tree when we move from a simpler to a more complex game. It means, however, that the exhaustive construction of the theory space will not always be the best strategy for realistically large problems, especially as I want to avoid being selective in the inclusion of cases in the background.

Table 6.4: Original AGATHA Results.

Name	Number of Cases		Number of Nodes	Tree Depth	Leaves
	Plaintiff	Defendant			
original1	2	2	106	7	60
original2	3	3	653	8	377
original3	4	4	2855	11	1730

6.7 Conclusion

Running AGATHA with brute force to produce the complete theory space:

1. shows that theories can be constructed using this method
2. shows "completeness" for the Wild Animal Domain, reflecting the theories constructed in [10]
3. shows that the tree becomes unusably large for sizeable domains.

AGATHA has shown that it is possible to construct the space of theories of a case law domain by applying argumentation moves derived from work on reasoning with legal cases. By using these moves AGATHA is following a cognitively plausible strategy, and the sequence of moves is available to present the case to an opponent.

At this stage AGATHA generates the complete theory space. This space may include duplicate theories, and theories which seem less acceptable than others. I have made no attempt to prune the space since it may be of interest to see that theories can be reached by different routes, so that it can be seen whether the same solution is independent of who makes the first move, and the order in which moves are made. Moreover different routes may supply different justifications for various theory elements. None the less, as the size of the domain increases, there may be advantages in pruning the space. One obvious way to do this is expand only one instance of identical theories. An alternative, and more promising approach is to exploit the fact that we are dealing with a process that is a two player adversarial game. In such games, it is typically the case that it is impractical to generate the whole game tree and so techniques have been developed to address this problem in the analysis of two player games, using heuristic search techniques.

Therefore a second way of controlling the expansion of the tree is to provide some heuristic to select the moves to apply. I could rank moves according to their potential strength: one plausible ranking would be *Counter with Case, Distinguish with Case, Distinguish Problem, Distinguish with Arbitrary Preference.* In this way only a single branch of the tree would be produced from each node. A second source of expansion is the choice of cases to cite: again some heuristic in term of similarity to the problem case would provide a sensible means of limiting this growth.

A more sophisticated approach would be to use a variety of heuristic search. To do this I will need to have a means of evaluating the theories produced as the tree is developed. I will then be able apply a standard technique to prune the game tree, such as A* search or $\alpha\beta$ pruning e.g. [40]. The next step therefore is to develop a means to evaluate the theories. I will then apply this method to explore two varieties of heuristic search in Chapters 8 and 9

Chapter 7

Evaluation of Theories

To enable the use of search heuristics the theories produced by AGATHA need to be evaluated to provide a way of comparing the theories. ETHEL stands for Evaluation of THEories in Law and evaluates theories using criteria similar to those proposed by Bench-Capon and Sartor in [10], including explanatory power, simplicity, freedom from arbitrary preferences and the ability to generalise to new cases. ETHEL first analyses the constructed theories to create a table reflecting some key metrics of the theory.

7.1 Evaluation Criteria

The following five criteria are used to measure how "good" a theory is:

1. *Simplicity.* ETHEL counts the number of rule preferences in the theory, the number of arbitrary rule preferences and the number of rule preferences that are obtained from value preferences.

2. *Explanatory Power.* Each theory is executed (using a Prolog program automatically generated from the theory as described in Chapter 4) with the complete set of background cases and the results analysed. First the total number of cases in the background is found, and then the number of cases which received the same outcome from the theory as their actual outcome are counted (to give the number of correctly decided cases). Next the cases which received the wrong decision from the theory are counted (giving the incorrectly decided cases) and finally

the cases for which the theory could not give an outcome are counted (giving abstentions). This can be used to show how well (or badly) a theory generalises from the cases used in its construction.

3. *Completion Explanatory Power.* The *Explanatory Power* criterion executes a program which uses only the factors which are specifically used in the construction of the theory. This gives a restricted set of factors to be considered when deciding the cases. For the third criterion all the background factors are loaded into the theory, the theory is completed as described in Chapter 4 and the program produced from this extended theory is executed on the complete set of background cases. Again the results are analysed to give the total number of cases, the number of cases which are correctly decided, the number of cases which are incorrectly decided and the number of abstentions. This reflects how well the value preferences in the theory perform.

4. *Depth.* The theory is given a number corresponding to its depth in the tree.

5. *Leaf Node.* The table indicates if the theory is a leaf node in the tree and hence has no more moves that can be made to reverse its decision.

7.2 Evaluation Parameters

ETHEL now uses this set of metrics to calculate an *Evaluation Number* for each theory. This is intended to measure how good the theory is and is composed from the above five criteria. For each criterion I provide a way of turning the associated metrics into a number. I need to use a number of parameters which are to a certain extent pragmatic, justified only by their effect on the evaluation number: those given in this section were the initial choices, and were varied in some of the experiments described in subsequent chapters.

1. *Simplicity.* The value for *Simplicity* is composed of three parts; a value based on the number of rule preferences in the theory, a value based on the percentage of the rule preferences that are Arbitrary and a value based on the percentage of the rule preferences that are Value-Based. The values for the Arbitrary and Value-Based rule preferences are subtracted from the value for the total number of rule preferences. Table 7.1 shows some example calculations when the total number

of rule preferences is varied as well as the number of Arbitrary and Value-Based rule preferences.

A simpler theory is better than a more complex theory. The simplest theory would only contain one rule preference and this should not be an arbitrary rule preference or a rule preference from value preference.

If there are no rule preferences then the Simplicity value is zero: if there is only one rule preference then the value is 100 and otherwise the value is 100 decreased by a certain percent for each additional rule preference. In our experiments we used 10% as our discount factor.

For example, a theory with no rule preferences has a value of zero, a theory with one rule preference has a value of 100 and a theory with two rule preferences has a value of 90, a theory with three rule preferences 81 and so on.

The value given by the number of Arbitrary preferences is given by the percentage of the total number of rule preferences which are Arbitrary. If the theory has only one rule preference and it is Arbitrary then the Arbitrary value is 100 which is subtracted from the total rule value of 100 to give a *Simplicity* value of 0. If the theory has two rule preferences and only one is Arbitrary then the Arbitrary value is 50 and is subtracted from the total rule preference value of 90 to give a *Simplicity* value of 40.

The value for the Value-Based rule preferences is calculated in the same way as for the Arbitrary rule preferences but the value used is reduced to two fifths. This is because we consider preferences based on Value preferences to be more principled than those expressed as Arbitrary rule preferences. For example, a theory with one rule preference which is Value-Based has a value of 40 which is subtracted from the total rule preference value of 100 to give a *Simplicity* value of 60. If the theory has two rule preferences and only one is value based then the value is 20 and is subtracted from the total rule preference value of 90 to give a *Simplicity* value of 70.

2. *Explanatory Power*. The value for the Explanatory Power is given by the number of correctly decided cases plus half the abstention cases divided by the total number of cases and multiplied by 100 to give a percentage of the total number of cases.

Table 7.1: Values calculated for Simplicity with different numbers and types of rule preferences

Total No. Rules	No. Arbitrary Rules	No. Value Based Rules	Value For No. of rules	Subtract for Arbitrary Rules	Subtract for V-Based Rules	Simplicity Value
0	0	0	0	0	0	0
1	0	0	100	0	0	100
2	0	0	90	0	0	90
3	0	0	81	0	0	81
1	1	0	100	100	0	0
2	1	0	90	50	0	40
3	1	0	81	33.33	0	47.67
1	0	1	100	0	40	60
2	0	1	90	0	20	70
3	0	1	81	0	13.33	67.67
2	1	1	90	50	20	20
3	2	1	81	66.67	13.33	1
3	1	2	81	33.33	26.67	21

3. *Completion Explanatory Power.* This value is calculated in the same way as for the Explanatory Power.

   The above three values are summed to give a basic *Evaluation Number* which is based on how well the theory performs in explaining the background cases and its simplicity. I now adjust this number according to the position of the theory in the tree.

4. *Depth.* The basic *Evaluation Number* can be increased by adding a value which represents how deep the theory is in the tree. This encourages AGATHA to explore the search space more deeply. Initially the *Evaluation Number* is increased by 10 percent for each additional level greater than level 1.

5. *Leaf Node.* The depth-extended *Evaluation Number* can be increased again by adding a value which represents whether the theory is a leaf theory, to reflect the fact that this theory cannot be profitably modified by an opponent. If the theory is a leaf theory then the *Evaluation Number* is increased further, again initially by 10 percent.

These *Evaluation Numbers* give a value with which to compare the theories based on how well they explain the background, their structure and their position in the development of the game tree. They can be used to evaluate the nodes in the theory tree, and so guide a heuristic search.

7.3 Wild Animal Results

In this section I will illustrate the evaluation of ETHEL using a simple example. Table 7.2 shows the evaluation of ten theories obtained when the Wild Animal domain is used to explain the *Young* case. These theories were described in section 6.6 and have been listed in detail in appendix D.

The table shows the number of rule preferences (total, arbitrary and value-based) that each theory contains which will be used to calculate the *Simplicity* value. It also shows the results for both the *Explanatory Power* and the *Completion Explanatory Power* in terms of the number of cases, the number of cases correctly decided for each theory, the number of cases incorrectly decided and the number of cases that the theory abstained on. Finally it shows how deep the theory is in the tree and whether it is a leaf node.

Table 7.2: Evaluation of Wild Animal Theories

Theory Name	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Simplicity										
Number of Rules	0	1	1	1	1	2	2	2	2	2
Arbitrary Rules	0	0	0	0	0	1	0	1	0	0
Value-Based Rules	0	0	1	0	0	0	1	0	1	1
Explanatory Power										
Total Cases	3	3	3	3	3	3	3	3	3	3
Correct Cases	2	3	3	3	3	3	3	3	3	3
Incorrect Cases	1	0	0	0	0	0	0	0	0	0
Abstain Cases	0	0	0	0	0	0	0	0	0	0
Completion Explanatory Power										
Total Cases	3	3	3	3	3	3	3	3	3	3
Correct Cases	2	2	2	2	3	3	3	3	3	2
Incorrect Cases	1	1	1	1	0	0	0	0	0	1
Abstain Cases	0	0	0	0	0	0	0	0	0	0
Extensions										
Depth	1	1	2	2	2	2	2	3	3	3
Leaf	No	No	Yes	No	No	Yes	Yes	Yes	Yes	Yes

Table 7.3 then shows the values that are obtained when the metrics described in section 7.2 are applied to the results in Table 7.2. An *Evaluation Number* is calculated from the *Simplicity* value, the *Explanatory Power* and the *Completion Explanatory Power* and can be used to evaluate the theories. The next *Evaluation Number* is calculated by including the value for the depth contribution. This is used when the depth of the theory plays a part in deciding how "good" it is. A final *Evaluation Number* is calculated by including the value for the leaf contribution. This is used when it is important that a theory is a leaf node and cannot be modified.

Any of the *Evaluation Numbers* can be used to evaluated the theories depending on what is being tested.

Theory 5 is the best theory for the basic and the depth-extended *Evaluation Numbers* but it is not the best when the final *Evaluation Number* is considered because it is not a leaf node. It only has one rule preference so it has a high score for *Simplicity* and it gets all the cases correct so has a high score for both *Explanatory Power* and *Completion Explanatory Power*.

Theories 2 and 4 also have a high score for *Simplicity* as they only have one rule

Table 7.3: Calculation of Evaluation Numbers for Wild Animal Theories

Theory Name	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Total Value	0.00	100.00	100.00	100.00	100.00	90.00	90.00	90.00	90.00	90.00
Arbitrary Value	0.00	0.00	0.00	0.00	0.00	50.00	0.00	50.00	0.00	0.00
Value-Based Value	0.00	0.00	40.00	0.00	0.00	0.00	20.00	0.00	20.00	20.00
Simplicity	0.00	100.00	60.00	100.00	100.00	40.00	70.00	40.00	70.00	70.00
Explanatory Power	66.67	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Completion EP	66.67	66.67	66.67	66.67	100.00	100.00	100.00	100.00	100.00	66.67
Evaluation Number	133.33	266.67	226.67	266.67	300.00	240.00	270.00	240.00	270.00	236.67
Depth	0.00	0.00	22.67	26.67	30.00	24.00	27.00	48.00	54.00	47.33
Evaluation Number	133.33	266.67	249.33	293.33	330.00	264.00	297.00	288.00	324.00	284.00
Leaf	0.00	0.00	24.93	0.00	0.00	26.40	29.70	28.80	32.40	28.40
Evaluation Number	133.33	266.67	274.27	293.33	330.00	290.40	326.70	316.80	356.40	312.40

preference but they both get the *Young* case wrong when all the factors are included in the theory to calculate the *Completion Explanatory Power*. This is due to the value preference of $MProd > LLit$ which means that both *pLiv* and *dLiv* are preferred over *pNposs* and because there is no preference between *pLiv* and *dLiv* they are sorted alpha-numerically and so *dLiv* appears in the sorted rule list above *pLiv* and so it is *dLiv* which decides *Young* and not the wanted factor of *pLiv*.

Theories 6, 7, 8 and 9 score well for both *Explanatory Power* and *Completion Explanatory Power* as they get all the cases correct but they score badly for *Simplicity* because they each have two rule preferences and one of them is either an arbitrary preference (Theories 6 and 8) or a value based preference (Theories 7 and 9).

7.4 US Trade Secret Misappropriation Law results

To provide a basis for comparison for later experiments I first ran AGATHA on some selected examples. Here I produce the whole search space: in later chapters ETHEL will be used to guide the search heuristics.

Tables 7.4, 7.5 and 7.6 show the results for Backgrounds 1, 2 and 3 (described in section 6.6.2) respectively when *Mason*, *CMI* and *Digital Development* are used as the seed case. Tables D.1, D.2 and D.3 in Appendix D show the results for all 3 backgrounds when every case is used as a seed case. *Mason* is a well balanced case with three plaintiff factors and two defendant factors, *CMI* is a strongly pro-defendant case with two plaintiff factors and five defendant factors, and *Digital Development* is a strongly pro-plaintiff case with five plaintiff factors and one defendant factor. Because *CMI* is a case in the third background only the first two backgrounds can be used to try to explain *CMI*.

Table 7.4: Complete AGATHA Results for *Mason* when using the three backgrounds.

Name	Background	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
Mason1	1	106	6	28	28
Mason2	2	653	8	30	30
Mason3	3	2855	10	30	30

As the number of background cases increases, the number of moves that could be used also increases (for both *Distinguish with Case* and *Counter with Case* moves) and

Table 7.5: Complete AGATHA Results for *CMI* when using the first two backgrounds.

Name	Background	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
CMI1	1	2	2	10	24
CMI2	2	4	2	10	24

Table 7.6: Complete AGATHA Results for *Digital Development* when using the three backgrounds.

Name	Background	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
DigDev1	1	30	4	28	28
DigDev2	2	37	4	28	28
DigDev3	3	65	4	28	28

hence the number of theories that are constructed also increases. This also means that the depth of the tree may also increase.

Mason obtains the best results followed by *Digital Development* and then *CMI* has the worst results. The results for *Mason* improve as the background gets larger but for *Digital Development* and *CMI* even though the theory tree increases the results do not improve.

Digital Development has similar factors to *Mason* and so has several cases in the backgrounds to be used by the plaintiff and defendant agents. *CMI* does not have many factors matching those in the background cases so the dialogues and theories produced are not very good and cannot explain many of the background cases.

Figures 7.1 and 7.2 show the tree constructed by AGATHA when the case of *CMI* is used as the seed case with backgrounds 1 and 2. *CMI* only produces a small tree with backgrounds 1 and 2 because it does not have many factors in common with the case backgrounds and so AGATHA is very limited in the moves it can use.

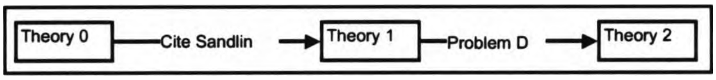

Figure 7.1: Theory Tree produced when CMI is the seed case and the first background is used with no search heuristic.

Figures 7.3 and 7.4 show the tree constructed by AGATHA when the case of *Digital Development* is used as the seed case with backgrounds 1 and 2. *Mason* is not drawn

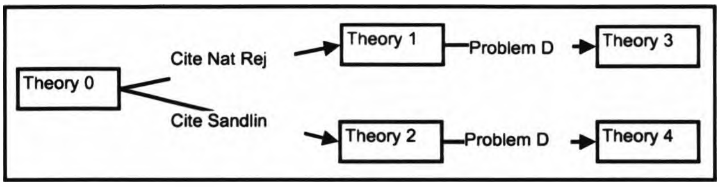

Figure 7.2: Theory Tree produced when CMI is the seed case and the second background is used with no search heuristic.

as it produces too many theories. Figure 7.3 shows the tree produced when AGATHA has the four cases from background 1 to produce moves from and Figure 7.4 shows the tree produced when the six cases from background 2 are used to produce moves. As the number of cases in the background increases AGATHA has more moves that can be made at each node in the dialogue but the depth of the tree does not increase.

Out of all the seed cases *Mason* always obtains the best results with *Valco-Cincinnati* next best. This is to be expected as the backgrounds were chosen to explain the *Mason* case.

Mason and *Valco-Cincinnati* are both Plaintiff cases with a large number of factors which match with many of the factors in the background cases and so produce large rule preferences which can be used to decide many of the cases. *College Watercolor* performs poorly for background 1 and improves for backgrounds 2 and 3. This is because *College Watercolor* is a much smaller case and only has factors in common with *Space Aero* introduced in background 2 so the results for background 1 are very poor and they improve for backgrounds 2 and 3 because of *Space Aero*.

Another poorly performing seed case is *Ferranti*. *Ferranti* has no plaintiff factors in common with any of the background cases. This means that there are no rule preferences produced and so the qulaity of the theory is determined by the alpha-numerical sorting of the rules when the theory is executed and the seed case makes no contribution to the theory.

These experiments show that it is very difficult for a human user to choose background cases because a background chosen to explain a single case can be very good at explaining that case and can generalise to other cases but if it is used to create a theory to explain a different case it can perform very badly and generalise badly.

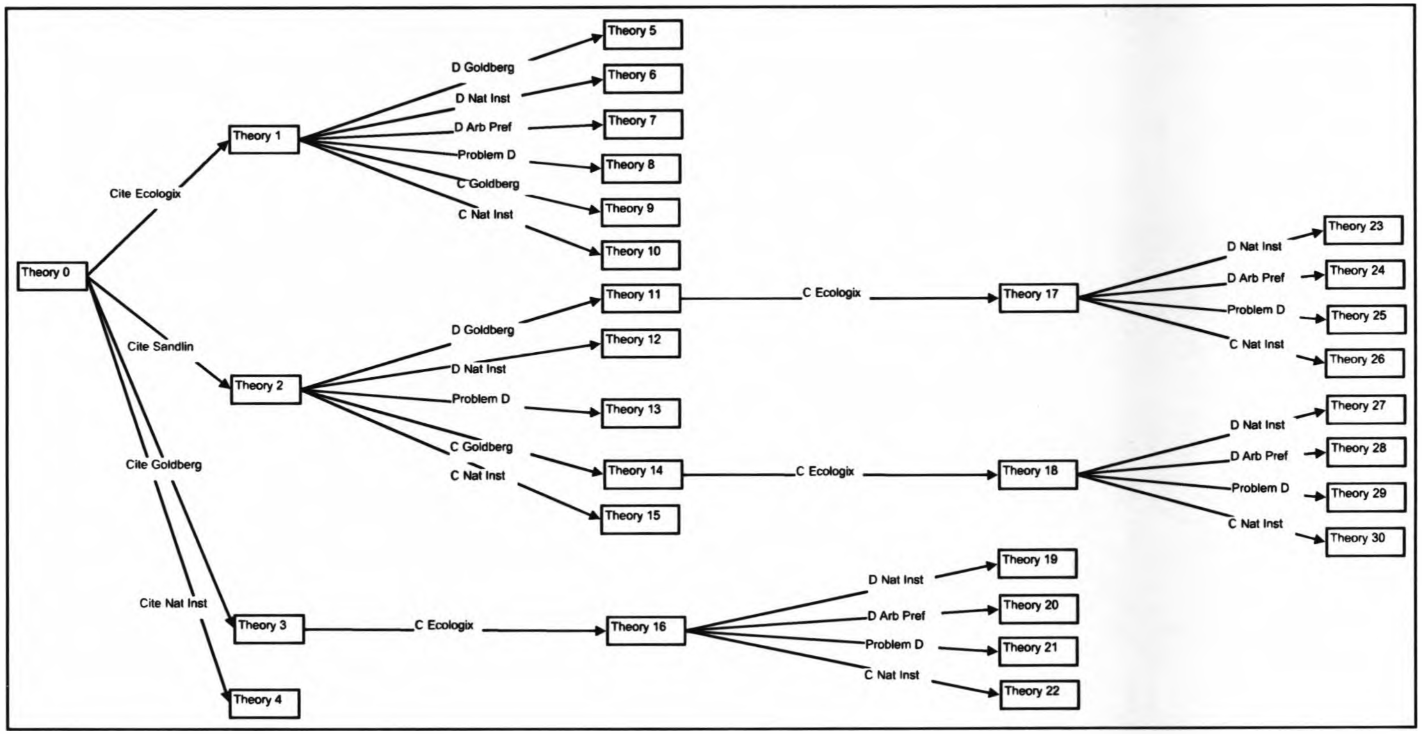

Figure 7.3: Theory Tree produced when Digital Development is the seed case and the first background is used with no search heuristic.

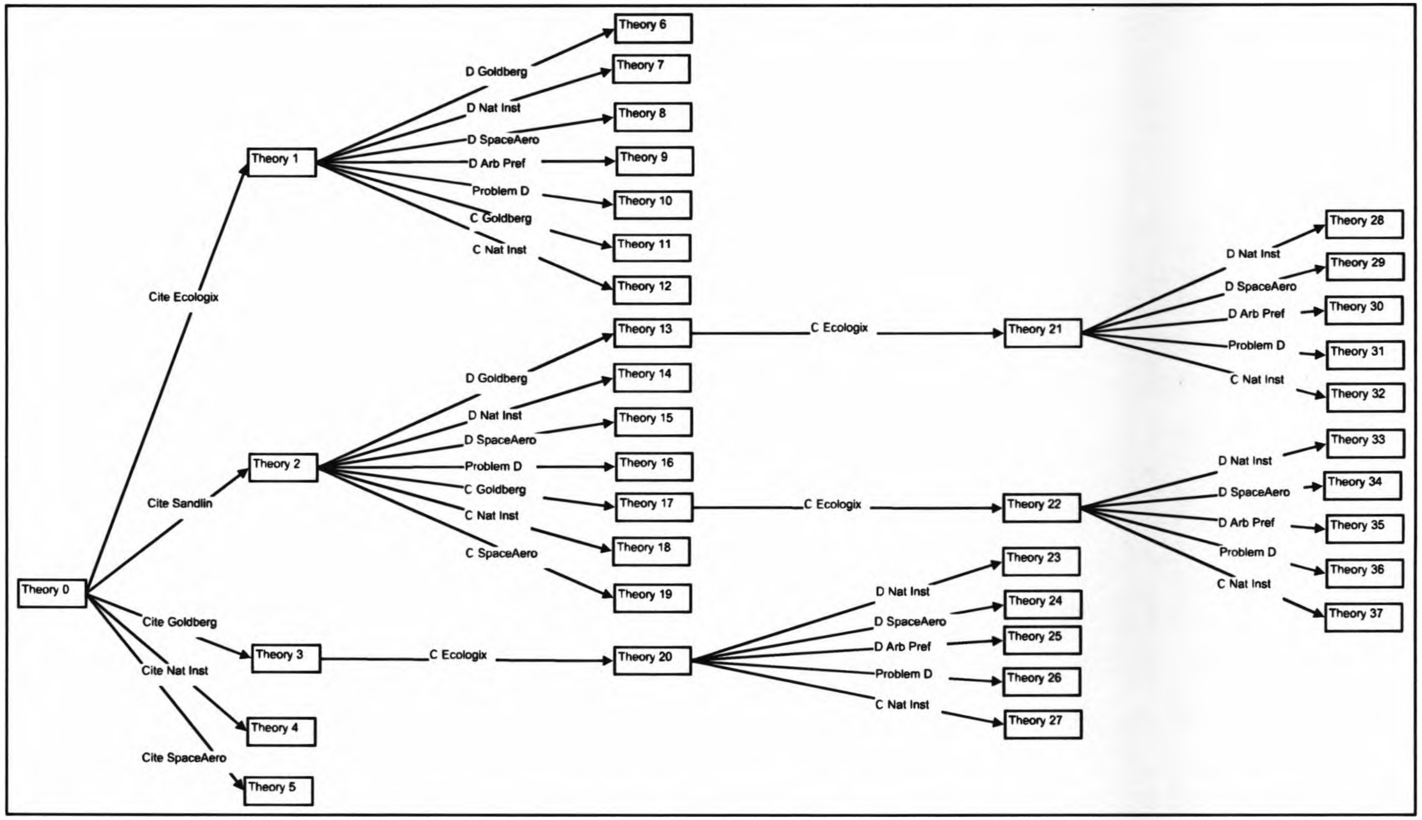

Figure 7.4: Theory Tree produced when Digital Development is the seed case and the second background is used with no search heuristic.

