

Proceedings of the Workshop on
Modelling Legal Cases and Legal Rules

in conjunction with
JURIX 2010

December 15, 2010
University of Liverpool
Liverpool, United Kingdom

Adam Zachary Wyner (ed.)

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF LIVERPOOL

JURIX Conference Committee Chairs

Radboud Winkels Program Committee Chair

Katie Atkinson Organisation Committee Chair

Workshop Programme Committee

Kevin Ashley University of Pittsburgh, USA

Katie Atkinson University of Liverpool, UK

Floris Bex University of Dundee, UK

Trevor Bench-Capon University of Liverpool, UK

Tom Gordon Fraunhofer FOKUS, Germany

Robert Richards Seattle, Washington, USA

Giovanni Sartor European University Institute, Italy

Burkhard Schafer Edinburgh Law School, Scotland

Douglas Walton University of Windsor, Canada

Conference Website

<http://pcwww.liv.ac.uk/~confcsc/Jurix2010.html>

Copyright

The authors of the papers hold the copyright of their paper.

Preface

In conjunction with JURIX 2010, the 23rd International Conference on Legal Knowledge and Information Systems Conference, in Liverpool UK, I organised a Workshop on Modelling Legal Cases and Legal Rules, December 15, 2010. Broadly, the aim of the workshop was to provide a forum in which researchers could present their most recent research on modelling legal cases and legal rules. It follows on from successful workshops at Jurix 2007 and ICAIL 2009, which were organised by Katie Atkinson. This volume contains the papers which were accepted to workshop. In addition to the presentation of the papers, time was set aside at the workshop for informal topics to be briefly presented.

Legal cases and legal rules in common law contexts have been modelled in a variety of ways over the course of research in AI and Law to support different styles of reasoning for a variety of problem-solving contexts, such as decision-making, information retrieval, teaching, etc. Particular legal topic areas and cases have received wide coverage in the AI and Law literature including wild animals (e.g. *Pierson v. Post*), intellectual property (e.g. *Mason v. Jack Daniel Distillery*), and evidence (e.g. the *Rijkbloem* case). As well, some legal rules have been widely discussed, such as legal argument schemes (e.g. Expert Testimony) or rules of evidence. However, other areas have been less well covered. For example, there appears to be less research on modelling legal cases in civil law contexts; investigation of taxonomies and ontologies of legal rules would support abstraction and formalisation; additional legal rules could be brought under the scope of investigation, such as those bearing on criminal assault or causes of action. The workshop provided, then, an opportunity to focus scholarly attention on these and related topics, to demonstrate the benefits of particular approaches to case and rule analysis, to discuss useful overlapping features, as well as to bring forth topics for future investigation.

Papers were solicited that model a particular legal case or a small set of legal rules. Authors were free to choose the case or set of legal rules and analyse them according to the authors' preferred model of representation; the theoretical discussion was grounded in or exemplified by the case or rules at hand. Papers outlined the particular distinctive features of the approach and explained why these features were useful in modelling the chosen case or rules.

The papers present a range of approaches and topics. van Driel and Prakken use the argumentation visualisation software *Rationale* to analyse the structure of a report by a Dutch expert witness in a criminal case. It is claimed that such a visualisation may help a judge to understand the report and to identify points for further inquiry. An informal evaluation of the visualisation was conducted. Gordon's contribution analyses a range of open source licenses and presents a prototype system to identify incompatibilities between the licences. An ontology of copyright and software concepts is modelled in the Web Ontology Language (OWL); components of a software system are instances in the ontology. The Legal Knowledge Interchange Format (LKIF) is used to model alternative theories (e.g. licenses) of the copyright and software concepts. The Carneades argumentation system, using the OWL and LKIF models, constructs arguments for and against the claim that a particular license is compatible with components found in a particular software system. Lloyd-Kelly, Wyner, and Atkinson outline a variety of ways in which emotional arguments appear in legal case argument and decisions. They propose that rather than filter out "bad" emotional argument or to make emotional arguments an adjunct to rational argumentation, emotional argument can be formalised and integrated into an overall representation of arguments. It is claimed that this would be particularly useful in the analysis of legal case decisions. Ronainen describes a knowledge-based system prototype which implements a model of an open-textured

legal rules about European trade marks using fuzzy logic. A machine learning approach is adopted, where a training set of cases is constructed and used to train the parameters for the case facts. The parameters are then used in a fuzzy logic rule, which determines whether or not a trade mark claim should be refused. It reports a high level of success in applying the rule to the test set of cases. Schafer's paper discusses and formalises a debate by the European Court of Justice on geographical indications concerning the term "Feta Cheese". It shows that the justices informally make use of the semantics of natural kind terms as discussed in analytic philosophy. Wyner and Bench-Capon describe a Prolog program which generates visualisations to support legal case based reasoning. A current, undecided case is compared to a precedent; the comparative aspects are then input to argumentation schemes which represent reasoning about a case with respect to a precedent (e.g. distinguishing a case, downplaying a distinction, etc); the graph of the schemes can be used to reason about the utility of the precedent with respect to arguing the current case.

I wish to thank the JURIX conference committee chairs, the workshop program committee members, and all the authors for their contribution to the success of this workshop.

Liverpool, United Kingdom
December 2010

A.Z.W.

Contributed Papers

Visualising the Argumentation Structure of an Expert Witness Report with <i>Rationale</i> <i>Steven van Driel and Henry Prakken</i>	1
Analyzing Open Source License Compatibility Issues with Carneades <i>Thomas F. Gordon</i>	9
Emotional Argumentation Schemes in Legal Cases <i>Martyn Lloyd-Kelly, Adam Wyner, and Katie Atkinson</i>	19
MOSONG, a Fuzzy Logic Model of Trade Mark Similarity <i>Anna Ronkainen</i>	23
Say “Cheese”: Natural Kinds, Deontic Logic and European Court of Justice Decision C-21089 <i>Burkhard Schafer</i>	33
Visualising Legal Case-based Reasoning Argumentation Schemes <i>Adam Wyner and Trevor Bench-Capon</i>	43

Visualising the argumentation structure of an expert witness report with Rationale (extended abstract)

Steven VAN DRIEL ^a, Henry PRAKKEN ^b

^a *Department of Information and Computing Sciences, Utrecht University, The Netherlands*

^b *Department of Information and Computing Sciences, Utrecht University and Faculty of Law, University of Groningen, The Netherlands*

Abstract. This paper reports on a case study in which the use of the Rationale software was investigated to analyse the argumentation structure of a Dutch expert witness report in a criminal case. The underlying motivation of the case study was to explore the usefulness of argumentation visualisation software for increasing a judge's understanding of expert reports and for assisting him or her in asking the proper critical questions to the expert. By way of an initial exploration of this usefulness, an expert report was analysed with the Rationale software. The visualisation was informally discussed with a legal expert, who was generally positive but also expressed some concerns and expected that the main usefulness of the tool is in training and education of judges.

1. Introduction

It has often been suggested that argument visualisation software (AVS) can be useful in managing the complexity of argumentation and proof in legal cases (Schum and Tillers; 1991; Verheij; 2005; Van Gelder; 2007; Walker; 2007; Sombekke et al.; 2007; Van den Braak; 2010). This extended abstract reports on a case study in which the use of AVS was investigated to analyse the argumentation structure of a Dutch expert witness report in a criminal case. The case study was carried out in collaboration with the Dutch Council of the Judiciary (Raad voor de Rechtspraak, henceforth RvdR)¹.

The practical motivation for the project was as follows. The increasing complexity of legal cases has led to an increase in the number and complexity of expert witness testimonies. Since judges are laypersons in the fields of expertise of the expert witnesses, they often find it hard to understand the expert reports and to ask the proper questions to the expert (Kwakman; 2006). The RvdR is therefore very interested in education programmes and support tools that can help judges in dealing with expert evidence.

One tool which can possibly provide support to judges is an AVS. If a judge analyses the argumentation structure of an expert report with an AVS, this could increase the judge's understanding of the report and assist him or her in asking the proper questions

¹We thank Peter van Dam of the RvdR for his support and collaboration.

to the expert. Alternatively, these benefits could result if the judge is provided with such an analysis made by someone else. The purpose of the present case study was to give an initial exploration of this potential usefulness of AVS. To this end, an expert report containing a psychological examination of a suspect of a robbery with attempted murder was analysed by us with the Rationale software (Van Gelder; 2007). We were in particular interested to see whether the report contains a substantial amount of argumentation and, if so, to what extent the argumentation can be visualised with a tool like Rationale. The resulting visualisation was informally discussed with a staff member of the RvDR who is involved in the RvDR's programmes concerning expert evidence. The case study was carried out as part of a two-and-a-half month undergraduate project of the first author of this extended abstract, supervised by the second author.

In this extended abstract we first briefly introduce the used Rationale software and motivate our choice for Rationale in light of alternatives such as Avers (Van den Braak; 2010) and Araucaria (Reed and Rowe; 2004). We then briefly summarise the expert report and outline the analysis of its argumentation structure with Rationale. We conclude with some comments on the experiences gained during the study and the potential of AVS like Rationale for practical use in legal settings.

2. Argument visualisation software

Different AVS implicitly or explicitly commit to different models of reasoning and argumentation. Most AVS support the visualisation of 'standard' argumentation models, in which conclusions are supported by one or more grounds, which can be horizontally linked (all grounds needed to support the conclusion) or regarded as alternative grounds for the conclusion (one ground suffices to support the conclusion). Grounds can be 'vertically' supported by further (combinations of) grounds, resulting in a tree structure. Several AVS of this kind also support the visualisation of objections or counterarguments. Araucaria allows statements in two trees to be horizontally connected, meaning that the two connected statements are incompatible. Rationale, by contrast, allows vertical 'objection' links from one statement to another, meaning that the former statement supports the negation of the latter statement (this interpretation is based on personal communication with Tim van Gelder and on ter Berg et al. (2009, pp. 25–6)). In addition, Rationale allows that a statement attacks the connection between grounds and a conclusion. Thus it can be said in terms of Prakken (2010a) that Araucaria only supports the visualisation of rebutting and undermining attacks while Rationale also supports the visualisation of undercutting attacks.

The possibility to represent undercutters was not the reason why Rationale was adopted in this case study. The reason was instead that Rationale, being a commercial product, has superior facilities for zooming in and out and thus for maintaining overview over large graphs. This proved essential for visualising the expert report.

As explained further below, the expert made use of abductive reasoning, which raises the question why Avers (Van den Braak; 2010) was not used. Avers is an AVS for crime investigation in which scenarios about what may have happened can be constructed and then linked with 'standard' argumentation to the available evidence. However, as also explained further below, the expert's use of abduction could be modelled in with an argumentation scheme, which obviated the need for Avers.