Chapter 8

A* Search Heuristic

8.1 A* Search

In Chapter 6 I showed that AGATHA is able to construct a set of plausible theories and for a small case background they can be exhaustively examined and analysed. However as the size of the case background increases the search space can rapidly become very large causing computational problems and difficulties in interpreting the output. Because the automation process is modelled as a two player game, there are several search heuristics, described in many textbooks including [40], [35] and [31], that could be implemented to guide the agents into making the best move. Heuristic search can give very good but not necessarily optimal results [31].

A* is a form of best first search which finds the cheapest solution through a tree or graph from the start node to the goal node. A* search combines the two approaches of *uniform-cost search*, which minimises the path so far represented by g(n), and *Greedy search*, which estimates and minimises the cost to the goal represented by h(n), to get the estimated cost of the cheapest solution through node n, represented by f(n).

Consider the simple graph shown in Figure 8.1 in which five towns A, B, C, D and E are linked by roads labelled with the distance between them. A person is in Town A and wishes to reach Town E and they have a choice to go through B, C or D. travelling through Town C is the shortest route and travelling through Town D is the longest journey. Table 8.1 shows the values calculated by the three search heuristics.

Using uniform-cost search (g(n)) the person wants to minimise the distance from

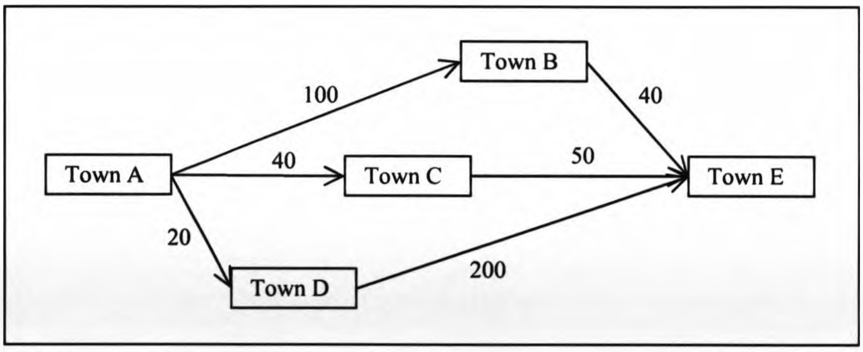

Figure 8.1: Example to show A* search.

Table 8.1: The values for the example.

	Greedy Search	*Uniform-Cost Search*	*A* Search*
	h(n)	g(n)	f(n)
Town B	40	100	140
Town C	50	40	90
Town D	200	20	220

town A to the next town. This means they will travel to town D because it is the shortest distance travelled even though it will prove to be the longest journey in total.

Using Greedy search (h(n)) the person wants to minimises the distance from the next town to the goal. This means they will travel to town B because the distance to town E is less than that from towns C and D. This is still not the shortest route.

When the two values are combined in A* search (f(n)) the person will travel through town C and this has the smallest cost to travel to town E. This is the optimal solution because it is the shortest distance between A and E.

8.2 AGATHA's Version of A* Search

A* is not an adversarial search, and so using it in AGATHA makes theory construction a cooperative process as no account is taken of how good a response a move permits, and so involves no notion of blocking the "opponent's" best moves. I will explore the effect of a genuinely adversarial search in the next chapter. Standard A* search uses two parameters which must be adapted because I do not have any real target: I want only to produce the best possible theory and I will not necessarily know when I have got there. Also it is unimportant in this context how many moves are required to produce it. For h(n), which estimates the cost to reach the goal from the next node, I subtract the *Evaluation Number* (which has been calculated by ETHEL) for the next Theory from the *Evaluation Number* for the *Ultimate Theory* which is the ideal theory and cannot be improved and for g(n), which represents the actual cost of reaching the current node from the initial state, I use only the cost of the next move from the current theory and do not consider the history of how I reached the node.

The A* value f(n) for each theory is now given by summing the h(n) and g(n) values and only the theories with the lowest A* value are expanded. Before A* starts to work out which move to make, the new theories are checked to ensure that they have a larger *Evaluation Number* than the original theory. This is to ensure that is the new theory represents an improvement.

8.2.1 h(n) Values

To replace the notion of a goal state, I calculate the h(n) value by calculating how similar the next theories are to the best theory imaginable. This is done by calculating the *Evaluation Number* for the *Ultimate Theory*, which consists of one rule and gets all the cases correct for both the *Explanatory Power* section and the *Completion Explanatory Power* section, so that its basic *Evaluation Number* is maximum. A complete tree with five levels would thus result in an *Ultimate Evaluation Number* of 420. (Simplicity value = 100, Explanatory Power = 100, Completion Explanatory Power = 100, and the depth will increase the total by 40%). The hope is to eventually reach the *Ultimate Theory* (or very close to it) by progressively improving the theory by making new argument moves.

Now an h(n) value for each theory can be calculated by subtracting the *Evaluation Number* for the theory from the *Ultimate Evaluation Number*. This means that a "good" theory will have a smaller h(n) value than a "bad" theory because it will be more similar to the *Ultimate Theory* (and have fewer rules and/or decides more of the background cases correctly) than the "bad" theory and as I want to choose the best theories possible I want to choose the smallest h(n) value.

8.2.2 g(n) Values for each Theory Move

The g(n) value is given by the cost of making the move to get to the next theory. Each move defined in Chapter 6 is ranked according to our view of its desirability and associated with a cost. The moves are ranked as follows: *Counter with Case, Distinguish with Case, Distinguish Problem, Distinguish with Arbitrary Preference. Analogise Case* is given the highest value as I want it to be made only at the beginning, otherwise the dialogue would effectively restart and *Counter with Case* is given the lowest as this is the move that I feel is most desirable. The g(n) values for each move are given in Table 8.2. These are intended to reflect my view of which moves would be seen as most powerful by human players.

Table 8.2: g(n) values for each move.

Move Name	*g(n)*
Counter with Case	50
Distinguish with Case	100
Distinguish Problem	150
Distinguish with Arbitrary Preference	200
Analogise case	250

The A* search value for each theory is now given by summing the h(n) and g(n) values and only the theories with the lowest A* value are expanded. This may mean that the "best" theory may not be reached because that theory may be produced by using a move with a high cost whereas a less good theory which is produced by a move with a low cost will be choosen instead.

8.2.3 Wild Animal Example

Figure 8.2 shows the theory tree produced when AGATHA is not using a heuristic. Table 8.3 shows the Evaluation Number calculated for each theory, the h(n) value, the g(n) value and the f(n) value for each theory.

From Theory 0 AGATHA constructs and evaluates Theories 1 and 2. Theory 1 does not contain a rule preference and so has a lower *Evaluation Number* than Theory 2 which has a single rule preference. This means that when the *Evaluation Number* for the theories is subtracted from the *Evaluation Number* for the *Ultimate Theory*, Theory 2 will have a lower h(n) value. Because both theories are constructed by using the *Analogise Case* move the g(n) value for both theories is the same. This means that

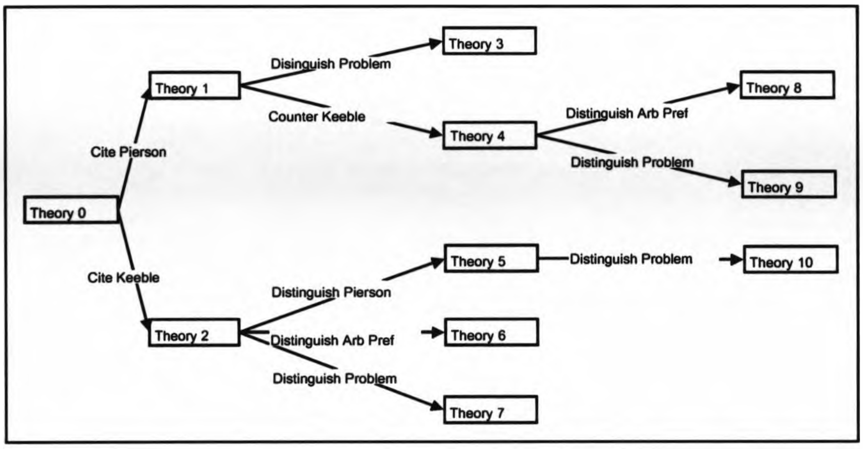

Figure 8.2: Example to show A* search.

Table 8.3: Wild Animal theory tree to show A* working.

	Evaluation Power	*h(n)*	*g(n)*	*f(n)*
Theory 1	167	253	250	503
Theory 2	267	153	250	403
Theory 3	250	170	150	320
Theory 4	294	126	50	176
Theory 5	330	90	100	190
Theory 6	264	156	200	356
Theory 7	297	123	150	273
Theory 8	288	132	200	332
Theory 9	324	96	150	246
Theory 10	285	135	150	285

when the f(n) value is calculated Theory 2 will get the lowest value and will be kept and Theory 1 is discarded.

Because Theory 1 is not retained, Theories 3, 4, 8 and 9 will not be constructed and this has already reduced the number of theories that AGATHA has to construct and evaluate by four.

From Theory 2 AGATHA constructs and evaluates Theories 5, 6 and 7. Theories 6 and 7 have two rule preferences which reduces their *Evaluation Number* compared to Theory 5 which only has one rule. The *Evaluation Number* for Theories 6 and 7 is further reduced because Theory 6 has an arbitrary rule preference and Theory 7 has a value-based rule preference. This gives Theory 5 the lowest h(n) value.

Because the theories are constructed using different moves the g(n) value for each theory is different. Theory 5 used the *Counter with Case* move and has the lowest value, Theory 7 used the *Distinguish Problem* move and has the next lowest value and Theory 6 used the *Distinguish with Arbitrary Preference* move and has the largest value.

When the f(n) value is calculated for each theory, Theory 5 obtains the lowest value and so will be retained whilst Theories 6 and 7 will be discarded.

From Theory 5 AGATHA constructs and evaluates Theory 10. Because it is the only theory constructed AGATHA would usually keep the theory and move onto the next theory. However the final constraint comes into play which ensures that every constructed theory represents an improvement. The *Evaluation Number* for Theory 10 is smaller than the *Evaluation Number* for Theory 5 and so constructing Theory 10 produces a worse theory and hence does not represent a good move.

The theories produced by AGATHA were all analysed in Chapter 7 to show how ETHEL evaluated theories. In that analysis I showed that Theory 5 is the best theory produced. It only has one rule preference so it has a high score for *Simplicity* and it gets all the cases correct so has a high score for both *Explanatory Power* and *Completion Explanatory Power*.

8.3 User Selected Backgrounds in US Trade Secret Misappropriation Law Domain

In the Wild Animal Example brute force is possible but as I have shown the Trade Secret Misappropriation domain requires the search space to be constrained.

A number of experiments were conducted to explore a series of questions by using various different combinations of parameters. The background cases comprise thirty two cases taken from various writings on CATO, in particular [1], [15] and [3]. The overall measure of success will be the number of cases that can be explained by a theory. Comparison targets are suggested by Table 1 in [15]. In that paper ten techniques were tested on 187 cases. The best performer was the algorithm of Ashley and Brüninghaus themselves with 170 right, 15 wrong and one abstention for an accuracy of 91.4%. Next best was Naïve Bayes with an accuracy of 86.5%. No other technique did better than 77.8%. As I am restricted to the 32 cases I have been able to reconstruct from the published literature, 30+ correct classifications would represent a performance comparable to IBP and 28-29 correct classifications a performance comparable to Naïve Bayes, and 26 cases a performance better than any other technique considered in [15].

8.3.1 Comparison with Complete Space and A*

Table 8.4: Original Backgrounds

	Plaintiff Cases	*Defendant Cases*
Background 1	Goldberg National Instrument	Ecologix Sandlin
Background 2	Space Aero	National Rejectors
Background 3	Trandes	CMI

The backgrounds used were first described in Chapter 6 and are shown again in Table 8.4. Background 1 uses 4 cases, Background 2 has the 4 cases from Background 1 and an extra two cases and Background 3 extends Background 2 by another 2 cases. These backgrounds were chosen to try to explain the *Mason* case and may not generalise to other seed cases very well.

Tables 8.5, 8.6 and 8.7 show the results for all 3 backgrounds when *Mason*, *Digital Development* and *CMI* are used as the seed case. Tables E.1, E.2 and E.3 in Appendix E show the results for backgrounds 1, 2 and 3 respectively when every case is used as a seed case.

Table 8.5: A* AGATHA Results for Mason when using the three backgrounds.

Name	Background	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
A*Mason1	1	8	2	27	28
A*Mason2	2	27	5	29	28
A*Mason3	3	36	5	29	29

Table 8.6: A* AGATHA Results for Digital Development when using the three backgrounds.

Name	Background	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
A*DigDev1	1	6	2	19	28
A*DigDev2	2	7	2	19	28
A*DigDev3	3	8	2	19	28

Table 8.7: A* AGATHA Results for CMI when using the first two backgrounds.

Name	Background	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
A*CMI1	1	2	2	10	24
A*CMI2	2	4	2	10	24

The main improvement from using A* is the substantial reduction in the number of theories created (from 106 to 8 theories or 653 to 27 theories for *Mason*) with only a small reduction in the ability of the theory to decide cases correctly. My hope is that the ability to include more cases from which to select moves resulting from the pruned search space will more than compensate for missing the "best" theory from a limited background. Moreover since the selection of cases depends on the seed case, selection would be difficult to automate since human skill and judgement is needed to find cases appropriate to the seed. Note also that the performance is good in all cases: A* attaining the level of Naïve Bayes, and the exhaustive search the level of IBP when at least six cases are used.

As is to be expected A* performs worse than the original AGATHA program but this is explained by the fact that A* does not use all the moves available, but chooses the best move which *improves* the theory. A* will only make the move if the next theory is better than the previous theory, whereas AGATHA originally did not impose this condition, and so sometimes there is a theory with a lower *Evaluation Number* than its previous theory which can subsequently be modified to produce a better following theory. This is not possible within the spirit of adversarial search, since there is no obligation to make a move unless a better theory has been proposed. So, even here the performance of A* is of acceptable quality.

Figures 8.3 and 8.4 show the tree constructed by AGATHA when the case of *Digital Development* is used as the seed case with backgrounds 1 and 2.

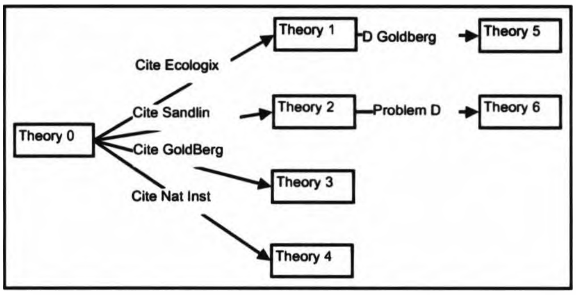

Figure 8.3: Theory Tree produced when Digital Development is the seed case and the first background is used with the A* search heuristic.

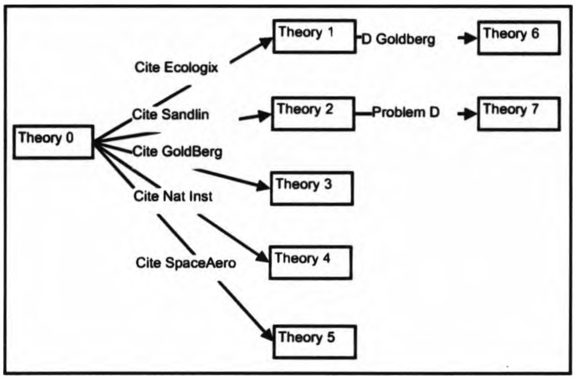

Figure 8.4: Theory Tree produced when Digital Development is the seed case and the second background is used with the A* search heuristic.

Figures 8.5 and 8.6 show the tree constructed by AGATHA when the case of *CMI* is used as the seed case with backgrounds 1 and 2. These are the exact same trees produced as for when there is no heuristic used. Because CMI does not have many factors in common with the background cases, AGATHA cannot produce very good theories and when the heuristic is used there is no need to search the theory space.

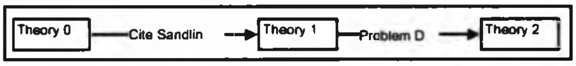

Figure 8.5: Theory Tree produced when CMI is the seed case and the first background is used with the A* search heuristic.

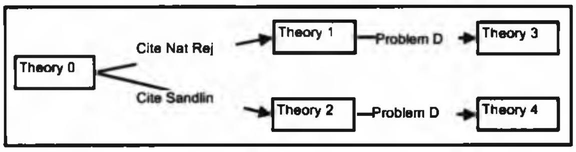

Figure 8.6: Theory Tree produced when CMI is the seed case and the second background is used with the A* search heuristic.

8.4 Results When Using the Entire Case Background

It is not desirable for the cases to be selected by a human user: I want AGATHA itself to select the best cases to cite from the whole background set. Using all the cases, however, is not viable without using a search heuristic because the search space is too large and this is where the A* search heuristic is useful.

As was seen with user defined backgrounds the use of the search heuristic limited the number of theories expanded but did not reduce the power of the theories to explain cases by much. The following sections explain various experiments that were performed using the A* search heuristic to see how it performs and also to find the correct parameters for ETHEL to evaluate the theories.

8.4.1 Is it better to use the Most-On-Point cases or all the cases?

HYPO and CATO use the Most-On-Point cases because they are concerned with only one case and creating an argument to explain this one case. Using Most-On-Point cases can limit the moves available to the opponent and prevent counter attacks. AGATHA is different in that it is trying to explain all the background cases not just one case and so it may be that a less on point case produces a better theory which can explain more of the background cases.

Using only Most-On-Point cases means that AGATHA can only use some of the cases from the background as determined by the seed case. This limits AGATHA to a small subset of the background cases, although these will be different for different seed cases. Although this means that intuitively only the most pertinent cases are used, these cases are only most pertinent for the seed case and these cases may be unsuitable to decide the remaining cases from the case background. As part of the evaluation of the theory depends on how well the theory can generalise to other cases this limitation of cases may be undesirable. Therefore AGATHA was initially run using the limitation of Most-On-Point cases with the results shown in table E.4 in Appendix E and then was modified to use all the background cases with the results shown in table E.5 also in Appendix E.

Table 8.8 shows the results obtained when AGATHA can only use the Most-On-Point cases from the background for selected cases and Table 8.9 shows the results obtained when all of the background cases are used for the same selected cases. Using the Most-On-Point cases AGATHA performs worse than when using the 3 backgrounds defined earlier. This shows that using the Most-On-Point background cases may not be the best idea for this work.

When AGATHA was able to use all the background cases the number of theories increased for all the seed cases. For three of the cases (*Arco, Ferranti and Sandlin*) the increase was not large but in all of the other seed cases the increase is usually large and sometimes very large. Due to the increase in the number of theories the depth of the trees produced also increases for most of the seed cases. Four cases did not increase their depth and these were the cases that did not produce many more theories. Three of the cases had a very large increase in the depth of the tree. *Laser, Mason* and *Technicon* all produced many more theories and hence a tree with many more levels. Even so the size of the tree stayed within reasonable bounds - smaller than complete search on the background of 4 Cases.

The results when using all the background cases are never worse than when using only the Most-On-Point cases and are usually better.

Ecologix, Technicon and *Televation* were the best performing seed cases when using Most-On-Point cases from the background but they do not improve by much when

Table 8.8: A* AGATHA Results selected seed cases when AGATHA is limited to using only the Most-On-Point background cases for each seed case. Selected from table (E4).

Project Name	Number of Nodes	Tree Depth	Best Results Explanatory	Best Results Completion
Mason	49	4	25	26
DigDev	5	2	18	18
CMI	11	2	23	26
Ecologix	38	2	27	26
MBL	11	2	23	26
Technicon	102	9	28	25
Televation	12	4	28	24

Table 8.9: A* AGATHA Results for the cases when AGATHA can use all the background cases. Selected from table (E5).

Project Name	Number of Nodes	Tree Depth	Best Results Explanatory	Best Results Completion
Mason	863	12	27	28
DigDev	36	3	26	27
CMI	301	6	28	26
Ecologix	85	2	27	26
MBL	2950	9	28	27
Technicon	1007	14	28	28
Televation	130	6	28	28

all the background cases can be used. *Laser* and *Valco-Cincinnati* are the best performing seed cases when all the basckground cases can be used. These two cases improved their results by 8 cases for the Explanatory result and 5 (or 6) cases for the Completion result. This improvement means that they outperform the three best cases from the Most-On-Point cases from the background.

Even though the number of theories increases when using all the cases in the background the improvement to the results is worth it.

8.4.2 Is the depth contribution important?

To see how important considering depth is I ran several experiments and varied the contribution of depth to the Evaluation Number from 0 to 50%. For the earlier experiments the contribution of depth was fixed at 10% but for Table 8.10 it ranges from 0 to 50%.

For example, when comparing the experiments, which are using *Mason* as the seed

Table 8.10: A* AGATHA Results for Mason when the Depth Contribution is varied.

Project Name	Depth Contribution	Number of Nodes	Tree Depth	Best Results	
				Explanatory	Completion
depth1	0	118	4	27	28
depth2	10	863	12	27	28
depth3	25	1439	18	28	29
depth4	50	4038	20	28	29

case (depth1 to depth4 in Table 8.10), I find there is no improvement in the number of cases decided correctly after the depth factor reaches a value of 25% of the evaluation power.

Increasing the contribution of depth means that it is easier for AGATHA to use the *Distinguish with Arbitrary Preference* move because the depth value helps to counteract the large g(n) value.

Although increasing the depth contribution means that more moves could be made to the theories and so allow greater exploration of the space it appears that this effects little real improvement in the quality of the theories.

An additional problem with having a deeper tree means that the theories may become over fitted to the cases used and hence do not generalise very well.

8.4.3 Is the cost of the moves correct? Modifying the g(n) values for *Counter with Case* and *Distinguish with Case* Moves

In all the previous experiments the different moves are ranked with a different cost for each move, as shown in Table 8.2. However I wanted to test the hypothesis that the *Distinguish with Case* move is at least as desirable as the *Counter with Case* move. Table E.6 shows the results when the moves *Counter with Case* and *Distinguish with Case* are given the same cost of 50. Table 8.11 shows the selected cases from earlier.

With equal weights the *Distinguish with Case* move is made more often, and the performance is broadly similar. When comparing the selected cases in Table 8.9 with the same cases in Table 8.11, *Mason* and *Technicon* produce a tree with fewer nodes and fewer levels and have an improvement in results. *Valco-Cincinnati* produces more nodes but has no improvement. Using the same weight does not improve or degrade the performance of the theories.

Table 8.11: A* AGATHA Results for selected cases when the values for *Counter* and *Distinguish with Case* moves are the same.

Project Name	Number of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Mason	419	9	29	29
DigDev	36	3	27	28
CMI	301	6	28	27
Ecologix	85	2	27	26
MBL	3124	9	28	27
Technicon	304	5	28	28
Televation	85	3	28	28

8.4.4 Summary of Results

From these results I conclude that using all the cases is preferable to using only the Most-On-Point cases and that the contribution of depth is of some importance: 25% giving better results but almost doubling the search space. Particularly interesting is the improvement given by using cases which are not the Most-On-Point. On pointedness is important in both HYPO and CATO, and using such cases has a tactical point in that they are the least open to distinction. On the other hand, using portions of precedents has also long had its advocates (e.g. [13]). Deciding a case often involves considering a number of sub issues and it may well be that a precedent is very relevant for one of these sub issues, although otherwise very dissimilar from the case under consideration. To include such cases starting from a given seed, therefore, we need to go beyond the set of cases most on point to the seed case. Whether on pointedness becomes more useful with adversarial search is something we shall consider in future work. Issue based selection of cases is also a feature of IBP [15]. Using different weights for the *Distinguish with Case* and *Counter* move does not seem to matter.