3. The expert report

The expert report concerns a psychological examination of a suspect of a robbery with attempted murder. To support the judge in deciding whether the suspect was mentally accountable for the crime, the expert had to provide insight the mental state of the suspect during the crime and its effect on the suspect's behaviour. In this abstract we focus on the subquestion whether the suspect is suffering from some mental disease. In his report the expert mixes a description of several psychological tests and examinations with two alternative diagnoses. Initial observations and tests suggest that the suspect is a timid, sensitive boy whose emotions block quickly and who is strongly hindered by feelings of inadequacy and inferiority. This diagnosis predicts, among other things, that the suspect is introvert and neurotic and suffers from a form of fear of failure that negatively impacts on performance. However, these predictions are contradicted by later findings. The expert then adopts a second diagnosis, namely, that the suspect suffers from a severe inability to experience and express his inner feelings and emotions (in Figure 1 summarised as 'Predominant feelings of inadequacy'), which the expert then classifies as an atypical kind of autism. The expert supports his second diagnosis in two ways. He first gives an abductive argument, stating that the findings on the suspect's behaviour can be explained by the severe inability to experience and express his inner feelings and emotions. He then gives additional direct evidence for the existence of this severe inability.

An overview of the expert's reasoning is provided in Figure 1 (made by us by summarising several subtrees in single nodes).

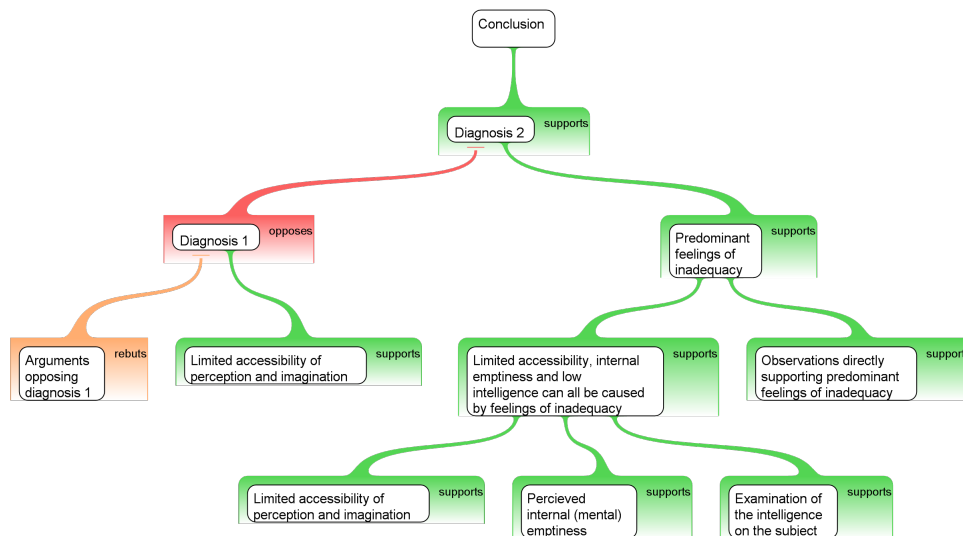


Figure 1. Overview

In this brief summary of the expert report in fact already a choice has been made for a particular theoretical view on the expert's argumentation, namely, as essentially being abductive diagnosis. In fact, this account was only adopted after a first analysis in a more rule-based style (effectively applying modus ponens to observations and implicit gener-

alisations) proved unsatisfactory. After adopting the abductive theoretical view, the further analysis was guided by the following argumentation scheme for abductive diagnosis (similar to Walton (1996)'s scheme from evidence to hypothesis).

$$\frac{\begin{array}{l} \text{Person } P \text{ exhibits behaviours } B_1, \dots, B_n \\ \text{Diagnosis } D \text{ explains behaviours } B_1, \dots, B_n \end{array}}{\text{Therefore, Person } P \text{ suffers from disease } D}$$

The critical questions are:

- CQ1: Does diagnosis D predict other behaviours that are contradicted by observations?
- CQ2: Can behaviours B_1, \dots, B_n be explained by alternative diagnoses?

Positive answers to these questions generate counterarguments. The support for the first premise of the two abductive arguments was visualised in modus ponens style with generalisations left implicit, ultimately based on implicit generalisations that tests and examinations of a certain kind are reliable indicators of their results.

4. Some findings

4.1. Visualising abductive reasoning

AI models of abductive diagnosis are not stated in terms of argumentation (Prakken; 2010b). Nevertheless, in the present case a 'standard' argumentation format proved sufficient, since the expert's diagnostic reasoning could be modelled with the above-listed argument scheme for abductive diagnosis (although Rationale does not provide support for constructing instances of schemes, as in Avers, nor for indicating which scheme has been applied, as in Araucaria). The positive answer to CQ1 was visualised as an objection to the first diagnosis (See Figure 2, which displays the first four levels of the argument in support of the diagnosis plus the conclusion of the objection).

Alternatively it could have been modelled as an undercutter. In that case, it would in Figure 2 have been linked to the support box containing the two grounds that directly support diagnosis 1. The positive answer to CQ2 was visualised by letting the alternative diagnosis be the main diagnostic conclusion and letting the first diagnosis be an objection to that diagnosis. What cannot be modelled as such in Rationale is the reasons why the second diagnosis is preferred, namely, that the second diagnosis explains both the initial findings (also explained by the first diagnosis) and the further findings (contradicted by the predictions of the first diagnosis). In fact, these reasons are also largely left implicit by the expert. All he does is saying that the suspect's behaviour is not caused by the facts assumed in the first diagnosis but by the severe inability to experience and express his inner feelings and emotions (the second diagnosis). This expresses a choice for the second diagnosis without giving explicit reasons for this choice (See the bottom-left node in Figure 3).

Note that besides the abductive argument the expert gives an additional evidential argument for the existence of the defect. This was visualised as two separate arguments for the same conclusion (see Figure 3). Arguably these arguments accrue Prakken (2005) in that together they provide stronger support for the second diagnosis than alone but accrual can in Rationale not be visualised.

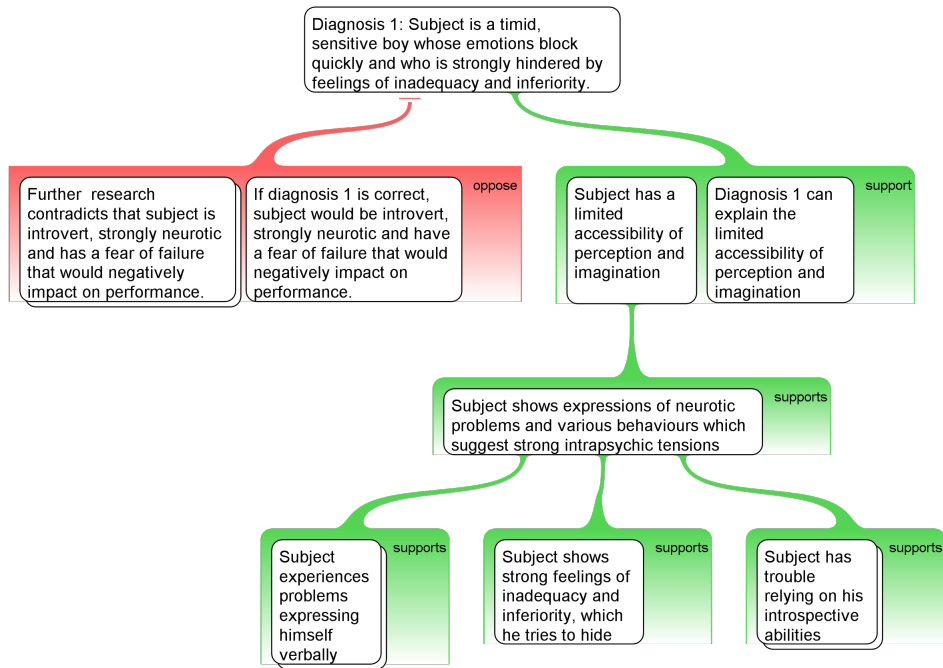


Figure 2. Diagnosis 1 and an objection

4.2. Objectivity of the visualisation

A visual reconstruction of a realistic piece of argumentation inevitably involves interpretation. In this case study a particular problem was that the expert seemed to use many different ways to formulate essentially the same findings or diagnoses. We have tried to unify them but we may have been inaccurate. Apart from this we did not find many argumentative ambiguities in the expert's arguments but other analysts might disagree.

4.3. The legal expert's feedback

After the visual reconstruction of the expert report was completed, it was informally discussed with a legal expert of the RvdR involved in the RvdR's programmes concerning expert evidence. He was generally positive but also expressed some concerns. We briefly summarise his feedback.

To start with, the expert feared that judges might be intimidated by the size and detail of the graph (Figure 4 displays the complete graph, to give an indication of its size). Combined with the fact that most nodes are green, that is, there are not many counter-arguments, this might make that judges will not check the graph in detail. According to the expert, tools for abstraction and summary are needed, such as a feature to summarise uncontroversial parts of the graph in a single abstraction node. For example, ideally a summary as provided by Figure 1 could be automatically linked to the detailed analysis, so that clicking on a node in Figure 1 would display the corresponding subtree in detail.