8.5 Use of Other Cases as the Problem Case

In the next set of experiments I wanted to explore the use of cases with particular features as the seed case. For all of the experiments in this section AGATHA used all the cases, a depth factor of 10% and different weights for the *Distinguish with Case* and *Counter* moves.

I chose a range of different cases to test various classes of case: these cases are shown in Table 8.12 and the results of each experiment are shown in Table 8.13.

Table 8.12: Cases and the types and numbers of factors which describe them

Case	Number of Factors	Plaintiff	Defendant	Case Outcome
Sandlin	5	0	5	D
Ferranti	5	1	4	D
Reinforced	6	5	1	P
Boeing	7	5	2	P
Technicon	7	4	3	P
CMI	7	2	5	D
CollegeWatercolor	3	2	1	P
Sheets	3	1	2	D

Table 8.13: Results when using different problem cases different weights.

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Sandlin	10	1	11	23
Ferranti	20	2	9	23
Reinforced	52	3	20	27
Boeing	96	3	28	28
Technicon	1007	14	28	28
CMI	301	6	28	26
CollegeWatercolor	116	5	24	24
Sheets	230	5	25	28

The first experiment used the case of *Sandlin* to see what happens when the problem case only has one type of factor as *Sandlin* only has defendant factors.

Because there are only defendant factors, only the defendant player can make a move, which is why the theory tree only has 10 theories and a depth of 1. The Plaintiff player cannot make a move because there are no plaintiff factors to use.

The theories constructed do not perform very well as they only get 11 cases correct out of 32 for the *Explanatory Power* and 23 correct out of 32 for the *Completion Explanatory Power*. This shows that AGATHA can only perform effectively when there are factors from both sides present in the seed case.

There are no cases in the background with only Plaintiff factors so we could not perform the reciprocal experiment. Instead we chose a Defendant case with only one Plaintiff factor and a Plaintiff Case with one defendant factor.

For the Defendant case we used *Ferranti*, which has one Plaintiff factor and four

Defendant factors. The theory tree has more theories and goes to an extra level compared to *Sandlin*. However it performs much worse compared to using *Sandlin* as it only gets 9 cases correct out of 32 for the *Explanatory Power*. This low number arises from the very high number of abstentions each theory makes which is why the results improve when all the factors are included in the theory.

I used *Reinforced-Moulding* as the reciprocal Plaintiff case as it has five Plaintiff factors and only one Defendant factor. The theory tree again has more theories and reaches a depth of 3. It also gets much better results with 20 cases correct out of 32 for the *Explanatory Power* and 27 correct out of 32 for the *Completion Explanatory Power*.

These three experiments show that having factors of both types present in the problem case means that AGATHA can use the Argument Moves effectively to improve the theories. If the seed case contains few factors, completion of the theory seems essential.

To explore this point further we investigated whether the number of factors in the case description is important. To show this I split the experiment in two and chose cases with the most number of factors and cases with the smallest number of factors.

For the large cases I chose *Boeing*, which is a Plaintiff case with five Plaintiff factors and two Defendant factors, *Technicon*, which is a Plaintiff case with four Plaintiff factors and three Defendant factors, and finally *CMI*, which is a Defendant case with two Plaintiff factors and five Defendant factors. I chose three cases because I wanted to compare a very Plaintiff case, a very Defendant case and a balanced case.

When the experiments were run the larger cases improved over the first set of experiments on strongly biased cases. Of the three experiments, using *Technicon* as the problem case performed the best as it got 28 cases correct out of 32 for both the *Explanatory Power* and the *Completion Explanatory Power*, the best combined result obtained during our series of experiments. However the theory tree is large, containing over 1000 theories and reaching a depth of 14. Still this is a great improvement on complete search and not "too" large. Other than *CMI*, completing the theory by including all the factors gives no improvement to these three cases.

This suggests that a more balanced problem case will produce a larger theory tree and obtain better results than if the case is biased towards one of the parties, especially with respect to the uncompleted theory.

For the experiments with the smallest cases, I chose *College Watercolor*, which is a Plaintiff case with two Plaintiff factors and one Defendant factor and *Sheets*, which is a Defendant Case with one Plaintiff factor and two Defendant factors. When the experiments were run, *Sheets* performed better because it obtained 25 cases correct out of 32 for the *Explanatory Power* and 28 correct out of 32 for the *Completion Explanatory Power* compared to 24 correct for both the *Explanatory Power* and the *Completion Explanatory Power*, even though the theory tree for *Sheets* has fewer theories and two fewer levels. When comparing the small cases with the large cases, the larger cases perform better, and, for small cases, completion gives improvement.

When comparing all of these experiments, the experiments with very biased problem cases perform worst whereas the experiments with well-balanced problem cases perform best. For the question of the size of the cases, the problems cases with the most factors perform better than those with fewer factors.

These experiments also show that the tree must go to at least the third level to get good results. This seems to correspond to the 3-ply arguments of HYPO and CATO.

Overall I would conclude that the best way to generate a theory automatically would be to select as seed the most balanced background case with the most factors, and use A* with all cases and a depth factor of 10%. This technique, represented by the entry for *Technicon* in Table 8.13, produces a theory which gives performance at a level similar to that of IBP.

8.6 Conclusions

Some search heuristic is necessary if AGATHA is to make full use of available background cases and the use of A* shows that the ability to use a more extensive background does improve the results for AGATHA. Moreover AGATHA produces better theories than the hand constructed theories reported in [18] and section 4.3, and theories comparable in explanatory power to the best performing reported technique, IBP [15], [3]. Note also that AGATHA can be used even when there is no accepted structural model of the domain, whereas IBP relies on using the structure provided by the Restatement of Torts.

From the results I conclude that using all the cases is preferable to using only the Most-On-Point cases and that a depth factor of 10% gives good results. Giving moves different costs produces a bigger tree, but typically does not produce more explanatory theories. Since the theories are not perfect, it might be possible to improve the evaluation used during the search by tuning the parameters. None the less I regard the performance as sufficient to indicate that the parameters and criteria used are at least in the right area.

I have also discovered that the best seed case to use is one with a large number of factors and where these factors are divided equally between Plaintiff and Defendant. This case can, of course, be identified automatically, and so AGATHA can be used to construct a theory from a given background without manual guidance. The ideal case background is one where there is a mix of Plaintiff and Defendant cases, the cases must have a reasonable number of factors and have factors present that promote both outcomes.

I find the results reported here highly encouraging: they provide some support for the theoretical account of reasoning with cases in terms of theories which use factors and values proposed in [10]. Moreover they suggest that the process of theory construction may be open to automation, once the domain analysis required to produce the background has been carried out.

Chapter 9

Alpha Beta Search Heuristic

The first heuristic implemented in AGATHA and described in Chapter 8 is a co-operative heuristic based on A* search, where the two agents are co-operating to produce the "best" theory possible and it does not matter who the winner is. Legal argumentation is, however, an adversarial process where the two lawyers each want to win but they also want to prevent the other lawyer from winning. To model this behaviour a second heuristic was implemented based on the $\alpha\beta$ pruning heuristic. This means that the agent will sometimes seek to avoid powerful moves from the opponent at the expense of following its own best line.

9.1 Explanation of $\alpha\beta$ Pruning

$\alpha\beta$ pruning, described in many textbooks including [40], [35] and [31], is widely used in chess and other two player games so that it is an obvious choice to implement next. $\alpha\beta$ pruning is a search heuristic that reduces the number of nodes that need to be evaluated in the game tree by the MINIMAX algorithm. MINIMAX is used to choose the next move in a two player game. MAX wants to maximise the utility he receives from the game and MIN (the opponent) wants to minimise the utility received.

Figure 9.1 shows a game tree generated by the MINIMAX algorithm. MAX can make moves A1 to A3 and MIN can reply by making moves B1 to B9. MAX has to generate the entire game tree to discover the value for each terminal node. If MAX uses move A1 the utility he will receive will be 3 because MIN will choose to use move B1 to minimise the utility. If MAX uses moves A2 or A3 he will only receive 2. Hence

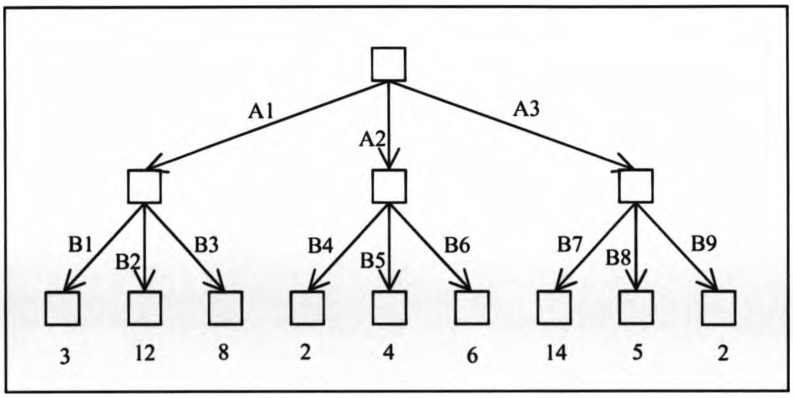

Figure 9.1: Example problem to show the working of MINIMAX and $\alpha\beta$ Pruning.

the best move for MAX is A1 and MIN's best reply is B1. Note that although MAX could possibly achieve 14 by using move A3, MIN will not play B7, but rather B9 thus giving a payoff for A3 of only 2.

It is very costly to create the entire tree both in terms of time and memory so a way has to be found to limit the branches of the tree that have to be searched. $\alpha\beta$ pruning stops evaluating a move when a reply has been found that proves the move to be worse than a previously examined move.

Still using the tree from Figure 9.1, MAX starts by generating the subtree produced by using move A1. The best value he can expect to receive is 3. Next he starts to generate the subtree produced by using A2. When he has created the first branch, which is produced when MIN uses move B4, MAX finds that the utility for this node is less than he expects to get from performing A1 and so he resolves not to perform A2 and prunes this subtree from the graph. Next he starts to generate the subtree produced by using move A3. For this move he has to generate the entire subtree to find out that he will only receive 2 if he performs the move. Hence the best move for MAX is again A1 and the best reply for MIN is B1.

The effectiveness of $\alpha\beta$ pruning is affected by the ordering of the nodes in the subtree. For the A2 move subtree MAX only had to generate one node but for the A3 move he had to generate the entire subtree to discover it was a worse move than A1. The effectiveness could be improved by trying to sort the moves and perform them in an order to aid the $\alpha\beta$ pruning.

9.2 Example of $\alpha\beta$ Pruning used by AGATHA

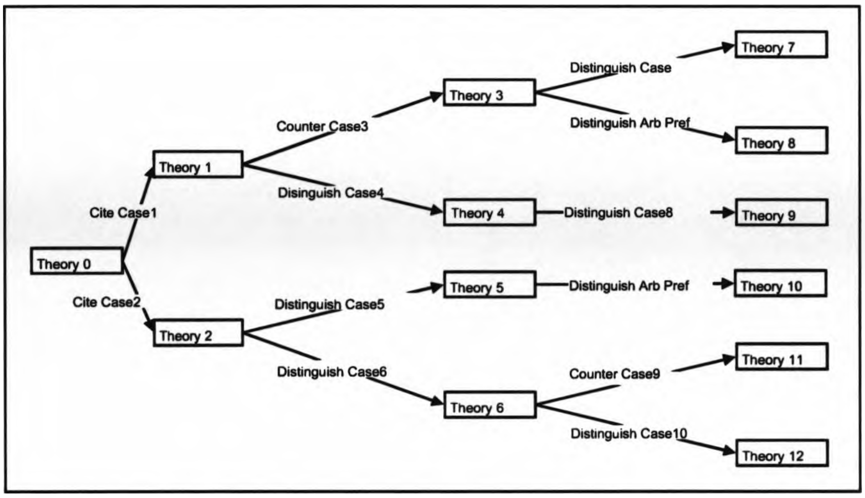

Figure 9.2: Example Theory Tree to show the working of $\alpha\beta$ Pruning in AGATHA.

In modelling this adversarial dialogue, AGATHA applies a 3-ply method on each theory that has been created. The moves are ranked in the same order as in section 8.2.2 and performed in this order to try to ensure that the "best" theory is produced and compared to the other theories. Figure 9.2 shows an example of an entire theory tree produced by AGATHA. AGATHA first applies all the moves to the theory that are possible to create a group of 1-ply theories. These 1-ply theories represent all the moves that the player can make in the current situation and AGATHA has to choose which of them to play. In the example, the 1-ply theories are Theories 1 and 2.

AGATHA takes the first 1-ply theory (Theory 1) and expands it by only one move to create Theory 3. This move represents how the opponent could respond to the theory. AGATHA then expands this 2-ply theory by one move to give the 3-ply theory (Theory 7) which the current player could make in response to the opponent's theory. AGATHA then assesses the 3-ply theory using the ETHEL program and stores the 3-ply theory.

AGATHA then takes the next 1-ply theory (Theory 2) and expands it to the 3-ply theory, creating Theories 5 and 10. The *Evaluation Number* for the 3-ply theory is calculated by ETHEL and compared to the first *Evaluation Number*. If the new theory is better (eg. if Theory 10 has a higher *Evaluation Number* than Theory 7) then the 3-ply theory is stored otherwise it is discarded.

AGATHA continues in this way until there are no more 1-ply theories left. It then finds the "best" 3-ply theory and the grandparent 1-ply theory of this theory will represent the best move for the player to make. Ties are broken using the 2-ply theories: the worst of these is chosen, to restrict the opponent's opportunities. In the example, if the *Evaluation Number* for Theory 7 is larger than for Theory 10 then Theory 1 represents the "best" move otherwise Theory 2 represents the "best" move. If the *Evaluation Numbers* are the same then the tie is broken by comparing the 2-ply Theories 3 and 5. If the *Evaluation Number* for Theory 3 is lower than that for Theory 5 then Theory 3 is a "worse" theory and so Theory 1 represents the best move to make.

When the search tree is exhausted and the dialogue ends, AGATHA finds the best Plaintiff theory and the best Defendant theory for each branch in the pruned tree.

Using this search heuristic enables AGATHA to ignore branches that result in less good theories. Ordering the moves enables AGATHA to just construct theories along a single branch because of the idea that certain moves create "better" theories than others.

This search heuristic makes assumptions on what the opponent will do if it is reasoning rationally. The problem is that AGATHA only looks three moves ahead and the opponent is also using the 3-ply method to decide what to do. Thus the opponent will consider a level deeper in the tree and may, as a consequence, choose a move different from anticipated.

9.3 Results for $\alpha\beta$ Pruning

For the following experiments AGATHA is initially restricted to the three backgrounds described in Chapters 6 and 8 to enable a comparison to be performed. Then the results for when AGATHA is restricted to using the Most-On-Point background cases and when AGATHA can use the entire background are compared.

9.3.1 Comparison with Complete Space and A* results using user defined backgrounds

Tables 9.1, 9.2 and 9.3 show the results for *Mason, Digital Development* and *CMI* when the adversarial version of AGATHA is restricted to the 4, 6 and 8 cases that were used in Chapters 6 and 8 in the original and A* versions of AGATHA. The Adversarial version of AGATHA explores more theory nodes and expands the tree to a greater depth than the A* version but expands fewer theory nodes than the original version. The Adversarial version obtains better results than A* and the results are almost as good as the original version. The adversarial version follows the branch to the terminal node and then finds the best theories but it does not branch as much as A* search.

Table 9.1: $\alpha\beta$ AGATHA Results for Mason when using the three backgrounds

Name	Background	Nodes	Depth	Plaintiff Results		Defendant Results	
				Explan	Comp	Explan	Comp
Mason1	1	19	5	27	28	27	26
Mason2	2	39	7	30	30	29	28
Mason3	3	60	9	29	30	30	30

Table 9.2: $\alpha\beta$ AGATHA Results for *Digital Development* when using the three backgrounds

Name	Background	Nodes	Depth	Plaintiff Results		Defendant Results	
				Explan	Comp	Explan	Comp
DigDe1	1	10	4	28	28	13	21
DigDe2	2	11	4	28	28	13	21
DigDe3	3	23	4	28	28	19	24

Table 9.3: $\alpha\beta$ AGATHA Results for *CMI* when using the first two backgrounds

Name	Background	Nodes	Depth	Plaintiff Results		Defendant Results	
				Explan	Comp	Explan	Comp
CMI1	1	2	2	16	15	10	24
CMI2	2	4	2	16	15	10	24

The Theory Trees shown in Figures 9.3 to 9.6 also show that $\alpha\beta$ pruning does not allow the tree to branch but follows the branch to the terminal nodes.

9.3.2 Using All Cases

Table 9.4 shows the results obtained when AGATHA is restricted to only using the Most-On-Point cases from the background and Table 9.5 when AGATHA is allowed to use all the cases in the background. The full set of results for all the cases are in Tables F.4 and F.5 in Appendix F. AGATHA obtains better results when it is allowed to use all the background cases instead of just the Most-On-Point cases. This is the same

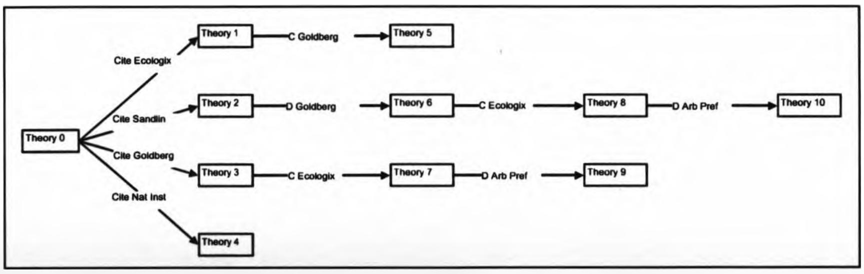

Figure 9.3: Theory Tree produced when Digital Development is the seed case and the first background is used with the $\alpha\beta$ search heuristic.

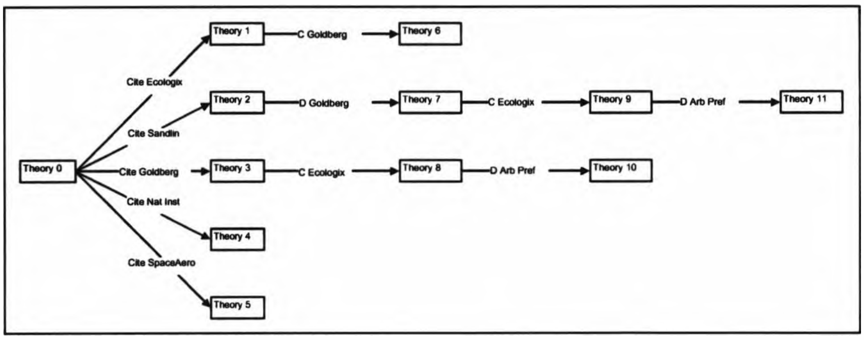

Figure 9.4: Theory Tree produced when Digital Development is the seed case and the second background is used with the $\alpha\beta$ search heuristic.

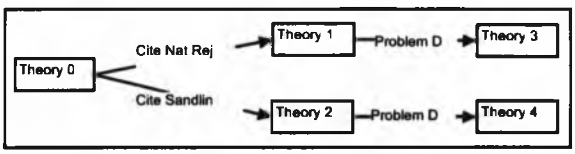

Figure 9.5: Theory Tree produced when CMI is the seed case and the first background is used with the $\alpha\beta$ search heuristic.

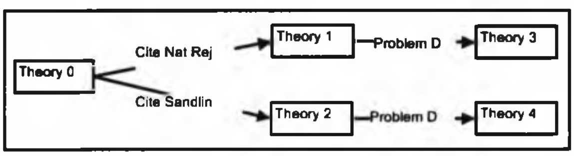

Figure 9.6: Theory Tree produced when CMI is the seed case and the second background is used with the $\alpha\beta$ search heuristic.

result as for the A* version and seems to show that to obtain a good explanatory theory AGATHA needs to use all of the background cases in whichever version is being used.

Table 9.4: $\alpha\beta$ Results for selected seed cases when AGATHA is limited to using only the Most-On-Point background cases for each seed case.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Mason	98	12	27	28	27	26
DigitalDevelopment	5	2	22	27	8	21
CMI	50	14	27	25	29	25
Ecologix	52	24	27	26	27	26
MBL	28	6	20	18	24	26
Technicon	47	10	28	25	27	24
Televation	17	6	28	25	28	24

Table 9.5: $\alpha\beta$ Results for selected cases when AGATHA can use all the background cases.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Mason	731	38	29	29	29	29
DigitalDevelopment	65	7	27	28	14	22
CMI	427	29	28	25	28	26
Ecologix	69	24	27	26	27	26
MBL	198	28	28	25	28	27
Technicon	439	20	29	29	28	26

When both players can use the full background AGATHA produces a deeper tree than when using A* search but visits far fewer nodes, e.g. for *Mason*, 731 rather than 2427 when using all the cases, and obtains a similar quality of results. So when both sides are fully informed adversarial search improves efficiency without degradation of performance.

When the theories are analysed the player that wins is usually the one that won the case originally.

9.4 Conclusions

When comparing the results for the user defined backgrounds both A* and $\alpha\beta$ Pruning produce fewer theories than the complete version of AGATHA. A* creates more branches but does not search the tree to a large depth. $\alpha\beta$ Pruning follows one branch all the way to the terminal node and then finds the "best" theory.

When AGATHA is able to use the entire background the search heuristics are necessary to limit the number of theories created and to limit the amount of searching that has to be performed in the theory tree. From the results I conclude that it is better to use all the background cases rather than restrict to the Most-On-Point cases for both of the search heuristics. Other programs like CATO and IBP use the Most-On-Point cases but they are trying to explain a single case and to prevent damaging counter moves. AGATHA is trying to explain the entire case background and it seems that limiting the cases that can be used to the ones that are Most-On-Point to a particular seed case is not the way to ensure that the theory can generalise.

$\alpha\beta$ Pruning will usually produce more theories because it explores the tree to a greater depth than A*, unless A* produces a tree with a large amount of branching where $\alpha\beta$ pruning will create fewer theories. This is the case for *Technicon*: A* creates 1007 theories and explores to a depth of 14 whereas $\alpha\beta$ pruning creates less than half that number (439) and goes 6 levels deeper into the tree. $\alpha\beta$ pruning also produces a "better" result showing that it has found a "better" theory. The possible reason for the improvement (shown by the dialogues in Chapter G) is that adversarial search forces refinement to meet "arbitrary" objections. But these refinements may be needed to meet grounded objections in the future.

From results performed in a similar way to A* and given in Appendix F I conclude that the best seed case to use is one where there are a large number of factors, equally shared between Plaintiff and Defendant. Also, if the background is restricted for any reason, the results are better when the background cases are larger with similar numbers of Plaintiff and Defendant factors.

9.5 ROSALIND - Two Player Adversarial Dialogue

Using AGATHA, both players have access to the same complete set of precedent cases. To explore the effect of different information being available to the two parties I produced the program ROSALIND, which enables the Plaintiff and Defendant each to have their own sets of cases. In an actual situation it may well be that the opposing sides are aware of different precedents. Does this give one side an advantage and what is the effect of the quality of the theories produced?

There is little effect to the results if one player has all precedent cases and the other only the precedent cases favouring its own side. But if the information available to one side is inferior (for example, in terms of the number of factors in its precedent cases) to that of the other the final theories are significantly worse: the better equipped player does not have to produce a very high quality theory to win the game.

Table 9.6 shows all the cases from the case background sorted by size into groups. The Defendant cases are labelled to show how they are spread across the five groups.