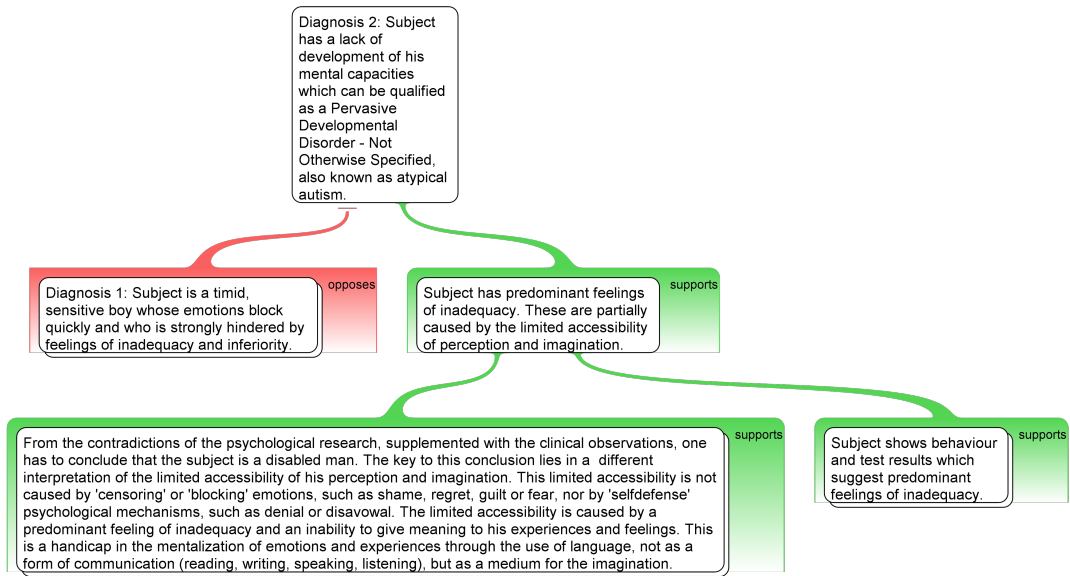


Figure 3. Diagnosis 2

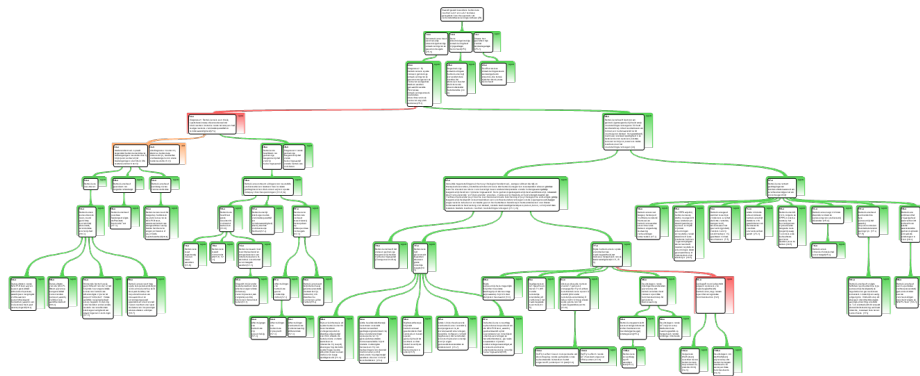


Figure 4. The complete graph

After the idea of argument schemes was explained to the legal expert, he found support for their use desirable, particularly to help in identifying implicit premises and in developing a critical attitude towards the expert witnesses' reasoning.

Finally, the legal expert expected that a tool like this is likely to be more useful in training of judges and education of law students than in legal practice. Once judges have become aware of the potential sources of doubt in argumentation, they can develop a critical attitude towards expert reports without the need to visualise them with an AVS.

5. Conclusions

It goes without saying that a project of such limited scope cannot result in firm conclusions. With this in mind it can be concluded that this project provides some initial support for the hypothesis that AVS may be useful in increasing a judge's understanding of expert reports and in assisting him or her in asking the proper questions to the expert, especially when used as a training or education tool. In particular, we found that the expert report we analysed contains a substantial amount of argumentation, which could be visualised in a natural way with Rationale. However, the usefulness of AVS should be further investigated in studies of larger size, preferably with involvement of judges and expert witnesses as test subjects and/or evaluators. In particular, in the present case study the visualisation was made by us and was not shown to a judge. Letting judges make the analysis or confronting them with an analysis made by someone else may lead to better insights into the potential of AVS to support judges.

Although the Rationale software proved generally suitable for visualising the expert's reasoning, the study has also yielded some suggestions for extending the software. First, although Rationale provides excellent support for maintaining overview of large graphs, additional tools for abstraction and summary may further increase the usefulness of the system. Another useful addition would be support for argument schemes. For example, a feature like in Avers (Van den Braak; 2010) could be added, where users can indicate that they want to add an instance of a scheme to the graph, after which the system automatically adds templates for the premises, conclusion and critical questions to the graph, to be completed by the user. It should be noted that currently Rationale allows links from "Basis Boxes" (for example, "Common belief", "Law" or "Expert opinion") to argument premises, to indicate the source of the premise. However, these links are not meant to be argumentative (Tim van Gelder, personal communication and ter Berg et al. (2009, p. 27)). In particular, they cannot be undercut. It seems a natural idea to replace these boxes by source-based argumentation schemes, so that the support that sources provide for argument premises can be critically examined.

References

- Kwakman, N. (2006). De verhouding tussen wetenschap en recht: een cultuuromslag waard?, *Trema: Tijdschrift voor de Rechterlijke Macht* **9**. In Dutch.
- Prakken, H. (2005). A study of accrual of arguments, with applications to evidential reasoning, *Proceedings of the Tenth International Conference on Artificial Intelligence and Law*, ACM Press, New York, pp. 85–94.
- Prakken, H. (2010a). An abstract framework for argumentation with structured arguments, *Argument and Computation* **1**: 93–124.
- Prakken, H. (2010b). On the nature of argument schemes, in C. Reed and C. Tindale (eds), *Dialectics, Dialogue and Argumentation. An Examination of Douglas Walton's Theories of Reasoning and Argument*, College Publications, London, pp. 167–185.
- Reed, C. and Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation, *International Journal of AI Tools* **13**: 961–980.
- Schum, D. and Tillers, P. (1991). Marshaling evidence for adversary litigation, *Cardozo Law Review* **13**: 657–704.

- Sombekke, J., van Engers, T. and Prakken, H. (2007). Argumentation structures in legal dossiers, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Law*, ACM Press, New York, pp. 277–281.
- ter Berg, T., van Gelder, T., Patterson, F. and Teppema, S. (2009). *Critical Thinking: Reasoning and Communicating with Rationale*, Pearson Education Benelux, Amsterdam.
- Van den Braak, S. (2010). *Sensemaking Software for Crime Analysis*, Doctoral dissertation Department of Information and Computing Sciences, Utrecht University.
- Van Gelder, T. (2007). The rationale for RationaleTM, *Law, Probability and Risk* **6**: 23–42.
- Verheij, B. (2005). *Virtual Arguments. On the Design of Argument Assistants for Lawyers and Other Arguers*, T.M.C. Asser Press, Den Haag.
- Walker, V. (2007). Visualizing the dynamics around the rule–evidence interface in legal reasoning, *Law, Probability and Risk* **6**: 5–22.
- Walton, D. (1996). *Argumentation Schemes for Presumptive Reasoning*, Lawrence Erlbaum Associates, Mahwah, NJ.