Table 9.6: Cases sorted by size depending on the number of factors describing the cases

3 Factors	*4 Factors*	*5 Factors*	*6 Factors*	*7 Factors*
Arco (d)	Ecologix (d)	Bryce	Digital Development	Boeing
College Watercolor	Forrest	Den-Tal-Ez	FMC	CMI (d)
Emery	Goldberg	Ferranti (d)	MBL (d)	KG
Lewis	Mineral Deposits	Laser	Reinforced	National Rejectors (d)
National Instrument	Yokana (d)	Robinson (d)	Scientology (d)	Technicon
Sheets (d)		Sandlin (d)	Valco-Cincinnati	Televation
		Space Aero		
		Trandes		

In the experiments using unbalanced information I used three cases, all taken from the group with five factors. *Mason* was chosen as a balanced case, *Bryce* as a strongly pro-Plaintiff case and *Ferranti* as a strongly pro-Defendant case. For the first experiment (labelled P1) the Plaintiff is given all of the cases from the group with seven factors and the Defendant is given is given the cases from the group with only three factors. The second experiment (labelled P2) has the Plaintiff using the two groups with the largest number of factors (the cases with six and seven factors) and the Defendant gets the two groups with the smallest number of factors. The third and fourth experiments (labelled D1 and D2 respectively) are the reverse with the Defendant receiving the larger cases and the Plaintiff the smaller cases.

The results when *Mason* is the seed case are shown in Table 9.7. *Mason* produces the best theory when compared with the other seed cases because it is a well balanced case. The Plaintiff always reaches the best theory even when it is disadvantaged with the smaller cases. The theories improve when more cases are used but actually the best theory is produced when the Plaintiff player is disadvantaged and the Defendant player has the larger cases. This may be because the player with the case which won in practice is pushed harder when the opponent has better cases, and so is driven to produce a better theory to win.

Table 9.7: Results with Mason as seed when choice of case is biased by size to alternately Plaintiff and Defendant.

Name	Cases used		Nodes	Depth	Plaintiff Result		Defendant Result	
	P	D			Explan	Comp	Explan	Comp
masonP1	7	3	21	8	23	23	21	25
masonP2	6,7	3,4	148	19	27	28	26	26
masonD1	3	7	30	6	27	29	28	28
masonD2	3,4	6,7	41	8	27	29	27	27

The results when *Bryce* is used as the seed case are shown in Table 9.8. The Plaintiff player always produces the best theory even when it is disadvantaged. The Plaintiff player starts all the dialogues (except for bryceP2 where both players can start the dialogue) and the only moves that the Defendant player can make are *Distinguish with Arbitrary Preference* and *Problem Distinguish*. This is because the only Defendant factor in *Bryce* is F1 and the only Defendant case with this factor is *Ecologix* in the 4 factor group and the dialogue can only proceed past a depth of 3 when the Defendant player can use this case.

The results when *Ferranti* is used as the seed case are shown in Table 9.9. *Ferranti* produces relatively poor quality theories because the only Plaintiff factor in *Ferranti* does not appear in any of the other Plaintiff cases and so none of them can be used. The tree only has a depth of 2 because the Defendant player *Analogises* a Defendant case and then the Plaintiff player responds with the *Problem Distinguish* move. The Defendant player always gets better results even when it is disadvantaged by having the smaller cases.

Table 9.8: Results with Bryce as seed when choice of case is biased by size to alternately Plaintiff and Defendant.

Name	Cases used		Nodes	Depth	Plaintiff Result		Defendant Result	
	P	D			Explan	Comp	Explan	Comp
bryceP1	7	3	5	2	12	15	10	11
bryceP2	6,7	3,4	43	18	24	28	8	12
bryceD1	3	7	3	2	19	11	8	21
bryceD2	3,4	6,7	9	3	19	28	19	28

Table 9.9: Results with Ferranti as seed when choice of case is biased by size to alternately Plaintiff and Defendant.

Name	Cases used		Nodes	Depth	Plaintiff Result		Defendant Result	
	P	D			Explan	Comp	Explan	Comp
ferrantiP1	7	3	4	2	8	14	8	24
ferrantiP2	6,7	3,4	8	2	8	14	8	24
ferrantiD1	3	7	4	2	8	14	8	24
ferrantiD2	3,4	6,7	8	2	8	14	8	24

9.6 Conclusion

I have come to a number of conclusions regarding the use of adversarial search to construct case law theories. With regard to the quality of the theory produced.

- It is important that a balanced case be used as the seed if a general explanatory theory is required. A case which is clear for one side of the other can be explained using a relatively simple theory, which does not address some of the more subtle interactions of factors required to give a theory which explains the domain in general.

- The adversaries need as good a stock of cases as possible. While performance is not much affected if one side is unaware of the cases favouring the other side, they need to be able to make their own arguments to force their opponent to refine the theory.

- Given these two conditions, adversarial search produces theories comparable or better in performance to A*, but is more efficient in terms of nodes examined during the search.

Chapter 10

Concluding Remarks

In this dissertation I set out to show that reasonable theories which explain a body of case law could be constructed automatically. These theories follow those described by Bench-Capon and Sartor in [10] and my approach to automation is to construct the theory by means of a dialogue based on argument moves as found in Case Based Reasoning approaches such as HYPO [2] and CATO [1].

10.1 Construction of Theories

I first implemented the Theory Constructors defined by Bench-Capon and Sartor in [10] to ensure the descriptions could be realised as a practical system. I demonstrated that the backgrounds for the system could supply the required knowledge for the process of theory construction and demonstrated the system by repeating the example given in [10] where the authors created several theories by hand to show how the Theory Constructors would work. CATE (CAse Theory Editor) was used to recreate the theories. Prolog code was then automatically generated from these theories so that they could be evaluated to demonstrate how many cases each theories obtained the correct decision for. This gave an indication of how "good" the theories were at generalising to decide other cases.

The next step was to use CATE to answer several research questions by performing a series of experiments. The questions and answers are as follows.

Q1: *How should we select cases and extract rules for inclusion in the theory?*

To explore this question I created three theories:

1. a "safe" theory using the ideas of Prakken and Sartor in [30] in which the theory contains rules which do not go beyond the minimum I am entitled to infer

2. a "simple" theory which introduces the fewest possible number of factors and is willing to make assumptions to produce rules mot strictly justified by the cases

3. a "value driven" theory where we made some assumptions about how I believe the domain should operate and produced a value ordering following these assumptions

I found that the "safe" and "value" driven theories performed the best and the "simple" theory was the worst due to overfitting to the training group of cases and not being able to generalise to the test group of cases. None of the theories generalise particularly well, failing to give the correct decision in at least three, and typically more of the nine cases. The cases misclassified varied somewhat for the different theories.

Q2: *Should we be inclusive or exclusive with regard to factors?*

I executed the above theories with only the factors included that were used to construct the theories and then repeated the execution with all the factor background loaded into the theories.

When all the factors were included all the abstentions were eliminated and the results for the "safe" and "simple" theories suggest that I should include all the factors.

Q3: *Is there evidence to suggest that values can be used to determine the relative importance of factors?*

The benefits of including all available factors, and the fact that these new factors are able to make a positive contribution using the value preferences determined by the other factors sharing their values, offers evidence that values are significant in accounting for the importance to be placed on factors.

Q4: *How should sets of values be compared?*

When the theory is executed it is translated to Prolog code and the rule and value preferences defined in the theory are used to order the rules in the theory. The first rule to match each case in the theory provided the outcome for the case. A second possibility is to discount the values found in both the pro-plaintiff and pro-defendant sets and use the next value.

The evidence is by no means clear cut here. Theories 1 and 3 perform better with cancellation on the training set, but this improvement did not generalise to the Group 2 cases.

Q5: *Is it possible to use a general principle to pre-order factors relating to the same value?*

This question is addressed by the exceptions and CATO methods used to pre-order the factors. These methods perform about the same, and perform the same as when no pre-ordering is used. The results suggest that pre-ordering works to a certain extent, and can reduce the need for refinement - which amounts to a manual ordering of factors within a value. It does not, however, improve the capacity to generalise.

Q6: *Is there evidence to suggest that factors promote values to different degrees?*

This question is addressed by the CATO and IBP based methods of assigning weights to factors. Using the rule preferences contained in the theory, each value is given a weight and the weight is given to the factors promoting the value. However the weight is modified according to whether the factor is an exception (for the exception method) or a weak or knock-out factor (for the IBP method).

The results show the need to use knowledge of the domain to modify the value weights because using the IBP method gives better results than when just using exceptions and better results than when using unmodified value weights.

Q7: *Is there evidence to suggest that factors relating to a particular value have a cumulative effect?*

A weight for each case is obtained by summing the weights either for the values promoted by the factors in the case (using "sets" of values) or by summing the weights for the factors in the case (using "bags" of values).

On the basis of these experiments, there appears to be little difference between accumulating weights from values, from all the factors, and from giving differential weights to factors within values. For Q7 and both theories 1 and 3 the same cases are misclassified whichever method is used. With regard to Q6, the experience of [15], however, which reports IBP, which does use differential weights, as significantly outperforming programs that do not, suggests that we should investigate this further. I certainly have no evidence to deny this, and would need to run the experiment on a larger data set then I have available before coming to any firm conclusions.

The extended Theory Constructors described by Bench-Capon and Sartor in [10] were then implemented as an extension to CATE. These allow the background cases to be described using dimensions as in HYPO [2] in stead with factors as in CATO [1]. This means I can strengthen or weaken a case along a particular dimension and can create theories containing these hypothetical cases.

Similar questions to those asked for the factor-based representation were posed and a series of experiments were performed to answer them.

The first step was to convert the current case background so that the cases were described using dimensions and we employed two methods to explore this. The first method simply took each value to be a dimension and all the factors promoting the value were arranged according to their relative strengths. The second method explored the fact that the factors can have several aspects and in CATO [1] and IBP [15] relate to several high level legal issues.

I found that the two versions of the dimensions performed about the same and there was no real advantages gained over the factor-based representation when using unweighted factors. When I used weights for the factors I found that the method which assigned weights according to where the factor appeared on the dimension performed the best, supporting the idea that factors promote values to different degrees and that these factors can be ordered using dimensions.

In sum I found that the Theory Constructors described in [10] could be used to construct theories able to explain bodies of case law. I then investigated how the construction of such theories could be automated.

10.2 Automation of Theory Construction

The next phase in the project was to automate the process of Theory Construction and I drew on various ideas from Case-Based Reasoning and implemented AGATHA. AGATHA models a two player dialogue game where the plaintiff and defendant agents take turns to make argument moves and, as each move is associated with several Theory Constructors, a theory is constructed and refined as a side effect of the dialogue. AGATHA shows that this method is able to construct a set of plausible theories and for a small case background the space of theories can be exhaustively examined and analysed. However as the size of the case background increases the search space rapidly becomes very large causing computational problems and difficulties in interpreting the output.

Because the automation process is modelled as a two player game, there are several search heuristics, described in many textbooks including [40], [35] and [31], that could be implemented to guide the agents into making the best move. But before I could implement any heuristics I had to determine the relative worth of each theory by evaluating them to find the "best" ones. ETHEL (Evaluation of THEories in Law) was implemented and ETHEL analysed each theory produced by AGATHA and evaluated them using a set of criterial including simplicity and the ability to generalise.

AGAHTA was then extended to use two search heuristics to guide the agents through the search space. The first heuristic is based on A* search where the two agents are cooperating to produce the "best" theory possible and the actual winner of the dialogue game is unimportant. This heuristic can dramatically reduce the size of the search space but can still produce theories with an explanatory power comparable to the best of the other systems such as IBP ([3], [15], [14] and [16]).

The second heuristic is based on $\alpha\beta$ pruning where the two agents are competing with each other to win but also prevent their opponent from winning. This heuristic produces theories of equal explanatory power to those produced by the A* heuristic, but the dialogues are much longer with many refinements made to the theories by the agents. Although $\alpha\beta$ goes deeper into the tree, it explores fewer nodes than A*.

The final program, ROSALIND (AGATHA's Daughter) was used to explore what happens when the two agents have access to a different set of background cases. Each agent must base their next move on what they assume the other agent would do if it had their cases, but as the the other agent has it own set of cases, this introduces uncertainty into the dialogue game.

We found that there is little effect to the results if one player has all precedents and the other only the precedents favouring its own side. But if the information available to one side is inferior (for example, in terms of the number of factors in its precedent cases) to that of the other the final theories are significantly worse: the better equipped player does not have to produce a very high quality theory to win the game.

The aim of this project was to thoroughly investigate the view that reasoning with legal cases could be seen as theory construction and to investigate the approach of theory construction using the Theory Constructors of Bench-Capon and Sartor in [10].

From these investigations I found that the Theory Constructors can produce usable theories to explain bodies of case law and that the process of theory construction can be automated using dialogue games and associated heuristics restrict the theory space for large case backgrounds.

10.3 Future work

For future work the programs described here should be applied to other domains and larger domains so that the tentative conclusions I reached here can be confirmed. Domains

A second aim of my work was to construct theories through the use of plausible dialogues using techniques from standard Case Based Reasoning. Some preliminary work has been reported in [19] but further refinement to give criteria for what makes a "plausible" Dialogue is needed. Some dialogues have been described in Appendix G but this work is very embryonic.

Appendix A

CATO factors

F1 Disclosure-In-Negotiations (d)

Description: Plaintiff disclosed its product information in negotiations with defendant. This factor shows that defendant apparently obtained its information by fair means. Also, it shows that plaintiff showed a lack of interest in maintaining the secrecy of its information.

The factor applies if: Plaintiff disclosed the information to defendant in the context of (negotiations about) a joint venture, licensing agreement, sale of a business, etc.

The factor does not apply if: Defendant acquired knowledge of plaintiffs information in the course of employment by plaintiff.

F2 Bribe-Employee (p)

Description: Defendant paid plaintiff's former employee to switch employment, apparently in an attempt to induce the employee to bring plaintiff's information. This factor shows that defendant may have acquired plaintiff's information through questionable means.

The factor applies if: Defendant offered plaintiff's employee or former employee a substantial bonus or salary increase in order to work for defendant.

F3 Employee-Sole-Developer (d)

Description: Employee defendant was the sole developer of plaintiff's product. This factor shows that defendant may have ownership rights in the information.

The factor does not apply if: Defendant contributed to the development or improvement of plaintiff's product, but was not the sole developer.

F4 Agreed-Not-To-Disclose (p)

Description: Defendant entered into a nondisclosure agreement with plaintiff. This factor shows that defendant was on notice that using or disclosing the information would be a breach of confidentiality. Also, it shows that there was an express agreement to keep the information confidential. Also, that plaintiff took efforts to maintain the secrecy of its information.

The factor does not apply if: Plaintiff obtained nondisclosure agreements from other employees but not from the defendant.

F5 Agreement-Not-Specific (d)

Description: The nondisclosure agreement did not specify which information was to be treated as confidential. This factor shows that it was not the case that defendant was on notice that using or disclosing the information would be a breach of confidentiality.

The factor does not apply if : There is no information about the contents of the nondisclosure agreement.

F6 Security-Measures (p)

Description: Plaintiff adopted security measures. This factor shows that plaintiff took efforts to maintain the secrecy of its information.

The factor applies if: Plaintiff took active measures to limit access to and distribution of its information, for example through employee nondisclosure agreements, notifying employees that information is confidential and not to be divulged to outsiders, keeping important documents under lock and key, document distribution systems, stamping documents confidential, computer passwords, plant security, requiring outsiders to whom information is disclosed to sign nondisclosure agreements, keeping sensitive information hidden when plant tours are conducted, etc.

F7 Brought-Tools (p)

Description: Plaintiff's former employee brought product development information to defendant. This factor shows that defendant may have used plaintiff's information and usurped a competitive advantage. Also, it shows that defendant may have acquired plaintiff's information through questionable means.

The factor applies if: Plaintiff's (former) employee, took product development information such as copies of blueprints, documents, customer lists, computer printouts, disks, tapes, actual specimen of plaintiff's product, parts, tools, etc.

The factor does not apply if: Defendant had somehow come into possession of plaintiff's documents, blueprints, etc., but there was no evidence that an employee of plaintiff's was involved.

F8 Competitive-Advantage (p)

Description: Defendant's access to plaintiff's product information saved it time or expense. This factor shows that defendant may have used plaintiff's information and usurped a competitive advantage. Also, it shows that plaintiff's information was valuable for plaintiff's business.

The factor applies if: It was documented that defendant developed its product at lower cost or in less time than it took plaintiff.

The factor does not apply if: All we know is that the information afforded the plaintiff a competitive advantage (e.g., by enabling it to manufacture a product that was superior to the products made by competitors). Or if all we know is that plaintiff spent considerable time and money in developing the information.

F10 Secrets-Disclosed-Outsiders (d)

Description: Plaintiff disclosed its product information to outsiders. This factor shows that plaintiff's information was known in the industry or available from sources outside plaintiff's business. Also, it shows that plaintiff showed a lack of interest in maintaining the secrecy of its information.

The factor applies if: Plaintiff disclosed its product information for example to licensees, customers, suppliers, subcontractors, etc.

The factor does not apply if: Plaintiff published the information in a public forum. (In that situation, F27 applies.) All we know is that plaintiff marketed a product from which the information could be ascertained by reverse engineering.

F11 Vertical-Knowledge (d)

Description: Plaintiff's information is about customers and suppliers (which means that it may be available independently from customers or even in directories). This factor shows that defendant obtained or could have obtained its information by legitimate means.

The factor applies if: Plaintiff's information consists of customer information such as customer lists or information about customer business methods.

F12 Outsider-Disclosures-Restricted (p)

Description: Plaintiff's disclosures to outsiders were subject to confidentiality restrictions. This factor shows that the information apparently was not known or available outside plaintiff's business. Also, it shows that plaintiff took efforts to maintain the secrecy of its information.

The factor applies if: Plaintiff required that outsiders who received the information keep it confidential or do not use it for any purpose other than for which it was given.

The factor does not apply if: All we know is that plaintiff restricted the number of disclosees or the extent of the information that was disclosed.

F13 Noncompetition-Agreement (p)

Description: Plaintiff and defendant entered into a noncompetition agreement. This factor shows that defendant was on notice that using or disclosing the information would be a breach of confidentiality.

The factor applies if: Defendant entered into an agreement, promising not to compete with plaintiff or work for a competitor after termination of his or her employment by plaintiff.

F14 Restricted-Materials-Used (p)

Description: Defendant used materials that were subject to confidentiality restrictions. This factor shows that defendant was on notice that using or disclosing the information would be a breach of confidentiality. Also, it shows that defendant may have acquired plaintiff's information through questionable means.

The factor applies if: Defendant used documents or materials that plaintiff had marked as confidential or that were subject to a confidentiality agreement between plaintiff and defendant.

F15 Unique-Product (p)

Description: Plaintiff was the only manufacturer making the product. This factor shows that the information apparently was not known or available outside plaintiff's business. Also, it shows that plaintiff's information was valuable for plaintiff's business.

The factor applies if: Plaintiff's product or process was unique on the market or industry, or had marketable features not found in competitors' products.

F16 Info-Reverse-Engineerable (d)

Description: Plaintiff's product information could be learned by reverse engineering. This factor shows that plaintiff's information was known in the industry or available from sources outside plaintiff's business.

The factor applies if: Plaintiff's information could be ascertained by reverse engineering, that is, by inspecting or analyzing plaintiff's product (regardless of whether defendant actually obtained the information in this way).

F17 Info-Independently-Generated (d)

Description: Defendant developed its product by independent research. This factor shows that defendant's information was the result of independent development efforts and investment. Also, it shows that defendant apparently obtained its information by fair means.

The factor applies if: Defendant developed its product or information independently, without recourse to plaintiff's information.

F18 Identical-Products (p)

Description: Defendant's product was identical to plaintiff's. This factor shows that defendant may have used plaintiff's information and usurped a competitive advantage.

F19 No-Security-Measures (d)

Description: Plaintiff did not adopt any security measures. This factor shows that plaintiff showed a lack of interest in maintaining the secrecy of its information.

The factor does not apply if: Plaintiff took at least some security measures, even if other security measures were not taken, or if there is no information about security measures. Or if all we know is that plaintiff disclosed its information to defendant or to outsiders. (In those situations, F1 and F10 apply, respectively.)

F20 Info-Known-To-Competitors (d)

Description: Plaintiff's information was known to competitors. This factor shows that plaintiff's information was known in the industry or available from sources outside plaintiff's business.

The factor applies if: The information plaintiff claims as its trade secret is general knowledge in the industry or trade.

The factor does not apply if: Competitors' knowledge of plaintiff's information results solely from disclosures made by plaintiff. (In this situation, F10 applies.) Or if the information could be compiled from publicly available sources, but there was no evidence that competitors had actually done so. (In this situation, F24 applies.)

F21 Knew-Info-Confidential (p)

Description: Defendant knew that plaintiff's information was confidential. This factor shows that defendant was on notice that using or disclosing the information would be a breach of confidentiality.

The factor applies if: Defendant knew that plaintiff intended its information to be treated as confidential (regardless of how defendant had come to know this).

The factor does not apply if: Defendant entered into a nondisclosure agreement with plaintiff, but there is no evidence that defendant knew specifically which information was to be treated as confidential. (In that situation, F4 applies.)

F22 Invasive-Techniques (p)

Description: Defendant used invasive techniques to gain access to plaintiff's information. This factor shows that defendant may have acquired plaintiff's information through questionable means.

The factor applies if: Defendant used invasive methods in a deliberate attempt to obtain plaintiff's information. These may be illegal methods, such as theft, surreptitious methods, such as rifling through trash or eavesdropping, methods devised specifically to circumvent security measures, methods against which it would be very difficult to guard, such as aerial photography, etc.

The factor does not apply if: Defendant tried to bribe plaintiff's employees to disclose confidential information. (In this situation, F2 applies.) Defendant obtained copies of documents, blueprints, tools, etc. via a (former) employee of plaintiff's. (In this situation, F7 applies.)

F23 Waiver-Of-Confidentiality (d)

Description: Plaintiff entered into an agreement waiving confidentiality. This factor shows that there was an explicit disclaimer of confidentiality. Also, it shows that it was not the case that defendant was on notice that using or disclosing the information would be a breach of confidentiality. Also, that plaintiff showed a lack of interest in maintaining the secrecy of its information.

The factor applies if: Plaintiff acknowledged that defendant did not receive any information in confidence.

F24 Info-Obtainable-Elsewhere (d)

Description: The information could be obtained from publicly available sources. This factor shows that plaintiff's information was known in the industry or available from sources outside plaintiff's business.

The factor does not apply if: Plaintiff's information was general knowledge in the industry. (In that situation, F20 applies.) Or if plaintiff's information could be discovered by reverse engineering plaintiff's product. (In that situation, F16 applies.)

F25 Info-Reverse-Engineered (d)

Description: Defendant discovered plaintiff's information through reverse engineering. This factor shows that defendant apparently obtained its information by fair means.

The factor applies if: Defendant reverse engineered plaintiff's product (i.e., examined or analyzed the product to find out its constituent parts or the process by which it was made).

F26 Deception (p)

Description: Defendant obtained plaintiff's information through deception. This factor shows that defendant may have acquired plaintiff's information through questionable means.

The factor applies if: Defendant deceived plaintiff so as to gain access to its information, or was otherwise dishonest in its dealings with plaintiff.

F27 Disclosure-In-Public-Forum (d)

Description: Plaintiff disclosed its information in a public forum. This factor shows that plaintiff showed a lack of interest in maintaining the secrecy of its information. Also, it shows that plaintiff's information was known in the industry or available from sources outside plaintiff's business.

The factor applies if: Plaintiff made presentations about its information during meetings that were open to the general public, for example, scientific seminars, trade shows, etc. Also if plaintiff published its information in magazine articles, trade publications, publicity material, patents, etc.

The factor does not apply if: Plaintiff disclosed its information to specific outsiders. (In that situation, F10 applies.)

Appendix B

Factor CATE Results

B.1 Factor Comparison Results

Table B.1: Best Factor Comparison when using only a selected number of factors. Only Theory 1 requires refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	No	d	No
Boeing	X	p	p	Refine	p	Refine
Bryce	p	p	p		p	
College Watercolour	p	p	p		p	
Den-Tal-Ez	X	p	p		p	
Ecologix	d	X	d		X	
Emery	p	p	p		p	
Ferranti	d	d	d		d	
Robinson	d	d	d		d	
Sandlin	d	d	d		d	
Sheets	d	d	d		d	
Space Aero	X	p	p		p	
Televation	p	p	p		p	
Yokana	d	d	d		d	
Group 2						
CMI	d	d	d		d	
Digital Development	p	p	p		p	
FMC	X	X	Abs		Abs	
Forrest	X	p	p		p	
Goldberg	X	p	X		p	
KG	p	p	p		p	
Laser	X	p	p		p	
Lewis	X	p	p		p	
MBL	d	d	d		Abs	
Mason	X	p	p		p	
Mineral Deposits	X	X	Abs		X	
National Instrument	p	p	p		p	
National Rejectors	d	d	X		X	
Reinforced	X	p	p		p	
Scientology	d	d	d		Abs	
Technicon	X	p	p		p	
Trandes	X	X	Abs		Abs	
Valco-Cincinnati	X	p	p		p	

Table B.2: Best Factor Comparison when using all factors. All Theories need refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	d	d	d
Boeing	X	p	X	p	p	p
Bryce	p	p	X	p	p	p
College Watercolour	p	p	p	p	p	p
Den-Tal-Ez	X	p	X	p	p	p
Ecologix	d	d	d	d	X	d
Emery	p	p	X	X	p	p
Ferranti	d	d	d	d	d	d
Robinson	d	d	d	d	d	d
Sandlin	d	d	d	d	d	d
Sheets	d	d	d	d	d	d
Space Aero	X	p	p	p	p	p
Televation	p	p	p	p	p	p
Yokana	d	d	d	d	d	d
Group 2						
CMI	d	X	d	d	X	X
Digital Development	p	p	p	p	p	p
FMC	X	p	X	X	p	p
Forrest	X	p	p	p	p	p
Goldberg	X	p	X	X	p	p
KG	p	p	p	p	p	p
Laser	X	p	X	p	p	p
Lewis	X	p	X	X	p	p
MBL	d	X	d	d	X	X
Mason	X	p	p	p	p	p
Mineral Deposits	X	X	X	X	X	X
National Instrument	p	p	X	d	p	p
National Rejectors	d	d	X	X	X	d
Reinforced	X	p	p	p	p	p
Scientology	d	X	d	d	X	X
Technicon	X	p	X	X	p	p
Trandes	X	p	X	p	p	p
Valco-Cincinnati	X	p	p	p	p	p

Table B.3: Cancellation Factor Comparison when using only a selected number of factors. Only Theory 1 requires refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	No	d	No
Boeing	p	p	p	Refine	p	Refine
Bryce	p	p	p		p	
College Watercolour	p	p	p		p	
Den-Tal-Ez	p	p	p		p	
Ecologix	d	d	d		d	
Emery	p	p	p		p	
Ferranti	d	d	d		d	
Robinson	d	d	d		d	
Sandlin	d	d	d		d	
Sheets	d	d	d		d	
Space Aero	X	p	p		p	
Televation	p	p	p		p	
Yokana	d	d	d		d	
Group 2						
CMI	Abs	Abs	d		d	
Digital Development	p	p	p		p	
FMC	Abs	Abs	Abs		Abs	
Forrest	p	p	p		p	
Goldberg	X	X	X		p	
KG	p	p	p		p	
Laser	p	p	p		p	
Lewis	X	X	p		p	
MBL	Abs	Abs	d		Abs	
Mason	p	p	p		p	
Mineral Deposits	X	X	Abs		d	
National Instrument	p	p	p		p	
National Rejectors	d	d	d		d	
Reinforced	p	p	p		p	
Scientology	Abs	Abs	d		Abs	
Technicon	p	p	p		p	
Trandes	Abs	Abs	Abs		Abs	
Valco-Cincinnati	p	p	p		p	

Table B.4: Cancellation Factor Comparison when using all factors. Only Theory 1 requires refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	No	d	No
Boeing	p	p	p	Refine	p	Refine
Bryce	p	p	p		p	
College Watercolour	p	p	p		p	
Den-Tal-Ez	p	p	p		p	
Ecologix	d	d	d		d	
Emery	p	p	X		p	
Ferranti	d	d	d		d	
Robinson	d	d	d		d	
Sandlin	d	d	d		d	
Sheets	d	d	d		d	
Space Aero	X	p	p		p	
Televation	p	p	p		p	
Yokana	d	d	d		d	
Group 2						
CMI	d	d	d		X	
Digital Development	p	p	p		p	
FMC	X	X	X		p	
Forrest	p	p	p		p	
Goldberg	X	X	X		p	
KG	p	p	p		p	
Laser	p	p	p		p	
Lewis	X	X	X		p	
MBL	d	d	d		d	
Mason	p	p	p		p	
Mineral Deposits	X	X	X		X	
National Instrument	p	p	X		p	
National Rejectors	d	d	d		d	
Reinforced	p	p	p		p	
Scientology	d	d	d		X	
Technicon	p	X	X		p	
Trandes	p	p	p		p	
Valco-Cincinnati	p	p	p		p	

Table B.5: Exception Factor Comparison when using only a selected number of factors. Only Theory 1 requires refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	No	d	No
Boeing	X	p	p	Refine	p	Refine
Bryce	p	p	p		p	
College Watercolour	p	p	p		p	
Den-Tal-Ez	X	p	p		p	
Ecologix	d	X	d		p	
Emery	p	p	p		p	
Ferranti	d	d	d		d	
Robinson	d	d	d		d	
Sandlin	d	d	d		d	
Sheets	d	d	d		d	
Space Aero	X	p	p		p	
Televation	p	p	p		p	
Yokana	d	d	d		d	
Group 2						
CMI	d	d	d		d	
Digital Development	p	p	p		p	
FMC	X	X	Abs		Abs	
Forrest	X	p	p		p	
Goldberg	X	p	X		p	
KG	p	p	p		p	
Laser	X	p	p		p	
Lewis	X	p	p		p	
MBL	d	d	d		Abs	
Mason	X	p	p		p	
Mineral Deposits	X	X	Abs		X	
National Instrument	p	p	p		p	
National Rejectors	d	d	X		X	
Reinforced	X	p	p		p	
Scientology	d	d	d		Abs	
Technicon	X	p	p		p	
Trandes	X	X	Abs		Abs	
Valco-Cincinnati	X	p	p		p	

Table B.6: Exception Factor Comparison when using all factors. No refinement is necessary or makes no improvement to the theories.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	no change	d	No	d	No
Boeing	p		p	Refine	p	Refine
Bryce	p		p		p	
College Watercolour	p		p		p	
Den-Tal-Ez	p		p		p	
Ecologix	d		d		d	
Emery	p		d		p	
Ferranti	d		d		d	
Robinson	d		d		d	
Sandlin	d		d		d	
Sheets	d		d		d	
Space Aero	p		p		p	
Televation	p		p		p	
Yokana	d		d		d	
Group 2						
CMI	X		d		X	
Digital Development	p		p		p	
FMC	p		X		p	
Forrest	p		p		p	
Goldberg	X		X		p	
KG	p		p		p	
Laser	p		p		p	
Lewis	X		X		p	
MBL	d		d		d	
Mason	p		p		p	
Mineral Deposits	X		X		X	
National Instrument	p		X		p	
National Rejectors	d		X		X	
Reinforced	p		p		p	
Scientology	X		d		X	
Technicon	p		X		p	
Trandes	p		p		p	
Valco-Cincinnati	p		p		p	

Table B.7: CATO Factor Comparison when using only a selected number of factors. Only Theory 1 requires refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	No	d	No
Boeing	X	p	p	Refine	p	Refine
Bryce	p	p	p		p	
College Watercolour	p	p	p		p	
Den-Tal-Ez	X	p	p		p	
Ecologix	d	X	d		p	
Emery	p	p	p		p	
Ferranti	d	d	d		d	
Robinson	d	d	d		d	
Sandlin	d	d	d		d	
Sheets	d	d	d		d	
Space Aero	X	p	p		p	
Televation	p	p	p		p	
Yokana	d	d	d		d	
Group 2						
CMI	d	d	d		d	
Digital Development	p	p	p		p	
FMC	X	X	Abs		Abs	
Forrest	X	p	p		p	
Goldberg	X	p	X		p	
KG	p	p	p		p	
Laser	X	p	p		p	
Lewis	X	p	p		p	
MBL	d	d	d		Abs	
Mason	X	p	p		p	
Mineral Deposits	X	X	Abs		X	
National Instrument	p	p	p		p	
National Rejectors	d	d	X		X	
Reinforced	X	p	p		p	
Scientology	d	d	d		Abs	
Technicon	X	p	p		p	
Trandes	X	X	Abs		Abs	
Valco-Cincinnati	X	p	p		p	

Table B.8: CATO Factor Comparison when using all factors. All Theories need refinement.

Cases	Theory 1		Theory 2		Theory 3	
	Before	After	Before	After	Before	After
Group 1						
Arco	d	d	d	d	d	d
Boeing	p	p	p	p	p	p
Bryce	p	p	X	p	p	p
College Watercolour	p	p	p	p	p	p
Den-Tal-Ez	p	p	p	p	p	p
Ecologix	X	d	d	d	X	d
Emery	p	p	X	X	p	p
Ferranti	d	d	d	d	d	d
Robinson	d	d	X	d	d	d
Sandlin	d	d	d	d	d	d
Sheets	d	d	d	d	d	d
Space Aero	p	p	p	p	p	p
Televation	p	p	p	p	p	p
Yokana	d	d	d	d	d	d
Group 2						
CMI	d	d	d	d	X	X
Digital Development	p	p	p	p	p	p
FMC	X	p	X	X	p	p
Forrest	p	p	p	p	p	p
Goldberg	p	X	X	X	p	p
KG	p	p	p	p	p	p
Laser	p	p	X	p	p	p
Lewis	p	X	p	p	p	p
MBL	d	d	d	d	d	d
Mason	p	p	p	p	p	p
Mineral Deposits	X	X	X	X	X	X
National Instrument	p	p	X	X	p	p
National Rejectors	d	d	X	X	X	X
Reinforced	p	p	p	p	p	p
Scientology	d	d	d	d	X	X
Technicon	p	p	p	X	p	p
Trandes	X	p	X	p	p	p
Valco-Cincinnati	p	p	p	p	p	p

B.2 Weighting Results

Table B.9: Results when using weights based on accumulation of Value Weights.

Cases	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Group 1						
Arco	-0.3	-1	-1.5	-2.2	-0.7	-1
Boeing	0.4	0.8	0.3	0.4	1.5	1.6
Bryce	0.5	0.8	0.3	0.4	1.6	1.6
College Watercolour	0.5	0.5	1.5	0.9	0.8	0.5
Den-Tal-Ez	0.5	0.8	0.3	0.4	1.6	1.6
Ecologix	*0.4*	-0.3	-0.4	-0.7	*1.2*	-0.3
Emery	0.5	0.5	0.3	*-0.3*	1.6	1.3
Ferranti	-0.3	-0.9	-2.2	-2.1	-0.3	-0.9
Robinson	-0.1	-0.1	-0.7	-0.5	-0.1	-0.1
Sandlin	-0.3	-1	-0.7	-2.2	-1	-1
Sheets	-0.2	-0.2	-0.7	-0.6	-0.2	-0.2
Space Aero	0.5	0.5	0.8	0.9	0.5	0.5
Televation	1.2	0.8	1.8	0.4	1.6	1.6
Yokana	-0.3	-0.9	-0.7	-2.1	-0.7	-0.9
Group 2						
CMI	-0.3	0	-2.2	-1.2	-0.7	*0.8*
Digital Development	1.2	1.5	1.8	1.9	2.3	2.3
FMC	*-0.3*	0.1	*-1.1*	*-1.1*	1.4	0.9
Forrest	1.1	1.4	1.8	1.8	2.2	2.2
Goldberg	0.4	0.4	*-0.4*	*-0.4*	1.5	1.2
KG	1.5	1.2	1.8	1.2	1.6	2
Laser	0.4	0.7	0.3	0.3	1.5	1.5
Lewis	0.4	0.5	0.3	*-0.3*	1.5	1.3
MBL	-0.3	-0.7	-1.5	-1.5	*1.4*	-0.7
Mason	1.1	0.7	1.8	0.3	1.5	1.5
Mineral Deposits	*-0.2*	*-0.9*	*-1.1*	*-2.1*	*-0.6*	*-0.9*
National Instrument	0.5	0.5	0.3	-0.3	1.6	1.3
National Rejectors	*0.5*	-0.1	*0.8*	-0.5	-0.2	-0.1
Reinforced	1.1	1.5	1.8	1.9	2.2	2.3
Scientology	-0.3	0	-1.5	-1.2	*1.4*	*0.8*
Technicon	0.4	0.1	0.3	*-1.1*	0.8	0.9
Trandes	*-0.3*	0.7	*-1.1*	0.3	1.4	1.5
Valco-Cincinnati	1.1	1.4	1.8	1.8	2.2	2.2

Table B.10: Results when using weights based on accumulation of Factor Weights.

Cases	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Group 1						
Arco	-0.3	-1.7	-1.5	-3.7	-0.7	-1.7
Boeing	0.1	1.5	0.3	0.7	1.5	3.1
Bryce	0.5	1.5	0.3	0.7	1.6	3.1
College Watercolour	0.5	0.5	1.5	0.9	0.8	0.5
Den-Tal-Ez	0.5	1.5	0.3	0.7	1.6	3.1
Ecologix	*0.1*	-0.6	-0.4	-1.4	*1.2*	-0.6
Emery	0.5	0.5	0.3	*-0.3*	1.6	1.3
Ferranti	-0.6	-1.9	-2.9	-4.3	-0.3	-1.9
Robinson	-0.7	-0.7	-0.7	-1.9	-0.1	-0.7
Sandlin	-1.2	-1.9	-1.4	-4.3	-1	-1.9
Sheets	-0.5	-0.5	-1.4	-1.3	-0.2	-0.5
Space Aero	0.2	0.3	0.8	0.3	0.5	0.3
Televation	1.2	1.1	1.8	1.1	1.6	1.9
Yokana	-0.6	-1.2	-0.7	-2.8	-0.7	-1.2
Group 2						
CMI	-0.6	-1.7	-2.2	-4.9	-0.7	-0.9
Digital Development	1.2	1.6	1.8	2	2.3	2.4
FMC	*-0.3*	0.4	0	*-0.4*	0	1.2
Forrest	1.1	1.4	1.8	1.8	2.2	2.2
Goldberg	*-0.2*	*-0.2*	*-0.4*	*-1.8*	1.5	0.6
KG	1.5	0.5	1.8	*-0.3*	1.6	1.3
Laser	0.1	0.7	0.3	0.3	1.5	1.5
Lewis	0.4	0.5	0.3	*-0.3*	1.5	1.3
MBL	-0.3	0	-1.5	-1.2	0	*0.8*
Mason	1.1	0.7	1.8	0.3	1.5	1.5
Mineral Deposits	*-0.2*	*-1.6*	0	*-3.6*	*-0.6*	*-1.6*
National Instrument	0.5	0.5	0.3	*-0.3*	1.6	1.3
National Rejectors	-0.1	-0.7	*0.1*	-1.9	-0.2	-0.7
Reinforced	1.1	2.2	1.8	2.2	2.2	3.8
Scientology	-0.3	-0.4	-1.5	-2	0	*0.4*
Technicon	0.4	*-0.3*	0.3	*-1.9*	0.8	0.5
Trandes	*-0.6*	0.7	0	0.3	0	1.5
Valco-Cincinnati	0.8	1.4	1.8	1.8	2.2	2.2

Table B.11: Results when using weights based on Exceptions.

Cases	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Group 1						
Arco	-0.3	-1.7	-1.5	-3.7	-0.7	-1.7
Boeing	0.1	6.9	0.3	13.3	1.5	8.5
Bryce	0.5	4.2	0.3	7	1.6	5.8
College Watercolour	6.8	6.8	15	14.4	7.1	6.8
Den-Tal-Ez	0.5	4.2	0.3	7	1.6	5.8
Ecologix	*0.1*	-6.9	-0.4	-4.1	*1.2*	-14.1
Emery	0.5	0.5	0.3	*-0.3*	1.6	1.3
Ferranti	-0.6	-1.9	-2.9	-4.3	-0.3	-1.9
Robinson	-0.7	-0.7	-0.7	-1.9	-0.1	-0.7
Sandlin	-1.2	-1.9	-1.4	-4.3	-1	-1.9
Sheets	-0.5	-0.5	-1.4	-1.3	-0.2	-0.5
Space Aero	6.5	6.6	14.3	13.8	6.8	6.6
Televation	7.5	12.8	15.3	27.2	7.9	13.6
Yokana	-0.6	-1.2	-0.7	-2.8	-0.7	-1.2
Group 2						
CMI	-0.6	*1*	-2.2	1.4	-0.7	*1.8*
Digital Development	7.5	10.6	15.3	21.8	8.6	11.4
FMC	*-0.3*	5.8	0	12.2	0	6.6
Forrest	7.4	10.4	15.3	21.6	8.5	11.2
Goldberg	*-0.2*	*-0.2*	*-0.4*	*-1.8*	1.5	0.6
KG	7.8	9.5	15.3	19.5	7.9	10.3
Laser	0.1	6.1	0.3	12.9	1.5	6.9
Lewis	0.4	0.5	0.3	*-0.3*	1.5	1.3
MBL	-0.3	-3.6	-1.5	2.4	0	-10
Mason	7.4	9.7	15.3	20.1	7.8	10.5
Mineral Deposits	*-0.2*	*-1.6*	0	*-3.6*	*-0.6*	*-1.6*
National Instrument	0.5	0.5	0.3	-0.3	1.6	1.3
National Rejectors	*6.2*	5.6	*13.6*	*11.6*	*6.1*	*5.6*
Reinforced	7.4	11.2	15.3	22	8.5	12.8
Scientology	-0.3	*5*	-1.5	*10.6*	0	*5.8*
Technicon	0.4	5.1	0.3	10.7	0.8	5.9
Trandes	*-0.6*	6.1	0	12.9	0	6.9
Valco-Cincinnati	7.1	13.1	15.3	27.9	8.5	13.9

Table B.12: Results when using weights based on IBP.

Cases	Theory 1		Theory 2		Theory 3	
	Chosen	All	Chosen	All	Chosen	All
Group 1						
Arco	-0.03	-7.1	-15	-15.22	-0.07	-7.1
Boeing	0.64	2.04	0.3	1.96	1.5	3.64
Bryce	0.77	1.77	0.3	1.33	1.6	3.37
College Watercolour	1.67	1.67	1.5	2.43	1.7	1.67
Den-Tal-Ez	1.67	2.67	0.3	2.23	2.5	4.27
Ecologix	-2.33	-3.03	-6.7	-7.07	-1.5	-3.03
Emery	0.77	0.77	0.3	0.33	1.6	1.57
Ferranti	-6	-13.6	-29	-30.4	-3	-13.6
Robinson	-1.96	-1.96	-7	-6.04	-1.9	-1.96
Sandlin	-6.06	-6.13	-14	-14.29	-3.07	-6.13
Sheets	-5.9	-5.9	-14	-13.9	-2.9	-5.9
Space Aero	*-2.23*	*-1.23*	*-5.5*	*-4.47*	*-2.2*	*-1.23*
Televation	1.47	2	1.8	3.08	2.23	2.8
Yokana	-3.03	-3	-7	-7.12	-0.07	-3
Group 2						
CMI	-3.03	-9.8	-22	-22.72	-0.07	-9
Digital Development	1.47	2.77	1.8	3.53	2.3	3.57
FMC	*-0.03*	0.67	0	0.23	0	1.47
Forrest	1.37	1.67	1.8	2.43	2.2	2.47
Goldberg	*-2.36*	*-2.36*	*-6.7*	*-6.84*	1.5	*-1.56*
KG	1.5	1.13	1.8	1.05	2.23	1.93
Laser	0.64	1.24	0.3	1.56	1.5	2.04
Lewis	0.67	1.67	0.3	1.23	1.5	2.47
MBL	-0.03	-6.03	-15	-14.07	0	-5.23
Mason	1.37	1.6	1.8	2.28	2.13	2.4
Mineral Deposits	0.07	*-0.7*	0	*-1.62*	0.03	*-0.7*
National Instrument	0.77	0.77	0.3	0.33	1.6	1.57
National Rejectors	-5.23	-5.2	-12.5	-12.52	-2.27	-5.2
Reinforced	1.37	3.37	1.8	3.73	2.2	4.97
Scientology	-0.03	-6.43	-15	-14.87	0	-5.63
Technicon	0.67	0.6	0.3	0.08	1.43	1.4
Trandes	*-0.06*	1.24	0	1.56	0	2.04
Valco-Cincinnati	1.34	1.94	1.8	3.06	2.2	2.74

Appendix C

Dimension CATE Results

C.1 Factor Comparison Results

Table C.1: Best Factor Comparison when using Simple Dimensions. Both Theories need refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	d
Boeing	X	p	p	p
Bryce	d	p	p	p
College Watercolour	p	p	p	p
Den-Tal-Ez	X	p	p	p
Ecologix	d	d	X	d
Emery	p	p	p	p
Ferranti	d	d	d	d
Robinson	d	d	d	d
Sandlin	d	d	d	d
Sheets	d	d	d	d
Space Aero	X	p	p	p
Televation	p	p	p	p
Yokana	d	d	d	d
Group 2				
CMI	d	d	X	X
Digital Development	X	p	p	p
FMC	X	X	p	p
Forrest	X	p	p	p
Goldberg	X	p	p	p
KG	p	p	p	p
Laser	X	p	p	p
Lewis	X	p	p	p
MBL	d	X	X	X
Mason	X	p	p	p
Mineral Deposits	X	X	X	X
National Instrument	p	p	p	p
National Rejectors	d	X	X	X
Reinforced	X	p	p	p
Scientology	d	d	X	X
Technicon	X	X	p	p
Trandes	X	p	p	p
Valco-Cincinnati	X	p	p	p

Table C.2: Best Factor Comparison when using Complex Dimensions. Both Theories need refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	d
Boeing	X	p	p	p
Bryce	X	p	p	p
College Watercolour	p	p	p	p
Den-Tal-Ez	X	p	p	p
Ecologix	d	d	X	d
Emery	X	p	p	p
Ferranti	d	d	d	d
Robinson	d	d	d	d
Sandlin	d	d	d	d
Sheets	d	d	d	d
Space Aero	p	p	p	p
Televation	X	p	p	p
Yokana	d	d	d	d
Group 2				
CMI	d	X	X	X
Digital Development	p	p	p	p
FMC	X	p	p	p
Forrest	X	p	p	p
Goldberg	X	p	p	p
KG	p	p	p	p
Laser	X	p	p	p
Lewis	X	p	p	p
MBL	d	X	X	X
Mason	X	p	p	p
Mineral Deposits	X	X	X	X
National Instrument	X	p	p	p
National Rejectors	d	d	X	X
Reinforced	p	p	p	p
Scientology	d	X	X	X
Technicon	d	p	p	p
Trandes	X	p	p	p
Valco-Cincinnati	X	p	p	p

Table C.3: Cancellation Factor Comparison when using Simple Dimensions. Only Theory 1 needs refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	No
Boeing	p	p	p	Refine
Bryce	p	p	p	
College Watercolour	p	p	p	
Den-Tal-Ez	p	p	p	
Ecologix	d	d	d	
Emery	p	p	p	
Ferranti	d	d	d	
Robinson	d	d	d	
Sandlin	d	d	d	
Sheets	d	d	d	
Space Aero	X	p	p	
Televation	p	p	p	
Yokana	d	d	d	
Group 2				
CMI	d	d	X	
Digital Development	p	p	p	
FMC	X	X	p	
Forrest	p	p	p	
Goldberg	X	X	p	
KG	p	p	p	
Laser	p	p	p	
Lewis	X	X	p	
MBL	d	d	d	
Mason	p	p	p	
Mineral Deposits	X	X	X	
National Instrument	p	p	p	
National Rejectors	d	d	d	
Reinforced	p	p	p	
Scientology	d	d	X	
Technicon	p	X	p	
Trandes	p	p	p	
Valco-Cincinnati	p	p	p	