Analyzing Open Source License Compatibility Issues with Carneades

Thomas F. Gordon
Fraunhofer FOKUS, Berlin

Abstract. Building on our prior theoretical and practical work on the Legal Knowledge Interchange Format (LKIF) and the Carneades argumentation system, we have developed a proof-of-concept, prototype system for helping developers to construct, explore and compare legal theories when analysing open source licensing issues in particular cases.

Keywords. knowledge representation, argumentation, theory construction, copyright law

1. Introduction

This paper presents a prototype system, developed in the European Qualipso project (IST-FP6-034763), for helping software developers to analyse Open Source license compatibility issues. The prototype is an application of the Legal Knowledge Interchange Format (LKIF) [2], which was developed in the European ESTRELLA project (IST-2004-027655). An ontology of relevant copyright and software concepts is modelled using the Web Ontology Language (OWL). Alternative theories of the copyright concept of a derivative work are modelled using LKIF rules. The components of a software system, as well as their properties and relations, are modelled as OWL instances. The Carneades [3,4] argumentation system uses the LKIF and OWL models to automatically construct arguments pro and con the claim that some Open Source license, such as the GPL, is compatible with the licenses of the components used by software system. The Carneades editor, an interactive argument mapping application, is then used to visualize, explore, and evaluate the effects of the alternative legal theories on the compatibility of particular licences.

We start by developing a simple ontology of concepts and relations for describing software licenses and use and derivation relationships between works of software. We include in this ontology formal models of some popular open source licenses, such as the GPL and BSD licenses. Next we show how to use the ontology to model relationships between works of software in a software project. We then show how to model some rules of copyright law using the Legal Knowledge Interchange Format, focusing on the issue of what kinds of uses of a work of software produce derivative works. These models are then used to construct, evaluate and visualize arguments about whether or not the project may publish its software using a particular open source license, i.e. whether a preferred license is compatible with the licenses of the software

used by the project. When constructing arguments we illustrate how abduction can be used to focus the search for issues or goals to work on which are helpful for proving, or disproving, depending on one's standpoint and interests, that the license is compatible.

2. Ontology for Open Source Licenses

Ontologies in computer science are an advanced kind of data model. Carneades uses the Web Ontology Language (OWL), a World Wide Web standard [6], for representing and interchanging ontologies. OWL is well supported by various tools, including the open source Protege editor.¹

The top-level classes, i.e. subclasses of the root Thing class, are:

- CopyrightLicense. Individual licenses, with which a particular legal entity, the licensor, grants rights to another legal entity, the licensee.
- CopyrightLicenseTemplate. Open source license templates, such as the GPL or BSD. A particular copyright license can be an instance of such a template.
- LegalEntity. Legal persons, such as humans, corporations and associations.
- LicenseTerm. The rights granted by a license and the conditions of the license, limiting the rights granted.
- Work. Various kinds of intellectual products protected by copyright, including software.

Two subclasses of CopyrightLicense have been defined:

- OpenSourceLicense. This class in turn has AcademicLicense and ReciprocalLicense subclasses.
- ProprietaryLicense

The CopyrightLicense class has the following properties:

- grantsRight: CopyrightLicense \times Right
- hasCondition: CopyrightLicense \times Condition
- hasLicensee: CopyrightLicense \times LegalEntity
- hasLicensor: CopyrightLicense \times LegalEntity
- instanceOfTemplate:
CopyrightLicense \times CopyrightLicenseTemplate

The CopyrightLicenseTemplate class has a property for license compatibility:

- isCompatibleWith:
CopyrightLicenseTemplate \times CopyrightLicenseTemplate

Notice we have defined this property for license templates, not for individual licenses. Whether or not two particular licenses are compatible with each other can be derived from the compatibility of their templates.

The isCompatibleWith property is reflexive (every license template is compatible with itself), but not symmetric. A template license may be compatible with another template license without the reverse necessarily being the case. For example, software derived from LGPL software may be licensed using the GPL, but not vice versa.

Open source copyright license templates, such as the GPL, are modelled in two parts in this ontology: 1) By *instances* of the CopyrightLicenseTemplate class, e.g. the ApacheLicense2.0Template; and 2) By *subclasses* of the OpenSourceLicense class, e.g. the ApacheLicense2.0 class.

¹ <http://protege.stanford.edu/>

These two parts of the model of each template license are linked together with an axiom stating that a license which is an instance of a given template is also an instance of the appropriate class, and vice versa. For example, the BSD class is linked to the BSD_Template instance with the following equivalence axiom:

$\text{BSD} \equiv \text{instanceOfTemplate value BSD_Template}$

The reason for modelling each template license as both an instance and a class is that, in order to analyse license compatibility issues, we need some way to reason about whether a license template is compatible with the licenses of the software entities used by a project. For this the license templates need to be individuals, not just classes, because the underlying logic of the Web Ontology Language is description logic, which is semantically a subset of first-order logic. A second-order logic would be needed for reasoning about classes directly.