Table C.4: Cancellation Factor Comparison when using Complex Dimensions. Only Theory 1 needs refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	No
Boeing	p	p	p	Refine
Bryce	p	p	p	
College Watercolour	p	p	p	
Den-Tal-Ez	p	p	p	
Ecologix	d	d	d	
Emery	X	p	p	
Ferranti	d	d	d	
Robinson	d	d	d	
Sandlin	d	d	d	
Sheets	d	d	d	
Space Aero	p	p	p	
Televation	p	p	p	
Yokana	d	d	d	
Group 2				
CMI	d	X	X	
Digital Development	p	p	p	
FMC	p	p	p	
Forrest	p	p	p	
Goldberg	X	p	p	
KG	p	p	p	
Laser	X	p	p	
Lewis	X	p	p	
MBL	d	d	d	
Mason	X	p	p	
Mineral Deposits	X	X	X	
National Instrument	X	p	p	
National Rejectors	d	d	d	
Reinforced	p	p	p	
Scientology	d	X	X	
Technicon	p	p	p	
Trandes	X	p	p	
Valco-Cincinnati	X	p	p	

Table C.5: Exception Factor Comparison when using Simple Dimensions. Neither Theory needs refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	No	d	No
Boeing	p	Refine	p	Refine
Bryce	p		p	
College Watercolour	p		p	
Den-Tal-Ez	p		p	
Ecologix	d		d	
Emery	p		p	
Ferranti	d		d	
Robinson	d		d	
Sandlin	d		d	
Sheets	d		d	
Space Aero	p		p	
Televation	p		p	
Yokana	d		d	
Group 2				
CMI	p		X	
Digital Development	p		p	
FMC	p		p	
Forrest	p		p	
Goldberg	X		p	
KG	p		p	
Laser	p		p	
Lewis	X		p	
MBL	d		d	
Mason	p		p	
Mineral Deposits	X		X	
National Instrument	p		p	
National Rejectors	X		X	
Reinforced	p		p	
Scientology	X		X	
Technicon	p		p	
Trandes	p		p	
Valco-Cincinnati	p		p	

Table C.6: Exception Factor Comparison when using Complex Dimensions. Only Theory 1 needs refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	No
Boeing	p	p	p	Refine
Bryce	p	p	p	
College Watercolour	p	p	p	
Den-Tal-Ez	p	p	p	
Ecologix	d	d	d	
Emery	p	p	p	
Ferranti	X	d	d	
Robinson	X	d	d	
Sandlin	d	d	d	
Sheets	X	d	d	
Space Aero	p	p	p	
Televation	p	p	p	
Yokana	p	d	d	
Group 2				
CMI	X	X	X	
Digital Development	p	p	p	
FMC	p	p	p	
Forrest	p	p	p	
Goldberg	p	p	p	
KG	p	p	p	
Laser	p	p	p	
Lewis	p	p	p	
MBL	d	d	d	
Mason	p	p	p	
Mineral Deposits	p	X	X	
National Instrument	p	p	p	
National Rejectors	X	X	X	
Reinforced	p	p	p	
Scientology	X	X	X	
Technicon	p	p	p	
Trandes	p	p	p	
Valco-Cincinnati	p	p	p	

Table C.7: CATO Factor Comparison when using Simple Dimensions. Only Theory 1 needs refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	No
Boeing	p	p	p	Refine
Bryce	X	p	p	
College Watercolour	p	p	p	
Den-Tal-Ez	p	p	p	
Ecologix	p	d	p	
Emery	p	p	p	
Ferranti	d	d	d	
Robinson	p	d	d	
Sandlin	d	d	d	
Sheets	d	d	d	
Space Aero	p	p	p	
Televation	p	p	p	
Yokana	d	d	d	
Group 2				
CMI	d	d	X	
Digital Development	p	p	p	
FMC	X	p	p	
Forrest	p	p	p	
Goldberg	p	p	p	
KG	p	p	p	
Laser	p	p	p	
Lewis	p	p	p	
MBL	d	d	d	
Mason	p	p	p	
Mineral Deposits	X	X	X	
National Instrument	p	p	p	
National Rejectors	p	X	X	
Reinforced	p	p	p	
Scientology	d	d	X	
Technicon	p	X	p	
Trandes	d	p	p	
Valco-Cincinnati	p	p	p	

Table C.8: CATO Factor Comparison when using Complex Dimensions. Both Theories need refinement

Cases	Theory 1		Theory 3	
	Before	After	Before	After
Group 1				
Arco	d	d	d	d
Boeing	p	p	p	p
Bryce	p	p	p	p
College Watercolour	p	p	p	p
Den-Tal-Ez	p	p	p	p
Ecologix	X	d	X	d
Emery	p	p	p	p
Ferranti	d	d	d	d
Robinson	X	d	d	d
Sandlin	d	d	d	d
Sheets	d	d	d	d
Space Aero	p	p	p	p
Televation	p	p	p	p
Yokana	d	d	d	d
Group 2				
CMI	d	X	X	X
Digital Development	p	p	p	p
FMC	X	p	p	p
Forrest	p	p	p	p
Goldberg	p	p	p	p
KG	p	p	p	p
Laser	p	p	p	p
Lewis	p	p	p	p
MBL	d	d	d	d
Mason	p	p	p	p
Mineral Deposits	X	X	X	X
National Instrument	p	p	p	p
National Rejectors	X	X	X	X
Reinforced	p	p	p	p
Scientology	d	d	X	X
Technicon	p	X	p	p
Trandes	X	p	p	p
Valco-Cincinnati	p	p	p	p

C.2 Weighting Results

Table C.9: Results when using weights based on accumulation of Value Weights.

Cases	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Group 1				
Arco	-1	-1	-1.1	-1.1
Boeing	0.8	1.6	1.5	1.5
Bryce	0.8	1.6	1.6	1.6
College Watercolour	0.5	0.5	0.5	0.5
Den-Tal-Ez	0.8	1.6	1.6	1.6
Ecologix	-0.3	-0.3	-0.3	-0.3
Emery	0.5	1.3	0.5	0.5
Ferranti	-0.9	-0.9	-1	-1
Robinson	-0.1	-0.1	-1	-1
Sandlin	-1	-1	-1.1	-1.1
Sheets	-0.2	-0.2	-1	-1
Space Aero	0.5	0.5	0.5	0.5
Televation	0.8	1.6	1.5	1.5
Yokana	-0.9	-0.9	-1	-1
Group 2				
CMI	0	*0.8*	*0.7*	*0.7*
Digital Development	1.5	2.3	2.3	2.3
FMC	0.1	0.9	1.5	1.5
Forrest	1.4	2.2	2.2	2.2
Goldberg	0.4	1.2	0.4	0.4
KG	1.2	2	1.8	1.8
Laser	0.7	1.5	1.4	1.4
Lewis	0.5	1.3	1.3	1.3
MBL	-0.7	-0.7	-0.8	-0.8
Mason	0.7	1.5	1.4	1.4
Mineral Deposits	*-0.9*	*-0.9*	*-1*	*-1*
National Instrument	0.5	1.3	1.3	1.3
National Rejectors	-0.1	-0.1	-0.3	-0.3
Reinforced	1.5	2.3	2.3	2.3
Scientology	0	*0.8*	*1.4*	*1.4*
Technicon	0.1	0.9	1.5	1.5
Trandes	0.7	1.5	1.4	1.4
Valco-Cincinnati	1.4	2.2	1.4	1.4

Table C.10: Results when using weights based on accumulation of Factor Weights.

Cases	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Group 1				
Arco	-1	-1	-1.1	-1.1
Boeing	0.8	1.6	1.5	1.5
Bryce	0.8	1.6	1.6	1.6
College Watercolour	0.5	0.5	0.5	0.5
Den-Tal-Ez	0.8	1.6	1.6	1.6
Ecologix	-0.3	-0.3	-0.3	-0.3
Emery	0.5	1.3	0.5	0.5
Ferranti	-0.9	-0.9	-1	-1
Robinson	-0.1	-0.1	-1	-1
Sandlin	-1	-1	-1.1	-1.1
Sheets	-0.2	-0.2	-1	-1
Space Aero	0.5	0.5	0.5	0.5
Televation	0.8	1.6	1.5	1.5
Yokana	-0.9	-0.9	-1	-1
Group 2				
CMI	0	*0.8*	*0.7*	*0.7*
Digital Development	1.5	2.3	2.3	2.3
FMC	0.1	0.9	1.5	1.5
Forrest	1.4	2.2	2.2	2.2
Goldberg	0.4	1.2	0.4	0.4
KG	1.2	2	1.8	1.8
Laser	0.7	1.5	1.4	1.4
Lewis	0.5	1.3	1.3	1.3
MBL	-0.7	-0.7	-0.8	-0.8
Mason	0.7	1.5	1.4	1.4
Mineral Deposits	*-0.9*	*-0.9*	*-1*	*-1*
National Instrument	0.5	1.3	1.3	1.3
National Rejectors	-0.1	-0.1	-0.3	-0.3
Reinforced	1.5	2.3	2.3	2.3
Scientology	0	*0.8*	*1.4*	*1.4*
Technicon	0.1	0.9	1.5	1.5
Trandes	0.7	1.5	1.4	1.4
Valco-Cincinnati	1.4	2.2	1.4	1.4

Table C.11: Results when using weights based on differential weights based on Exceptions.

Cases	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Group 1				
Arco	-1	-1	-7.13	-2
Boeing	3.5	4.3	2.4	9.6
Bryce	3.5	4.3	1.87	4.3
College Watercolour	6.8	6.8	0.77	6.8
Den-Tal-Ez	3.5	4.3	1.87	4.3
Ecologix	-6.6	-13.8	-3	-13.8
Emery	0.5	1.3	1.4	*-0.4*
Ferranti	-0.9	-0.9	-10	-1.9
Robinson	-0.1	-0.1	-3.07	-1.9
Sandlin	-1	-1	-10.1	-2
Sheets	-0.2	-0.2	-10	-1.9
Space Aero	6.8	6.8	*-2.2*	6.8
Televation	9.8	10.6	2.4	9.6
Yokana	-0.9	-0.9	-10	-1.9
Group 2				
CMI	*2.7*	*3.5*	-8.3	*2.5*
Digital Development	10.5	11.3	2.57	11.3
FMC	2.8	3.6	1.77	9.6
Forrest	10.4	11.2	2.47	11.2
Goldberg	0.4	1.2	*-8.6*	*-0.5*
KG	10.2	11	2.43	9.9
Laser	3.4	4.2	2.3	9.5
Lewis	0.5	1.3	1.57	1.3
MBL	-4.3	-11.5	-6.83	-12.5
Mason	9.7	10.5	2.3	9.5
Mineral Deposits	*-0.9*	*-0.9*	*-0.1*	*-1.9*
National Instrument	0.5	1.3	1.57	1.3
National Rejectors	6.2	*6.2*	-9.3	*5.1*
Reinforced	10.5	11.3	2.57	11.3
Scientology	*2.7*	*3.5*	-4.63	*9.5*
Technicon	2.8	3.6	2.4	9.6
Trandes	3.4	4.2	2.3	9.5
Valco-Cincinnati	10.4	11.2	2.3	9.5

Table C.12: Results when using weights based on differential weights using IBP.

Cases	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Group 1				
Arco	-7.03	-7.03	-1.1	-7.13
Boeing	1.07	1.87	10.5	2.49
Bryce	1.07	1.87	4.3	1.87
College Watercolour	1.67	1.67	6.8	1.67
Den-Tal-Ez	1.97	2.77	4.3	2.77
Ecologix	-3	-3	-13.8	-3
Emery	0.77	1.57	0.5	1.49
Ferranti	-9.9	-9.9	-1	-10.9
Robinson	-1.9	-1.9	-1	-2.08
Sandlin	-3.07	-3.07	-1.1	-11
Sheets	-2.9	-2.9	-1	-10.9
Space Aero	*-1.3*	*-1.3*	6.8	*-1.3*
Televation	1.7	2.5	10.5	2.49
Yokana	-2.97	-2.97	-1	-10.9
Group 2				
CMI	-9	-8.2	*3.4*	-9.2
Digital Development	2.67	3.47	11.3	3.47
FMC	0.37	1.17	10.5	1.77
Forrest	1.67	2.47	11.2	2.47
Goldberg	*-2.3*	*-1.5*	0.4	*-9.5*
KG	1.2	2	10.8	2.43
Laser	0.97	1.77	10.4	2.39
Lewis	1.67	2.47	1.3	2.47
MBL	-6.73	-6.73	-11.6	-6.83
Mason	1.6	2.4	10.4	2.39
Mineral Deposits	*-0.63*	*-0.63*	*-1*	*-0.1*
National Instrument	0.77	1.57	1.3	1.57
National Rejectors	-2.17	-2.17	*6*	-10.2
Reinforced	2.67	3.47	11.3	3.47
Scientology	-6.03	-5.23	*10.4*	-4.63
Technicon	0.37	1.17	10.5	2.4
Trandes	0.97	1.77	10.4	2.39
Valco-Cincinnati	1.67	2.47	10.4	2.39

Table C.13: Results when using weights based on Dimension weighting of Value.

Cases	Simple		Complex	
	Theory 1	Theory 3	Theory 1	Theory 3
Group 1				
Arco	-0.82	-0.82	-0.84	-0.76
Boeing	0.64	1.12	0.47	1.62
Bryce	0.71	1.19	0.55	1.46
College Watercolour	0.74	0.74	0.27	0.72
Den-Tal-Ez	0.76	1.24	0.6	1.52
Ecologix	-0.72	-1.2	-1.5	-0.9
Emery	0.21	0.53	0.14	0.64
Ferranti	-0.94	-0.94	-1.03	-1.04
Robinson	-0.15	-0.15	-0.29	-0.38
Sandlin	-0.44	-0.44	-1.1	-1.1
Sheets	-0.25	-0.25	-1.06	-1.08
Space Aero	0.5	0.5	0.01	0.5
Televation	1.07	1.39	0.36	1.49
Yokana	-0.3	-0.3	-1.02	-1
Group 2				
CMI	-0.16	*0.32*	-0.53	*0.46*
Digital Development	1.32	1.64	0.67	1.94
FMC	0.2	0.68	0.14	1.3
Forrest	1.22	1.54	0.57	1.84
Goldberg	0.1	0.42	*-0.77*	*-0.14*
KG	1.09	1.41	0.49	1.63
Laser	0.46	0.78	0.26	1.27
Lewis	0.32	0.64	0.37	0.94
MBL	-0.17	*0.23*	-1.2	*0.29*
Mason	1.08	1.4	0.41	1.53
Mineral Deposits	*-0.29*	*-0.29*	*-0.16*	*-0.36*
National Instrument	0.27	0.59	0.31	0.86
National Rejectors	*0.33*	*0.33*	-0.84	-0.36
Reinforced	1.46	1.94	0.82	2.24
Scientology	-0.1	*0.38*	-0.23	*1.09*
Technicon	0.22	0.54	0.23	1.16
Trandes	0.6	1.08	0.41	1.57
Valco-Cincinnati	1.16	1.48	0.4	1.62

Appendix D

Evaluation Results

D.1 Wild Animal Theories

D.1.1 Theory 1

```
Theory Cases :
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pNposs
Theory Rules :
  <{pNposs}, D>
Theory Preferences :
Theory Value Preferences :

pierson | d |  outcome(X, d)  :- factor(X, pnposs).
keeble  | d |  outcome(X, d)  :- factor(X, pnposs).
young   | d |  outcome(X, d)  :- factor(X, pnposs).
```

D.1.2 Theory 2

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLiv
  pNposs
Theory Rules :
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valpref({MProd}, {LLit})

pierson | d | outcome(X, d)  :- factor(X, pnposs).
keeble  | p | outcome(X, p)  :- factor(X, pliv).
young   | p | outcome(X, p)  :- factor(X, pliv).
```

D.1.3 Theory 3

```
Theory Cases :
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLiv
  pNposs
Theory Rules :
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv}, P>, <{pNposs}, D>)
       <|From Value Preference|>
Theory Value Preferences :
  valpref({MProd}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pliv).
young   | p | outcome(X, p) :- factor(X, pliv).
```

D.1.4 Theory 4

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLiv
  pNposs
Theory Rules :
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valpref({MProd}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pliv).
young   | p | outcome(X, p) :- factor(X, pliv).
```

D.1.5 Theory 5

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLand
  pLiv
  pNposs
Theory Rules :
  <{pLand}, P>
  <{pLiv, pLand}, P>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv, pLand}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valpref({MProd, MSec}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pland),
              factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, pnposs).
```

D.1.6 Theory 6

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLand
  pLiv
  pNposs
Theory Rules :
  <{pLand}, P>
  <{pLiv, pLand}, P>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv, pLand}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
  pref(<{pNposs}, D>, <{pLiv}, P>)
       <|Arbitrary Rule Preference|>
Theory Value Preferences :
  valpref({LLit}, {MProd})
  valpref({MProd, MSec}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pland),
              factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, pnposs).
```

D.1.7 Theory 7

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  dLiv
  pLiv
  pNposs
Theory Rules :
  <{dLiv, pNposs}, D>
  <{dLiv}, D>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{dLiv, pNposs}, D>, <{pLiv}, P>)
       <|From Value Preference|>
  pref(<{pLiv}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valpref({LLit, MProd}, {MProd})
  valpref({MProd}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, dliv),
              factor(X, pnposs).
```

D.1.8 Theory 8

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLand
  pLiv
  pNposs
Theory Rules :
  <{pLand}, P>
  <{pLiv, pLand}, P>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv, pLand}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
  pref(<{pNposs}, D>, <{pLiv}, P>)
       <|Arbitrary Rule Preference|>
Theory Value Preferences :
  valpref({LLit}, {MProd})
  valpref({MProd, MSec}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pland),
              factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, pnposs).
```

D.1.9 Theory 9

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  dLiv
  pLiv
  pNposs
Theory Rules :
  <{dLiv, pNposs}, D>
  <{dLiv}, D>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{dLiv, pNposs}, D>, <{pLiv}, P>)
       <|From Value Preference|>
  pref(<{pLiv}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
Theory Value Preferences :
  valpref({LLit, MProd}, {MProd})
  valpref({MProd}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pliv).
young   | d | outcome(X, d) :- factor(X, dliv),
              factor(X, pnposs).
```

D.1.10 Theory 10

```
Theory Cases :
  <Keeble, {pLiv, pLand, pNposs}, P>
  <Pierson, {pNposs}, D>
  <Young, {pLiv, pNposs, dLiv}, D>
Theory Factors :
  pLand
  pLiv
  pNposs
Theory Rules :
  <{pLand}, P>
  <{pLiv, pLand}, P>
  <{pLiv}, P>
  <{pNposs}, D>
Theory Preferences :
  pref(<{pLiv, pLand}, P>, <{pNposs}, D>)
       <|<Keeble, {pLiv, pLand, pNposs}, P>|>
  pref(<{pLiv}, P>, <{pNposs}, D>)
       <|From Value Preference|>
Theory Value Preferences :
  valpref({MProd, MSec}, {LLit})
  valpref({MProd}, {LLit})

pierson | d | outcome(X, d) :- factor(X, pnposs).
keeble  | p | outcome(X, p) :- factor(X, pland),
              factor(X, pliv).
young   | p | outcome(X, p) :- factor(X, pliv).
```

D.2 US Trade Secrets Results

D.2.1 Original results

Table D.1: Complete AGATHA Results for all the cases using Background 1

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	1	1	8	24
Boeing	52	4	28	28
Bryce	30	4	28	28
CMI	2	2	10	24
CollegeWatercolour	4	2	4	23
Den-Tal-EZ	104	6	28	28
DigitalDevelopment	30	4	28	28
Emery	7	2	27	25
FMC	2	2	9	23
Ferranti	4	2	8	24
Forrest	104	6	28	28
KG	6	2	19	25
Laser	52	4	28	28
Lewis	104	6	28	28
MBL	2	2	8	24
Mason	106	6	28	28
MineralDeposits	31	4	19	26
NationalRejectors	6	2	18	25
Reinforced	104	6	28	28
Robinson	22	4	18	25
Scientology	2	2	19	14
Sheets	6	2	16	22
SpaceAero	22	4	19	26
Technicon	8	3	27	28
Televation	8	3	27	28
Trandes	4	2	10	23
Valco-Cincinnati	52	4	28	28
Yokana	2	2	10	24

Table D.2: Complete AGATHA Results for all the cases using Background 2

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	2	1	8	24
Boeing	89	4	28	28
Bryce	108	6	29	29
CMI	4	2	10	24
CollegeWatercolour	32	4	12	24
Den-Tal-EZ	104	6	28	28
DigitalDevelopment	37	4	28	28
Emery	33	5	29	28
FMC	5	2	20	13
Ferranti	6	2	8	24
Forrest	433	8	29	29
KG	8	2	19	25
Laser	89	4	28	28
Lewis	329	7	29	29
MBL	4	2	8	24
Mason	653	8	30	30
MineralDeposits	289	7	23	27
Reinforced	344	8	29	29
Robinson	253	6	25	27
Scientology	4	2	19	14
Sheets	74	5	26	29
Technicon	12	3	27	28
Televation	31	4	29	29
Trandes	6	2	10	23
Valco-Cincinnati	437	6	29	29
Yokana	3	2	11	27

Table D.3: Complete AGATHA Results for all the cases using Background 3

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	3	1	9	24
Boeing	118	4	28	28
Bryce	176	8	29	29
CollegeWatercolour	32	4	15	24
Den-Tal-EZ	146	6	28	28
DigitalDevelopment	65	4	28	28
Emery	37	5	29	28
FMC	9	2	20	13
Ferranti	8	2	9	24
Forrest	614	8	29	29
KG	46	5	25	25
Laser	787	6	28	29
Lewis	329	7	29	29
MBL	52	5	26	27
Mason	2855	10	30	30
MineralDeposits	587	7	23	29
Reinforced	613	9	29	29
Robinson	520	6	25	27
Scientology	11	2	24	27
Sheets	103	5	26	29
Technicon	121	7	28	29
Televation	283	7	29	30
Valco-Cincinnati	3511	8	30	30
Yokana	5	2	11	27

Appendix E

Cooperative Results

E.1 User defined background

E.1.1 A* results

Table E.1: A* AGATHA Results for all the cases using Background 1

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	1	1	8	24
Boeing	6	2	25	28
Bryce	6	2	19	28
CMI	2	2	10	24
CollegeWatercolour	4	2	4	23
Den-Tal-EZ	8	2	19	28
DigitalDevelopment	6	2	19	28
Emery	5	2	25	28
FMC	2	2	9	23
Ferranti	4	2	8	24
Forrest	8	2	19	28
KG	6	2	19	25
Laser	6	2	25	28
Lewis	8	2	19	28
MBL	2	2	8	24
Mason	8	2	27	28
MineralDeposits	8	3	19	26
NationalRejectors	6	2	18	25
Reinforced	8	2	19	28
Robinson	8	3	18	25
Scientology	2	2	19	14
Sheets	6	2	16	22
SpaceAero	9	3	19	26
Technicon	5	2	27	28
Televation	5	2	27	28
Trandes	4	2	10	23
Valco-Cincinnati	6	2	25	28
Yokana	2	2	10	24

Table E.2: A* AGATHA Results for all the cases using Background 2

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	2	1	8	24
Boeing	8	2	25	28
Bryce	9	3	20	29
CMI	4	2	10	24
CollegeWatercolour	5	2	12	17
Den-Tal-EZ	8	2	19	28
DigitalDevelopment	7	2	19	28
Emery	10	3	28	28
FMC	3	2	9	23
Ferranti	6	2	8	24
Forrest	12	3	20	29
KG	8	2	19	25
Laser	8	2	25	28
Lewis	12	3	20	29
MBL	4	2	8	24
Mason	27	5	29	28
MineralDeposits	17	4	20	25
Reinforced	12	3	20	29
Robinson	17	3	25	27
Scientology	4	2	19	14
Sheets	15	4	26	29
Technicon	7	2	27	28
Televation	10	3	29	29
Trandes	6	2	10	23
Valco-Cincinnati	15	3	28	29
Yokana	3	2	11	27

Table E.3: A* AGATHA Results for all the cases using Background 3

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	3	1	9	24
Boeing	9	2	25	28
Bryce	12	3	21	28
CollegeWatercolour	5	2	12	17
Den-Tal-EZ	12	3	21	28
DigitalDevelopment	8	2	19	28
Emery	12	3	28	28
FMC	4	2	9	23
Ferranti	8	2	9	24
Forrest	13	3	20	29
KG	13	3	25	23
Laser	16	3	27	29
Lewis	12	3	20	29
MBL	9	3	24	27
Mason	36	5	29	29
MineralDeposits	26	4	20	25
Reinforced	16	3	21	28
Robinson	29	3	25	27
Scientology	6	2	24	27
Sheets	20	3	26	29
Technicon	15	3	28	29
Televation	15	3	29	29
Valco-Cincinnati	19	3	28	30
Yokana	5	2	11	27

E.2 Entire background

E.2.1 Is it better to use the Most on Point cases or all the cases?

Table E.4: A* AGATHA Results for all the cases when AGATHA is limited to using only the Most-On-Point background cases for each seed case.

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	1	1	9	23
Boeing	7	2	22	23
Bryce	10	2	8	21
CMI	11	2	23	26
CollegeWatercolour	66	4	24	24
Den-Tal-EZ	11	2	19	19
DigitalDevelopment	5	2	18	18
Ecologix	38	2	27	26
Emery	29	4	28	27
FMC	5	2	20	24
Ferranti	2	2	9	23
Forrest	9	2	13	18
KG	2	1	23	14
Laser	5	2	22	23
Lewis	9	3	22	23
MBL	11	2	23	26
Mason	49	4	25	26
MineralDeposits	11	4	25	26
NationalInstrument	16	3	19	23
NationalRejectors	6	3	26	25
Reinforced	3	2	13	17
Robinson	27	7	25	25
Sandlin	1	1	11	23
Scientology	5	2	23	26
Sheets	17	4	16	22
SpaceAero	5	2	22	23
Technicon	102	9	28	25
Televation	12	4	28	24
Trandes	3	2	19	22
Valco-Cincinnati	9	3	22	22
Yokana	3	2	11	26

E.2.2 All cases used in the background

Table E.5: A* AGATHA Results for the cases when AGATHA can use all the background cases.

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	8	1	9	23
Boeing	90	3	28	28
Bryce	41	3	26	27
CMI	301	6	28	26
CollegeWatercolour	116	5	24	24
Den-Tal-EZ	42	3	26	27
DigitalDevelopment	36	3	26	27
Ecologix	85	2	27	26
FMC	85	5	25	27
Ferranti	20	2	9	23
Forrest	48	3	19	28
KG	63	3	23	25
Laser	1233	11	30	28
Lewis	376	7	20	27
MBL	2950	9	28	27
Mason	863	12	27	28
MineralDeposits	281	7	25	26
NationalInstrument	710	7	26	27
NationalRejectors	82	4	28	27
Reinforced	52	3	20	27
Robinson	936	8	26	26
Sandlin	10	1	11	23
Sheets	230	5	25	28
SpaceAero	480	8	25	24
Technicon	1007	14	28	28
Televation	130	6	28	28
Trandes	249	4	22	23
Valco-Cincinnati	176	8	30	28
Yokana	27	3	12	26

E.2.3 Is the cost of the moves correct? Modifying the g(n) values for the *Counter with Case* and *Distinguish with Case* Moves

Table E.6: A* AGATHA Results for all the cases when the values for the *Counter* and *Distinguish with Case* moves are the same.

Project Name	No. of Nodes	Tree Depth	Best Results	
			Explanatory	Completion
Arco	8	1	9	23
Boeing	180	4	28	28
Bryce	40	3	26	27
CMI	301	6	28	27
CollegeWatercolour	100	5	24	24
Den-Tal-EZ	41	3	27	28
DigitalDevelopment	36	3	27	28
Ecologix	85	2	27	26
FMC	125	6	28	27
Ferranti	20	2	9	23
Forrest	47	3	27	28
KG	142	4	28	25
Lewis	87	3	26	27
MBL	3124	9	28	27
Mason	419	9	29	29
MineralDeposits	215	7	25	26
NationalInstrument	724	7	27	27
NationalRejectors	264	6	28	27
Reinforced	53	3	27	28
Robinson	582	6	23	25
Sandlin	10	1	11	23
Scientology	2640	9	28	26
Sheets	127	5	28	28
SpaceAero	609	8	26	26
Technicon	304	5	28	28
Televation	85	3	28	28
Valco-Cincinnati	432	7	30	28
Yokana	49	3	19	21

Appendix F

Adversarial Results

F.1 User Selected Case Background

Table F.1: $\alpha\beta$ Results for all the cases when using background 1.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Arco	1	1	0	0	8	24
Boeing	12	4	27	28	19	24
Bryce	10	4	28	28	13	21
CMI	2	2	16	15	10	24
CollegeWatercolour	4	2	6	13	4	23
Den-Tal-EZ	19	5	28	28	19	24
DigitalDevelopment	10	4	28	28	13	21
Emery	5	2	25	28	9	23
FMC	2	2	17	12	93	23
Ferranti	4	2	8	14	8	24
Forrest	19	5	28	28	19	24
KG	6	2	19	11	19	25
Laser	12	4	27	28	19	24
Lewis	19	5	28	28	19	26
MBL	2	2	16	11	8	24
Mason	19	5	27	28	27	26
MineralDeposits	11	4	17	19	19	26
NationalRejectors	6	2	20	12	6	24
Reinforced	19	5	28	28	19	24
Robinson	11	4	18	20	18	25
Scientology	2	2	19	14	8	24
Sheets	6	2	15	20	8	24
SpaceAero	11	4	19	19	19	26
Technicon	7	3	27	28	27	27
Televation	7	3	27	28	26	26
Trandes	4	2	12	13	10	23
Valco-Cincinnati	12	4	27	28	19	24
Yokana	2	2	10	11	10	24

Table F.2: $\alpha\beta$ Results for all the cases when using background 2.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Arco	2	1	0	0	8	24
Boeing	16	4	27	28	19	24
Bryce	21	6	28	29	15	22
CMI	4	2	16	15	10	24
CollegeWatercolour	11	4	14	17	4	23
Den-Tal-EZ	19	5	28	28	19	24
DigitalDevelopment	11	4	28	28	13	21
Emery	11	3	29	28	21	23
FMC	4	2	20	13	9	21
Ferranti	6	2	8	14	8	24
Forrest	32	7	29	29	20	25
KG	8	2	22	18	6	23
Laser	16	4	27	28	19	24
Lewis	25	6	28	29	20	27
MBL	4	2	16	11	8	24
Mason	39	7	30	30	29	28
MineralDeposits	23	5	19	19	23	27
Reinforced	32	7	29	29	20	25
Robinson	28	6	21	21	25	27
Scientology	4	2	19	14	8	24
Sheets	20	5	20	20	26	29
Technicon	10	3	27	28	27	27
Televation	14	4	29	29	29	27
Trandes	6	2	18	13	9	23
Valco-Cincinnati	28	6	25	28	25	28
Yokana	3	2	10	11	10	24

Table F.3: $\alpha\beta$ Results for all the cases when using background 3.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Arco	3	1	0	0	9	24
Boeing	17	4	27	28	19	24
Bryce	32	8	28	29	15	22
CollegeWatercolour	11	4	14	17	4	23
Den-Tal-EZ	23	6	28	28	19	24
DigitalDevelopment	23	6	28	28	19	24
Emery	13	3	25	28	25	28
FMC	5	2	18	13	9	23
Ferranti	8	2	8	14	8	24
Forrest	38	8	29	29	20	25
KG	17	5	25	23	24	25
Laser	32	6	27	29	25	25
Lewis	25	6	28	29	20	27
MBL	11	3	19	17	24	27
Mason	60	9	29	30	30	30
MineralDeposits	28	5	19	19	23	27
Reinforced	43	9	27	29	20	25
Robinson	34	6	21	21	25	27
Scientology	7	2	19	17	24	27
Sheets	24	5	22	20	26	29
Technicon	29	7	27	29	28	27
Televation	37	7	29	30	29	27
Valco-Cincinnati	37	7	28	30	26	26
Yokana	5	2	10	11	10	24

F.2 Entire Case Background

Table F.4: $\alpha\beta$ Results for all the cases when AGATHA is limited to using only the Most-On-Point background cases for each seed case.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Arco	1	1	0	0	9	23
Boeing	10	4	18	18	22	23
Bryce	80	20	22	26	19	19
CMI	50	14	27	25	29	25
CollegeWatercolour	119	15	20	20	24	24
Den-Tal-EZ	63	16	20	24	17	17
DigitalDevelopment	5	2	22	27	8	21
Ecologix	52	24	27	26	27	26
Emery	35	12	27	27	25	23
FMC	29	12	28	18	28	27
Ferranti	2	2	9	11	9	23
Forrest	10	6	18	18	8	21
KG	2	1	23	14	0	0
Laser	34	12	26	28	26	25
Lewis	68	23	25	28	23	24
MBL	28	6	20	18	24	26
Mason	98	12	27	28	27	26
MineralDeposits	40	14	23	21	25	25
NationalInstrument	78	20	25	27	23	23
NationalRejectors	17	8	22	22	27	27
Reinforced	3	2	22	27	28	21
Robinson	56	19	22	22	27	27
Sandlin	1	1	0	0	11	23
Scientology	12	6	28	18	23	26
Sheets	15	12	23	18	26	22
SpaceAero	47	19	20	22	24	25
Technicon	47	10	28	25	27	24
Televation	17	6	28	25	28	24
Trandes	15	8	16	17	23	24
Valco-Cincinnati	17	6	24	24	24	23
Yokana	5	3	12	15	11	26

Table F.5: $\alpha\beta$ Results for all the cases when AGATHA can use all the background cases.

Project Name	Nodes	Depth	Plaintiff		Defendant	
			Explan	Comp	Explan	Comp
Arco	8	1	0	0	9	23
Boeing	477	26	28	28	24	24
Bryce	374	34	28	28	22	21
CMI	427	29	28	25	28	26
CollegeWatercolour	181	16	20	20	24	24
Den-Tal-EZ	249	27	28	28	26	26
DigitalDevelopment	65	7	27	28	14	22
Ecologix	69	24	27	26	27	26
FMC	348	22	28	18	28	27
Ferranti	20	2	8	14	8	23
Forrest	118	11	29	29	22	26
KG	186	20	27	26	26	25
Laser	655	32	27	29	26	26
Lewis	328	24	27	28	24	24
MBL	198	28	28	25	28	27
Mason	779	39	30	31	30	30
MineralDeposits	180	15	18	18	27	26
NationalInstrument	268	26	28	28	25	25
NationalRejectors	346	26	27	24	29	28
Reinforced	218	23	28	29	24	24
Robinson	357	21	22	22	27	27
Sandlin	10	1	0	0	11	23
Scientology	503	2	28	22	28	27
Sheets	75	13	26	19	28	28
SpaceAero	333	20	25	24	27	27
Technicon	439	20	29	29	28	26
Trandes	364	27	24	24	26	25
Valco-Cincinnati	519	31	29	30	28	28
Yokana	36	4	18	14	19	21

Appendix G

Argumentation Dialogues

G.1 A* Dialogues

G.1.1 Mason dialogues and theories

Table G.1 shows the winning dialogues for Mason when using A* search. There are five dialogues producing theories which obtain results of 27 cases correct out of 32 cases when using the factors produced in constructing the theory and 28 cases correct out of 32 cases for the full set of available factors. The five dialogues are divided into two distinct types, with one set having all the same moves until the last move which is won by the plaintiff, and the other three have different starting moves and then the rest of the dialogue is identical and the winner is the defendant.

The dialogue for Theory 499 for Mason is shown in Table G.2. It starts with the Plaintiff player using the *Analogise* move with the College Watercolor case, which gives the rule preference of {*(F15-Unique-Product)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant player responds by *Distinguishing* the College Watercolor case by including the plaintiff factor of *F26-Deception* to modify the rule preference to {*(F15-Unique-Product, F26-Deception)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant player then *Analogises* National Rejectors to give the rule preference {*(F16-Info-Reverse-Engineerable)→D*} > {*(F15-Unique-Product)→P*}.

The Plaintiff player then responds by *Distinguishing* National Rejectors which modifies the rule preference to {*(F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-*

Table G.1: Mason dialogues produced using the A* Heuristic

Theory 499	*Theory 500*	*Theory 423*	*Theory 450*	*Theory 474*
P Cite College Watercolor		P Cite Digital Development	P Cite Reinforced	P Cite Valco-Cincinnati
D Distinguish National Rejectors		D Distinguish National Rejectors		
P Distinguish National Instrument		P Distinguish National Instrument		
D Distinguish CMI		D Distinguish Sandlin		
P Counter Technicon		P Distinguish Lewis		
D Distinguish Sandlin		D Distinguish CMI		
P Distinguish Goldberg	P Distinguish Lewis			

Table G.2: Explanation of Theory 499 dialogue.

Theory Move	*Factors in Cited Case*	*Rules Produced*
P Cite College Watercolor	F1, F15, F26	(F15)>(F1)
D Distinguish with National Rejectors	F7, F10, F15, F16, F18, F19, F27	(F15, F26)>(F1) (F16)>(F15)
P Distinguish with National Instrument	F1, F18, F21	(F10, F16, F19, F27)>(F15) (F21)>(F1)
D Distinguish with CMI	F4, F6, F10, F16, , F17, F20, F27	(F18, F21)>(F1) (F16)>(F6)
P Counter with Technicon	F6, F10, F12, F14, F16, F21, F25	(F6, F21)>(F16)
D Distinguish with Sandlin	F1, F10, F16, F19, F27	(F6, F12, F14, F21)>(F16) (F1, F16)>()
P Distinguish with Goldberg	F1, F10, F21, F27	(F1, F10, F16, F19, F27)>() (F21)>(F1)

Engineerable, F19-No-Security-Measures)→D}>{*(F15-Unique-Product)→P*} and *Analogises* National Instrument to include the rule preference {*(F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant now responds by *Distinguishing* National Instrument and *Analogising* CMI to include the modified rule preference {*(F18-Identical-Products, F21-Knew-Info-Confidential)→P*} >{*(F1-Disclosure-in-Negotiation)→D*} and the new rule preference {*(F16-Info-Reverse-Engineerable)→D*}>{*(F6-Security-Measures)→P*}.

The Plaintiff has a more-on-point case, Technicon, which they use to *Counter* CMI, removing the rule preference associated with CMI and replacing it with {*(F6-Security-Measures, F21-Knew-Info-Confidential)→P*}>{*(F16-Info-Reverse-Engineerable)→D*}. The Defendant responds by *Distinguishing* Technicon and *Analogising* with Sandlin. This modifies the rule preference for Technicon to {*(F6-Security-Measures, F12-Outsider-Disclosures-Restricted, F14-Restricted-Materials-Used, F21-Knew-Info-Confidential)→P*}>{*(F16-Info-Reverse-Engineerable)→D*} but because Sandlin only has defendant factors it cannot produce a rule preference and so only produces the complex rule {*(F1-Disclosure-in-Negotiation, F16-Info-Reverse-Engineerable)→D*}.

Finally the Plaintiff player *Distinguishes* Sandlin which only modifies the complex rule to {*(F1-Disclosure-in-Negotiation, F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F19-No-Security-Measures, F27-Disclosure-in-Public-*

Forum)→D} and then *Analogises* Goldberg to include the final rule preference of {*(F21-Knew-Info-Confidential)→P*} > {*(F1-Disclosure-in-Negotiation)→D*}. This rule preference was included earlier in the dialogue by National Instrument but this was then *Distinguished* by the Defendant. The Defendant cannot respond this time because either there are no more moves he can make or the subsequent theory is not as good.

Table G.3: Explanation of Theory 423 dialogue.

Theory Move	*Factors in Cited Case*	*Rules Produced*
P Cite Digital Development	F1, F6, F8, F15, F18, F21	(F6, F15, F21)>(F1)
D Distinguish with National Rejectors	F7, F10, F15, F16, F18, F19, F27	(F6, F8, F15, F18, F21)>(F1) (F16)>(F15)
P Distinguish with National Instrument	F1, F18, F21	(F10, F16, F19, F27)>(F15) (F21)>(F1)
D Distinguish with Sandlin	F1, F10, F16, F19, F27	(F8, F21)>(F1) (F1, F16)>()
P Distinguish with Lewis	F1, F8, F21	(F1, F10, F16, F19, F27)>() (F21)>(F1)
D Distinguish with CMI	F4, F6, F10, F16, , F17, F20, F27	(F8, F21)>(F1) (F16)>(F6)

The dialogue for Theory 423 for Mason is shown in Table G.3. It starts with the Plaintiff player using the *Analogise* move with the Digital Development case, which gives the rule preference of {*(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant player responds by *Distinguishing* the Digital Development case by including all the plaintiff factors to modify the rule preference to {*(F6-Security-Measures, F8-Competitive-Advantage, F15-Unique-Product, F18-Identical-Products, F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant player then *Analogises* National Rejectors to give the rule preference {*(F16-Info-Reverse-Engineerable)→D*}>{*(F15-Unique-Product)→P*}.

The Plaintiff player then responds by *Distinguishing* National Rejectors which modifies the rule preference to {*(F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F19-No-Security-Measures)→D*}>{*(F15-Unique-Product)→P*} and *Analogises* National Instrument to include the rule preference {*(F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant now responds in a different way from Theory 499 by *Distinguishing* National Instrument and *Analogising* Sandlin instead of CMI to include the modified rule preference {*(F18-Identical-Products, F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*} and the complex rule {*(F1-Disclosure-in-Negotiation, F16-Info-Reverse-Engineerable)→D*}.

The Plaintiff player *Distinguishes* Sandlin which only modifies the complex rule to {*(F1-Disclosure-in-Negotiation, F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F19-No-Security-Measures, F27-Disclosure-in-Public-Forum)→D*} and then *Analogises* Lewis to include the rule preference of {*(F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*}. The Defendant responds by *Distinguishing* Lewis and *Analogising* with CMI to include the modified rule of {*(F18-Identical-Products, F21-Knew-Info-Confidential)→P*}>{*(F1-Disclosure-in-Negotiation)→D*} and the final rule of {*(F16-Info-Reverse-Engineerable)→D*}>{*(F6-Security-Measures)→P*}. This rule preference has not been used in the dialogue and the Plaintiff cannot respond because either there are no more moves he can make or the subsequent theory is not as good. This means that the Defendant player wins even though it is contrary to the actual case decision.

G.1.2 Digital Development Dialogues and theories

The Digital Development dialogue is shown in the top rows of Table G.4. This dialogue produces two theories both won by the plaintiff, which obtain 28 cases out of 32 for selected factors and 26 cases out of 32 for the full set of factors.

For the first dialogue the Defendant starts the dialogue by *Analogising* with the Robinson case. The Plaintiff player then responds by *Distinguishing* Robinson and then *Analogising* with the Goldberg case. The Defendant cannot respond and so the Plaintiff player wins.

For the second dialogue the Plaintiff starts its dialogue by *Analogising* with Mineral Deposits. The Defendant can respond by *Countering* with the Robinson case. The plaintiff then responds by *Distinguishing* Robinson and *Analogising* with the Goldberg case. Again the Defendant player cannot respond and so the plaintiff player wins.

G.1.3 CMI dialogues and theories

The CMI dialogues are shown at the bottom of Table G.4. There are four dialogues which obtain the result of 28 cases out of 32 for selected factors and 26 cases out of 32 for the full set of factors. The dialogues all use the same cases in the moves but AGATHA uses the moves in different orders. All the dialogues start with the plaintiff player *Analogising* with the Mason case and end with the defendant player *Distinguishing the Problem* and winning.

G.1.4 Discussion

Because Mason is a well balanced case and contains factors that are present in many of the background cases AGATHA can use many more moves to create longer dialogues which continually refine the theory to produce a very good theory which can decide a large proportion of the background cases correctly.

Digital Development also contains factors which are present in many of the background cases but it is very unbalanced as it only contains one Defendant factor. This limits the number of moves that the Defendant can make and so the dialogue is small and there are not many refinements that can be made by the Defendant to the theory to improve it. In consequence the Plaintiff is not required to refine his theory beyond what is necessary to meet these limited objections.

Table G.4: Digital Development and CMI dialogues produced using the A* Heuristic

Digital Development Dialogues	
Theory 25	*Theory 35*
D Cite Robinson	P Cite Mineral Deposists
P Distinguish Goldberg	D Counter Robinson
	P Distinguish Goldberg

CMI Dialogues			
Theory 290	*Theory 291*	*Theory 292*	*Theory 293*
P Cite Mason	P Cite Mason	P Cite Mason	P Cite Mason
D Distinguish MBL	D Distinguish MBL	D Distinguish Scientology	D Distinguish Scientology
P Distinguish Laser	P Distinguish Valco-Cincinnati	P Distinguish Laser	P Distinguish Valco-Cincinnati
D Counter Scientology	D Counter Scientology	D Counter MBL	D Counter MBL
P Distinguish Valco-Cincinnati	P Distinguish Laser	P Distinguish Valco-Cincinnati	P Distinguish Laser
D Problem distinguish	D Problem distinguish	D Problem distinguish	D Problem distinguish

CMI is a strongly pro-defendant case but it has two plaintiff factors found in many cases so the plaintiff player can respond to the moves made by the defendant player and hence the dialogues are longer than those produced for Digital Development. This forces the defendant player to refine the theory more to be able to win.

G.2 Adversarial Dialogues in AGATHA

All the adversarial dialogues are longer than those obtained using A* search. This is because the *Distinguish with Arbitrary Preference* can be used much more and especially by the Defendant player. We might expect this: a co-operative opponent engaged in seeking the best solution will ground his objections in cases, but in so doing invites more powerful responses. When a player is also trying to give as few opportunities to his opponent as possible, however, hypothetical objections, not grounded in any cases, can be used to obstruct the deployment of these powerful cases.

G.2.1 Mason dialogues and theories

When using adversarial search in AGATHA, we get the dialogues shown in Table G.5. There are five winning theories with the Plaintiff and Defendant players obtaining the same results. Three of the dialogues are almost identical and use the same cases but they differ where each dialogue uses the starting move of the other two dialogues. The Plaintiff player creates their best theory by *Analogising* with College Watercolor and the Defendant responds by *Distinguishing the Problem case* of Mason. Towards the end of the dialogues the Defendant player resorts to using *Arbitrary Preferences* which the Plaintiff player responds to by introducing new cases.

The Mason dialogue for Theories 481 and 503 is shown in Table G.6. The plaintiff player starts the dialogue by *Analogising* the Boeing case to the problem case of Digital Development to produce the rule preference $\{$*(F6-Security-Measures, F21-Knew-Info-Confidential)*$\rightarrow$*P*$\}$ > $\{$*(F1-Disclosure-in-Negotiation)*$\rightarrow$*D*$\}$. The defendant responds by *Distinguishing* with the Sandlin case which modifies the previous rule preference to $\{$*(F4-Agreed-not-to-Disclose, F6-Security-Measures, F12-Outsider-Disclosures-Restricted, F14-Restricted-Materials-Used, F21-Knew-Info-Confidential)*$\rightarrow$*P*$\}$ > $\{$*(F1-Disclosure-in-Negotiation)*$\rightarrow$*D*$\}$ and introduces the complex rule of $\{$*(F1-Disclosure-in-Negotiation, F16-Info-Reverse-*

Table G.5: Mason dialogues produced using the $\alpha\beta$ heuristic

<table>
<tr><th>Theories 481 and 503</th><th>Theories 482 and 504</th><th>Theories 483 and 505</th><th>Theories 488 and 510</th><th>Theories 495 and 517</th></tr>