The ontology also provides classes modelling the rights granted and conditions of licenses and license templates. Here are some example rights and conditions:

1. LicenseTerm
 - a) Right
 - MayCopy
 - MayDistributeDerivativeWorksInObjectForm
 - MayDistributeDerivativeWorksInSourceForm
 - MayProduceDerivativeWorks
 - b) Condition
 - LimitedLiability
 - MustDistributeCopyOfLicense
 - MustLicenseDerivativeWorksUnderCompatibleLicense
 - MustOfferSourceCode
 - ProvidedWithoutWarranties

Superclass axioms are used to express that instances of a particular class of license grant certain rights and are subject to certain conditions. For example, the ApacheLicense2.0 class is defined to be a subclass of the following Right and Condition classes, among others:

- grantsRight some MayCopy
- grantsRight some MayDistributeDerivativeWorksInObjectForm
- grantsRight some MayDistributeDerivativeWorksInSourceForm
- grantsRight some MayProduceDerivativeWorks
- hasCondition some LimitedLiability
- hasCondition some MustDistributeCopyOfLicense
- hasCondition some MustOfferSourceCode
- hasCondition some ProvidedWithoutWarranties

Any license which is an instance of the Apache License 2.0 template entails these rights and conditions. More formally, if an individual is a member of the class `instanceOfTemplate value ApacheLicense2.0Template` then it is, because of the equivalence axiom

$\text{ApacheLicense2.0} \equiv \text{instanceOfTemplate value ApacheLicense2.0_Template}$

also a subclass of the Right and Condition classes of which the ApacheLicense2.0 is a subclass.

A license with all of the rights and conditions of the Apache License 2.0, however, is not necessarily an instance of the ApacheLicense2.0 class. For this to be the case it must also be an instance of the Apache License 2.0 template. That is, it must express these license terms using exactly the same language as the Apache License 2.0 template.

This way of modelling license terms in a uniform way, across templates makes it possible to use an OWL reasoner to automatically classify license templates by their terms and to find license templates with preferred or selected terms.

3. Ontology for Software Systems

Next we present the classes and properties of the ontology designed for modelling relationships between software used in a project. The main class, `Work`, models all works protectable by copyright. The `SoftwareEntity` subclass of `Work` is intended to cover all kinds of software artefacts, including not only source and object code, but also more abstract entities such as APIs and specifications. Currently the ontology includes the following subclasses of `SoftwareEntity`, in alphabetical order:

- `ApplicationServer`
- `ObjectCode`
- `OperatingSystem`
- `Program`
 - `RichInternetApplication`
- `SoftwareLibrary`
- `SoftwareService`
- `SourceCode`
- `Specification`
 - `API`
 - `AbstractMachine`
 - `ProgrammingLanguage`

These classes are not intended to be complete, at least not in this prototype, but have been included as needed to model the software entities used in the example project.

Keep in mind that classes in OWL need not be disjunct. Thus, without further axioms in the ontology, a particular software entity can be, for example, be an instance of both the `Program` and `ObjectCode` classes.

The main property of software entities of interest for license compatibility issues is the `isDerivedFrom` property, expressing that one entity has been derived from another. This legal issue depends on the jurisdiction and the interpretation of the governing law by the courts. The ontology includes properties for representing various ways that software can use other software. These properties are not from the domain of copyright law, but rather from the domain of software engineering.

- `uses`: `Work` × `Work`
 - `compiledBy`: `SoftwareEntity` × `SoftwareEntity`
 - `implementedIn`: `SoftwareEntity` × `ProgrammingLanguage`
 - `implements`: `SoftwareEntity` × `Specification`
 - `linksTo`: `SoftwareEntity` × `SoftwareLibrary`
 - `linksDynamicallyTo`:
`SoftwareEntity` × `SoftwareEntity`
 - `linksStaticallyTo`:
`SoftwareEntity` × `SoftwareEntity`
 - `modificationOf`: `Work` × `Work`

- runsOnOperatingSystem:
SoftwareEntity × OperatingSystem
- servedBy: Work × SoftwareEntity
- usesService: SoftwareEntity × SoftwareService
- usesSpecification: SoftwareEntity × Specification

In legal terms, they provide the means to represent the *material facts* of a case. The legal question is whether a particular use of software, such as linking, is sufficient to create a derivative work. In legal jargon, the question is whether a material fact, linking, can be *subsumed* under a legal concept. Or, more formally, using the properties of the ontology, whether linksTo is subsumed by, i.e. a subproperty of, isDerivedFrom. None of these use relations has been defined in the ontology as a subproperty of isDerivedFrom, because these legal issues have not been resolved and we want to leave room to argue about these issues.

In addition to these use relations, the ontology includes properties for representing the license templates which are compatible with the licenses of any works from which it is derived, and for recording the licenses which have been issued for the work:

- mayUseLicenseTemplate: Work × CopyrightLicenseTemplate
- hasLicense: Work × CopyrightLicense

4. Example Model of a Software Project

The software ontology was used to model an example system, roughly based on the Clojure port of Carneades currently being developed. The actual system may be different, since this is work in progress and subject to change. For our purposes here, it is not necessary to explain all of the entities and relationships in the model. Let us focus on the Carneades inference engine. It is represented in the model as:

```

CarneadesEngine
  type: SoftwareLibrary
  implementedIn: Clojure
  compiledBy: ClojureCompiler
  linksStaticallyTo: ClojureLib
  linksStaticallyTo: ClojureContrib
  linksStaticallyTo: cljSandbox
  linksStaticallyTo: JgraphX
  linksStaticallyTo: OWL_API
  linksStaticallyTo: Pellet
  usesSpecification: JVM
  usesSpecification: OSGI

```

Since the example is designed to illustrate the process of using the prototype to help developers select compatible open source licenses, the CarneadesEngine does not yet have a license in the model.