<tr><td>P Cite Boeing</td><td>P Cite Bryce</td><td>P Cite Den-Tal-Ez</td><td>P Cite Laser</td><td>P Cite Trandes</td></tr>
<tr><td colspan="3">D Distinguish Sandlin</td><td>D Distinguish Arco</td><td>D Distinguish Sandlin</td></tr>
<tr><td colspan="3">P Distinguish Trandes</td><td>P Distinguish Trandes</td><td>P Distinguish Lewis</td></tr>
<tr><td colspan="3">D Distinguish Robinson</td><td>D Distinguish Sandlin</td><td>D Distinguish Arco</td></tr>
<tr><td colspan="3">P Counter Lewis</td><td>P Distinguish Lewis</td><td>P Distinguish Boeing</td></tr>
<tr><td colspan="3">D Distinguish Arco</td><td>D Distinguish Yokana</td><td>D Distinguish Yokana</td></tr>
<tr><td>P Distinguish Bryce</td><td colspan="2">P Distinguish Boeing</td><td>P Distinguish National Instrument</td><td>P Distinguish Bryce</td></tr>
<tr><td colspan="3">D Distinguish Yokana</td><td>D Distinguish CMI</td><td>D Distinguish Robinson</td></tr>
<tr><td colspan="2">P Distinguish Den-Tal-EZ</td><td>P Distinguish Bryce</td><td>P Distinguish Den-Tal-EZ</td><td>P Distinguish Den-Tal-EZ</td></tr>
<tr><td colspan="3">D Distinguish CMI</td><td>D Distinguish National Rejectors</td><td>D Distinguish CMI</td></tr>
<tr><td colspan="3">P Distinguish National Instrument</td><td>P Distinguish Valco-Cincinnati</td><td>P Distinguish National Instrument</td></tr>
<tr><td colspan="3">D Distinguish National Rejectors</td><td>D Distinguish Arb Pref</td><td>D Distinguish National Rejectors</td></tr>
<tr><td colspan="3">P Distinguish Laser</td><td>P Cite Emery</td><td>P Distinguish Laser</td></tr>
<tr><td colspan="3">D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td></tr>
<tr><td colspan="3">P Cite Valco-Cincinnati</td><td>P Cite FMC</td><td>P Cite Valco-Cincinnati</td></tr>
<tr><td colspan="3">D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td></tr>
<tr><td colspan="3">P Cite Emery</td><td>P Cite Boeing</td><td>P Cite Emery</td></tr>
<tr><td colspan="3">D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td></tr>
<tr><td colspan="3">P Cite FMC</td><td>P Cite Bryce</td><td>P Cite FMC</td></tr>
<tr><td colspan="3">D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td><td>D Distinguish Arb Pref</td></tr>
<tr><td colspan="3">P Cite College Watercolour</td><td>P Cite College Watercolour</td><td>P Cite College Watercolour</td></tr>
<tr><td colspan="3">D Problem distinguish</td><td>D Problem distinguish</td><td>D Problem distinguish</td></tr>
</table>

Table G.6: Explanation of Adversarial Mason dialogue using Theories 481 and 503.

Theory Move	*Factors in Cited Case*	*Rules Produced*
P Cite Boeing	F1, F4, F6, F10, F12, F14, F21	(F6, F15, F21)>(F1)
D Distinguish with Sandlin	F1, F10, F16, F19, F27	(F4, F6, F12,F14, F21)>(F1) (F1, F16)>()
P Distinguish with Trandes	F1, F4, F6, F10, F12	(F1, F10, F16, F19, F27)>() (F6)>(F1)
D Distinguish with Robinson	F1, F10, F18, F19, F26	(F4, F6, F12,)>(F1) (F1)>()
P Counter with Lewis	F1, F8, F21	(F21)>(F1)
D Distinguish with Arco	F10, F16, F20	(F8, F21)>(F1) (F16)>()
P Distinguish with Bryce	F1, F4, F6, F18, F21	(F10, F16, F20)>() (F6, F21)>(F1)
D Distinguish with Yokana	F7, F10, F16, F27	(F4, F6, F18, F21)>(F1) (F16)>()
P Distinguish with Den-Tal-EZ	F1, F4, F6, F21, F26	(F10, F16, F27)>() (F6, F21)>(F1)
D Distinguish with CMI	F4, F6, F10, F16, F17, F20, F27	(F4, F6, F21, F26)>(F1) (F16)>(F6)
P Distinguish with NationalInstrument	F1, F18, F21	(F10, F16, F17, F20, F27)>(F6) (F21)>(F1)
D Distinguish with NationalRejectors	F7, F10, F15, F16, F18, F19, F27	(F8, F21)>(F1) (F16)>(F15)
P Distinguish with Laser	F1, F6, F10, F12, F21	(F10, F16, F19, F27)>(F15) (F6, F21)>(F1)
D Distinguish with Arb Pref		(F6, F12, F21)>(F1) (F1, F16)>(F6, F15, F21)
P Cite Valco-Cincinnati	F1, F6, F10, F12, F15, F21	(F6, F15, F21)>(F1)
D Distinguish with Arb Pref		(F6, F12, F15, F21)>(F1) (F1, F16)>(F6, F15, F21)
P Cite Emery	F10, F18, F21	(F21)>()
D Distinguish with Arb Pref		(F18, F21)>() (F1, F16)>(F6, F15, F21)
P Cite FMC	F4, F6, F7, F10, F11, F12	(F6)>()
D Distinguish with Arb Pref		(F4, F6, F7, F12)>() (F1, F16)>(F6, F15, F21)
P Cite CollegeWatercolor	F1, F15, F26	(F15)>(F1)
D Problem distinguish		(F1, F16)>(F6, F15, F21)

Engineerable)→D}.

The plaintiff player can now respond by *Distinguishing* Sandlin with the Trandes case to modify the previous rule preference to *{(F1-Disclosure-in-Negotiation, F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F19-No-Security-Measures, F27-Disclosure-in-Public-Forum)→D}* and introduce the new rule preference of *{(F6-Security-Measures)→P}>{(F1-Disclosure-in-Negotiation)→D}*. Now the defendant can *Distinguish* with the Robinson case to modify the rule preference to *{(F4-Agreed-not-to-Disclose, F6-Security-Measures, F12-Outsider-Disclosures-Restricted)→P} > {(F1-Disclosure-in-Negotiation)→D}* and to introduce the rule preference of *{(F1-Disclosure-in-Negotiation)→D}*.

The plaintiff player now *Counters* Robinson with the more-on-pointcase of Lewis. This changes the rule preference to *{(F21-Knew-Info-Confidential)→P}>{(F1-Disclosure-in-Negotiation)→D}*. The defendant responds by *Distinguishing* with Arco to change the rule preference to *{(F8-Competitive-Advantage, F21-Knew-Info-Confidential)→P}>{(F1-Disclosure-in-Negotiation)→D}*.

However as Arco only has defendant factors only the simple rule of *{(F16-Info-Reverse-Engineerable)→D}* can be included which the plaintiff can distinguish using Bryce to introduce a complex rule, *{(F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F20-Info-Known-to-Competitors)→D}*, to explain Arco and to introduce the rule preference of *{(F6-Security-Measures, F21-Knew-Info-Confidential)→P}>{(F1-Disclosure-in-Negotiation)→D}* which was introduced when the plaintiff player started the dialogue.

The defendant now *Distinguishes* Bryce by modifying the rule preference to *{(F4-Agreed-not-to-Disclose, F6-Security-Measures, F18-Identical-Products, F21-Knew-Info-Confidential)→P}>{(F1-Disclosure-in-Negotiation)→D}* and using Yokana, but again as the case only contains defendant factors only the simple rule of *{(F16-Info-Reverse-Engineerable)→D}* can be included.

The plaintiff player *Distinguishes* Yokana with Den-Tal-Ez to include the complex rule of *{(F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F27-Disclosure-in-Public-Forum)→D}* and to introduce the rule preference *{(F6-Security-Measures, F21-Knew-Info-Confidential)→P}>{(F1-Disclosure-in-Negotiation)→D}* again.

The defendant now *Distinguishes* with CMI to modify the previous rule preference to {*(F4-Agreed-not-to-Disclose, F6-Security-Measures, F21-Knew-Info-Confidential, F26-Deception)*→*P*} > {*(F1-Disclosure-in-Negotiation)*→*D*} and to include the new rule preference of {*(F16-Info-Reverse-Engineerable)*→*D*} > {*(F6-Security-Measures)*→*P*}.

The plaintiff can now *Distinguish* with National Instrument to modify the rule preference to {*(F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F17-Info-Independantly-Generated, F20-Info-Known-to-Competitors, F27-Disclosure-in-Public-Forum)*→*D*}>{*(F6-Security-Measures)*→*P*} and include the new rule preference of {*(F21-Knew-Info-Confidential)*→*P*}>{*(F1-Disclosure-in-Negotiation)*→*D*}.

The defendant now *Distinguishes* using the case of National Rejectors to modify the rule preference to {*(F8-Competitive-Advantage, F21-Knew-Info-Confidential)*→*P*}> {*(F1-Disclosure-in-Negotiation)*→*D*} and introduce {*(F16-Info-Reverse-Engineerable)*→*D*}>{*(F15-Unique-Product)*→*P*}.

The plaintiff now *Distinguishes* with Laser to change the previous rule preference to {*(F10-Secrets-Disclosed-Outsiders, F16-Info-Reverse-Engineerable, F19-No-Security-Measures, F27-Disclosure-in-Public-Forum)*→*D*}>{*(F15-Unique-Product)*→*P*} and include the new rule preference of {*(F6-Security-Measures, F21-Knew-Info-Confidential)*→*P*}>{*(F1-Disclosure-in-Negotiation)*→*D*}.

The Defendant player now resorts to stubbornly using the *Distinguish with Arbitrary Preference* move and so *Distinguishes* the Laser case to change the rule preference to {*(F6-Security-Measures, F12-Outsider-Disclosures-Restricted, F21-Knew-Info-Confidential)*→*P*}>{*(F1-Disclosure-in-Negotiation)*→*D*} and includes the arbitrary preference of {*(F1-Disclosure-in-Negotiation, F16-Info-Reverse-Engineerable)*→*D*} > {*(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)*→*P*}.

The plaintiff player responds to the *Distinguish with Arbitrary Preference* moves by *Analogising* Valco-Cincinnati with the rule preference of {*(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)*→*P*}>{*(F1-Disclosure-in-Negotiation)*→*D*}, then *Analogising* Emery with the simple rule of {*(F21-Knew-Info-Confidential)*→*P*}, then FMC with the simple rule of {*(F6-Security-Measures)*→*P*} and finally *Analogising* with College Watercolor to introduce the rule preference of {*(F15-Unique-Product)*→*P*}>{*(F1-Disclosure-in-Negotiation)*→*D*}.

Table G.7: Digital Development and CMI dialogues produced using the $\alpha\beta$ heuristic

Digital Development Dialogues			
Theories 44 and 55	*Theories 53 and 58*	*Theories 54 and 59*	*Theories 61 and 64*
P Cite CollegeWatercolour	D Cite Robinson	D Cite Sandlin	P Cite MineralDeposits
D Distinguish Sandlin	P Distinguish CollegeWatercolour	P Distinguish CollegeWatercolour	D Counter Robinson
P Distinguish Goldberg	D Distinguish Sandlin	D Distinguish Arb Pref	P Distinguish CollegeWatercolour
D Counter Ecologix	P Distinguish Goldberg	P Cite Goldberg	D Distinguish Sandlin
	D Counter Ecologix	D Counter Ecologix	P Distinguish Goldberg
			D Counter Ecologix

CMI Dialogues	
Theories 183 and 202	*Theories 198 and 217*
P Cite Boeing	D Cite Robinson
D Counter MBL	P Counter Boeing
P Distinguish Bryce	D Counter MBL
D Distinguish Arb Pref	P Distinguish Bryce
P Cite Den-Tal-EZ	D Distinguish Arb Pref
D Distinguish Arb Pref	P Cite Den-Tal-EZ
P Cite DigitalDevelopment	D Distinguish Arb Pref
D Distinguish Arb Pref	P Cite DigitalDevelopment
P Cite FMC	D Distinguish Arb Pref
D Counter Scientology	P Cite FMC
P Distinguish Valco-Cincinnati	D Counter Scientology
	P Distinguish Valco-Cincinnati

The defendant responds to all of these moves by using the *Distinguish with Arbitrary Preference* move to distinguish the preceding rule preference and include its own rule preference of {*(F1-Disclosure-in-Negotiation, F16-Info-Reverse-Engineerable)→D*}>{*(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)→P*} until the last move where it uses the *Problem Distinguish* move but still introduces the rule preference {*(F1-Disclosure-in-Negotiation, F16-Info-Reverse-Engineerable)→D*}>{*(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)→P*}.

G.2.2 Digital Development dialogues and theories

The top of Table G.7 shows the four "best" dialogues produced when CMI is used as the seed case. the plaintiff player wins because its "best" theory gets 27 cases correct out of 32 for the the original theory and 28 cases correct out of 32 for the completed theories. Three of the dialogues use the same sequence of dialogue moves to construct the "best" theories. *REWRITE The fourth dialogue in table G.7 uses all of the second dialogue apart from it* Counters *with the Robinson case.* The second dialogue uses all of the first dialogue. This sequence uses the same four cases of College Watercolor, Sandlin, Goldberg and Ecologix and in the same order. The third dialogue is different because it uses the Sandlin case out of order and has to resort to using an *Arbitrary Preference*.

The dialogues are very short and the defendant player cannot refine the theories enough to produce a good theory. This is due to Digital Development being a strongly pro-plaintiff case with only one defendant factor. This means that the defendant struggles to make a good theory because the plaintiff player can respond to all his moves.

G.2.3 CMI dialogues and theories

The bottom of table G.7 shows the two best dialogues produced when CMI is the seed case. The "best" theories are very similar and the defendant wins by getting one more case correct when the theory is completed to get 28 cases correct out of 32 for the original theory and 26 cases correct out of 32 for the completed version of the theory.

The two dialogues are almost identical because the second dialogue starts with the defendant player *Analogising* Robinson and then the Plaintiff player *Counter* with

Boeing. Then both players follow the same sequence of dialogue moves as for the first dialogue.

CMI is a pro-defendant case but it has two plaintiff factors that are present in many of the background cases. This means that the plaintiff player can respond to the defendant moves and refine the theory to get very good results.

G.2.4 Discussion

From this we conclude that if the quality of the theory is important, it is essential to use a balanced seed case to give both sides the opportunity to develop a reasonable theory

The dialogues is Mason seem to produce plausibly motivated moves, until the defendant is forced to come up with a series of *Distinguish with Arbitrary Preference* moves. Although a real defendant is unlikely to be so persistent, nor a court to allow such unfounded objections, meeting them does refine, and improve, the theory.

G.3 Adversarial Dialogues in ROSALIND

The seed cases we used in ROSALIND each contain five factors: Mason because it is well-balanced, Bryce because it is a strong pro-plaintiff case and Ferranti because it is a strong pro-defendant case. For the dialogues taken from ROSALIND, we bias things in favour of the Plaintiff by giving the Plaintiff precedents containing a large number of factors and the Defendant precedents containing a small number of factors. This background was described in section 9.5 and is the one labelled P1.

When using ROSALIND with this case background, we get the dialogues shown in Table G.8. The theories obtained for the Plaintiff and Defendant players are identical except for the last move. Because the players are restricted to a small set of background cases, the dialogues are usually very small.

The Mason dialogues are longer than the other dialogues but only by one move, because Mason is a well balanced case and so the players have more moves that they can make. The dialogue for Mason is shown in Table G.9. For this dialogue the Defendant player starts by *Analogising* with the Arco case, relying on *F16-Info Reverse Engineerable*, its best factor. This gives a rule of $\{$*(F16-Info-Reverse-Engineerable)*$\rightarrow$*D*$\}$ which is used to decide the Mason case but because there are no Plaintiff factors in Arco a rule preference is not produced. The Plaintiff responds by *Countering* with Technicon which gives a rule preference of *({(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)→P}>{(F16-Info-Reverse-Engineerable)→D})*. Since he has no effective cases available in his selection, the Defendant can only respond by *Distinguishing* Technicon and by stating an *Arbitrary Preference* which is *({(F16-Info-Reverse-Engineerable)→D}>{(F6-Security-Measures, F15-Unique-Product, F21-Knew-Info-Confidential)→P})* and is the reverse of the rule preference which the Plaintiff used from Technicon. This arbitrary preference results in a less good theory and so the Defendant loses and the Plaintiff wins.

For Bryce the Plaintiff wins by only using one move and the Defendant has to respond using the *Distinguish with Arbitrary Preference* which again gives a less good theory. For Ferranti the Defendant wins with one move so the Plaintiff has to respond with the *Problem Distinguish* move which also results in a less good theory.

When the cases are limited in this way there are not enough moves available to refine the theories to produce a theory which is able to explain the background cases in a satisfactory way.

Table G.8: ROSALIND Dialogues.

Mason	*Bryce*	*Ferranti*
D Cite Arco P Counter with Technicon D Distinguish with Arb Pref	P Cite Boeing D Distinguish with Arb Pref	D Cite Sheets P Problem distinguish

Table G.9: Explanation of ROSALIND Mason dialogue.

Theory Move	*Factors in Cited Case*	*Rules Produced*
D Cite Arco	F10, F16, F20	(F16)>()
P Counter with Technicon	F7, F10, F15, F16, F18, F19, F27	(F6, F15, F21)>(F16)
D Distinguish with Arb Pref		(F6, F12, F15, F18, F21)>(F16) (F16)>(F6, F15, F21)

G.4 Concluding Remarks

A second aim of our work was to construct theories through the use of dialogues using techniques from standard Case Based Reasoning. Our conclusions regarding this are:

- When the cooperative heuristic search is used, we get a sequence of cases which can be explained in terms of plausible domain arguments.
- When adversarial search is used, the dialogues tend to be much longer and there is a tendency to use arbitrary preferences, hypothesising theories that are not grounded in any cases. This reflects the desire of the adversary to avoid strong moves from their opponent but in practice this delays rather than prevents the use of significant cases.
- Where the result is clear because the seed case is strongly pro-plaintiff or pro-defendant, the search can terminate with a theory which meets the current case, but which does not generalise.
- Where the background information is unbalanced the outcome of the dialogue is not much affected by which side has the better information: often a better theory is produced if the side with the better case has the worse information.

Bibliography

[1] V. Aleven. *Teaching Case Based Argumentation Through an Example and Models*. Phd thesis, The University of Pittsburgh, 1997.

[2] K. Ashley. *Modelling Legal Argument*. Bradford Books. MIT Press, Cambridge, Mass, 1990.

[3] K. Ashley and S. Brüninghaus. A predictive role for intermediate legal concepts. In D. Bourcier, editor, *Proceedings of Jurix 2003*, pages 153–62. IOS Press Amsterdam, 2003.

[4] T. Bench-Capon. Knowledge based systems applied to law: A framework for discussion. In T. Bench-Capon, editor, *Knowledge Based Systems and Legal Applications*, pages 329–342. Academic Press, 1991.

[5] T. Bench-Capon. Practical legal expert systems: the relation between a formalisation of law and expert knowledge. In Bennun and Narayanan, editors, *Law, Computer Science and and Artificial Intelligence*, pages 191–201. Ablex, 1991.

[6] T. Bench-Capon. The missing link revisited: The role of teleology in representing legal argument. *Artificial Intelligence and Law*, 10(1–3):79–94, 2002.

[7] T. Bench-Capon and E. Rissland. Back to the future: Dimensions revisited. In In B. Verheij, A.R. Lodder, R.P. Loui, and A. J. Muntjewerff, editors, *Legal Knowledge and Information Systems*, pages 41–52, Amsterdam, 2001. IOS Press.

[8] T. Bench-Capon and G. Sartor. Using values and theories to resolve disagreement in law. In R. Leenes J. Breuker and R. Winkels, editors, *Legal Knowledge and Information Systems. Jurix 2000: The Thirteenth Annual Conference*, pages 73–84, Amsterdam, 2000. IOS Press.

[9] T. Bench-Capon and G. Sartor. Theory based explanations of case law domains. In *ICAIL '01: Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 12–21. ACM Press, 2001.

[10] T. Bench-Capon and G. Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1–2):97–143, 2003.

[11] D.H. Berman and C.L. Hafner. Representing teleological structure in case based reasoning: The missing link. In *In Proceedings of the Fourth International Conference on AI and Law*, pages 50–59. ACM Press, 1993.

[12] L.K. Branting. Representing and reusing explanations of legal precedents. In *Proceedings of the 2nd International Conference on Artificial intelligence and Law*, pages 103–110. ACM Press, 1989.

[13] L.K. Branting. Reasoning with portions of precedents. In *Proceedings of the Third International Conference on AI and Law*, pages 145–54. ACM Press, 1991.

[14] S. Brüninghaus and K. Ashley. Combining case-based and model-based reasoning for predicting the outcome of legal cases. In K. D. Ashley and D. G. Bridge, editors, *ICCBR*, volume 2689 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2003.

[15] S. Brüninghaus and K. Ashley. Predicting outcomes of case-based legal arguments. In *Proceedings of the Ninth International Conference on AI and Law*, pages 233–42. ACM Press, 2003.

[16] S. Brüninghaus and K. Ashley. Generating legal arguments and predictions from case texts. In *Proceedings of the Tenth International Conference on AI and Law*, pages 65–74. ACM Press, 2005.

[17] A. Chorley and T. Bench-Capon. Developing legal knowledge based systems through theory construction. Technical Report ULCS-03-013, Department of Computer Science, The University of Liverpool, http://www.csc.liv.ac.uk/research/techreports/tr2003/ulcs-03-013.pdf, 2003.

[18] A. Chorley and T. Bench-Capon. Reasoning with legal cases as theory construction: Some experimental results. In D. Bourcier, editor, *Proceedings of the 16th*

Annual Conference on Legal Knowledge and Information Systems (Jurix 2003), pages 173–82, Amsterdam, 2003. IOS Press.

[19] A. Chorley and T. Bench-Capon. Agatha: Using heuristic search to automate the construction of case law theories. *Artificial Intelligence and Law*, 2006.

[20] A. Chorley and T. Bench-Capon. An empirical investigation of reasoning with legal cases through theory construction and application. *Artificial Intelligence and Law*, 2006.

[21] K. Greenwood, T. Bench-Capon, and P. McBurney. Towards a computational theory of persuasion in law. In *Proceedings of the Ninth International Conference on AI and Law*, page 25. ACM Press, 2003.

[22] A. Kowalski. Case-based reasoning and the deep structure approach to knowledge representation. In *Proceedings of the Third International Conference on AI and Law (ICAIL 1991)*, pages 21–30. ACM Press, 1991.

[23] E.H. Levi. *An Introduction to Legal Reasoning*. University of Chicago Press, Chicago, 1949.

[24] T.A. Lipinski and J.J. Britz. Rethinking the ownership of information in the 21st century: Ethical implications. In *Ethics and Information Technology*, volume 2, pages 49–71, 2000.

[25] D.N. MacCormick and R. S. Summers. *Interpreting Precedents: A Comparative Study*. Dartmouth Publishing, Aldershot, U.K., 1997.

[26] L.T. McCarty. An implementation of eisner v macomber. In *Proceedings of the Fifth International Conference on AI and Law*, pages 276–286. ACM Press, 1995.

[27] H. Prakken. *Logical Tools for Modelling Legal Argument. A Study of Defeasible Reasoing in Law*. Kluwer Law and Philosophy Library, Dordrecht, 1997.

[28] H. Prakken. An exercise in formalising teleological case based reasoning. In J. Breuker, R. Leenes, and R. Winkels, editors, *Legal Knowledge and Information Systems: Jurix 2000*, pages 49–57, Amsterdam, 2000. IOS Press.

[29] H. Prakken. An exercise in formalising teleological case-based reasoning. *Artificial intelligence and law*, 10(1–3):113–133, 2002.

[30] H. Prakken and G. Sartor. Modelling reasoning with precedents in a formal dialogue game. *Artificial Intelligence and Law*, 6(2-4):231–287, 1998.

[31] E. Rich. *Artificial Intelligence*. McGraw-Hill, Inc., 1984.

[32] E. Rissland and K. Ashley. A note on dimensions and factors. *Artificial Intelligence and Law*, 10(1–3):65–77, 2002.

[33] E. Rissland, D. Skalak, and M.T. Friedman. BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*, 4(1):1–71, 1996.

[34] E. Rissland, D. Skalak, and M.T. Friedman. Evaluating a legal argument program: The BankXX experiments. *Artificial Intelligence and Law*, 5(1-2):1–74, 1997.

[35] S. Russell and P. Norvig. *Artificial Intelligence A Modern Approach*. Prentice Hall, 1995.

[36] G. Sartor. Teleological arguments and theory-based dialectics. *Artificial Intelligence and Law*, 10(2–3):95–112, 2002.

[37] M. Sergot, F. Sadri, R. Kowalski, F. Kriwaczek, P. Hammond, and H. Cory. The british nationality act as a logic program. *Communications of the ACM*, 29(5):370–386, 1986.

[38] D. Skalak and E. Rissland. Arguments and cases: an inevitable intertwining. *Artificial Intelligence and Law*, 1:3–44, 1992.

[39] J. C. Smith and Cal Deedman. The application of expert systems technology to case-based law. In *Proceedings of the First International Conference on AI and Law (ICAIL 1987)*, pages 84–93. ACM Press, 1987.

[40] P.H. Winston. *Artificial Intelligence*. Addison Wesley, Reading, Mass, third edition, 1992.