One of the libraries used by the Carneades engine is the Clojure library (ClojureLib) which is licensed using the EPL. In the model, the ClojureLib individual does not have either the EPL class or the EPL_Template instance as its license directly, but rather has its own license, which we have named ClojureLicense. This license is a member of the EPL class, which, as you may recall, is equivalent to the class:

```
instanceOfTemplate value EPL_Template
```

This way of modeling licenses allows each instance of a template license to have its own licensor and licensee, which is necessary since the licensor of software licensed using a template license is the copyright owner of the software and not the owner of the template license, such as the Eclipse Foundation, the owner and maintainer of the EPL template license.

Another library linked to by the Carneades engine uses GNU AGPL license template. Since the EPL and AGPL are incompatible reciprocal licenses, a central question will be whether the Carneades engine can be linked to both of these libraries. This will depend on whether the linking of software to a library causes the software to be a derivative work of the library. We will return to this question after showing, in the next section, how alternative interpretations of copyright law can be modelled using LKIF rules.

5. Rules

The Web Ontology Language, OWL, is based on description logic, which is semantically a decidable subset of first-order logic. This means that the inferences of an OWL reasoner are *strict*: if the axioms of an OWL ontology are true in some domain, then all of the inferences made by a (correctly implemented) OWL reasoner are necessarily also true, without exception. While OWL is very powerful and useful, first-order logic is not sufficient for modelling and reasoning about legal norms, such as the rules of copyright law, in a maintainable and verifiable way, isomorphic with the structure of legislation and regulations. Legislation is typically organized as general rules subject to exceptions. Arguments made by applying legal rules are *defeasible*. Their conclusions can be defeated with better counterarguments. Various legal rules may conflict with each other. These conflicts are resolved using legal principals about priority relationships between rules, such as the principal of *lex superior*, which gives rules from a higher authority, such as federal law, priority over rules from a lower authority, such as state law.

Thus we model legal rules using a defeasible rule language which has been developed especially for this purpose, as part of the Legal Knowledge Interchange Format (LKIF), and use OWL for more limited purposes: 1) to declare the language of unary and binary predicate symbols (classes and properties, in OWL terminology) of the application domain; and 2) to define relationships between these predicates, using OWL axioms, which are judged to be universally true and beyond dispute in the domain.

Here we illustrate the LKIF rule language by modeling some interpretations of the rules of copyright law. Since opinions differ about how to interpret copyright law in the context of open source licensing issues, for example about whether or not linking to a software library creates a derivative work, an important feature of our approach is the ability to include alternative interpretations in a single model, and to construct and compare competing arguments from these alternative formulations of the rules when analysing licensing issues of a software project.

We begin with the general rule that the copyright owner of software may license the software using any license template he chooses.

```
<rule id="DefaultLicenseRule">
  <head>
```

```

    <s pred="&oss;mayUseLicenseTemplate">
      <v>SE</v> may be licensed using the <v>T</v> template
    </s>
  </head>
</rule>

```

Since LKIF is an XML schema, rules are represented in XML. This particular rule has a head (conclusion) but no body (conditions). Even though the rule has no conditions, inferences made using this rule are not necessarily or universally true, but remain defeasible. We will make use of this feature to express exceptions to this general rule below.

The rule has been assigned an identifier, `DefaultLicenseRule`, which may be used to formulate statements about the rule. That is, rules are objects of the domain model, and may be reasoned about just like other objects. This feature is called “reification” in the field of knowledge representation.

The predicate symbol of the statement (proposition) in the head of the rule is specified using the `pred` attribute. Its value can be the name of a class or property in a OWL ontology, as in this example. The `&oss;` entity reference refers the ontology, using its URI. The entity is defined at the top of the LKIF file, as follows:

```

<!DOCTYPE rb [
  <!ENTITY oss "http://carneades.berlios.de/oss-licenses#">
]>

```

Declaring predicate symbols in ontologies makes it possible to divide up the model of a complex domain theory into modules, with a separate LKIF file for each module. OWL provides a way for ontologies to import the classes and properties of other OWL files, recursively. Similarly, LKIF provides a way to import both LKIF and OWL files. OWL makes it easy to manage predicate symbols across the boundaries of modules and to make sure that symbols in different modules refer to the same class or property when this is desired.

The XML syntax for rules in LKIF is rather verbose and not especially readable. Fortunately, it is easy to write programs for converting XML into more readable formats. Moreover, XML structure editors exist, such as Oxygen², which use style sheets to enable authors to edit XML documents directly in a more readable form. Using this feature, the above rule can be displayed in the editor as follows:

```

rule DefaultLicenseRule
  SE may be licensed using the T template

```

In this format, variables are underlined. The predicate symbol from the OWL ontology is not shown, but can be viewed and edited in a separate properties panel when the cursor is placed within the text of the statement. We will use this more readable format for displaying LKIF rules in the remainder of this article.

Next let us formulate an exception to the general rule that any license template may be used for reciprocal licenses:

```

rule ReciprocityRule
  not: SE1 may be licensed using the T1 template

```

² <http://www.oxygenxml.com/>

given

```

SE1 uses SE2
SE2 is licensed under L
L is reciprocal
SE1 is derived from SE2
L is an instance of the T2 template license
unless: T1 is compatible with T2

```

This reciprocity rule states that a software entity, SE1, may not be licensed using a template license, T1, if the software is derived from another software entity, SE2, licensed using a reciprocal license, L, unless L is an instance of a license template, T2, which is compatible with T1. Notice that the conclusion of the rule is negated and that the last condition of the rule expresses an exception (“unless ...”).

The first two conditions of the rule serve a heuristic purpose. They provides control information enabling fully instantiated arguments to be constructed when this rule is applied, without having to first search for arguments for the conditions of the rule.

Let us end this brief overview with rules modelling two conflicting views about whether or not linking creates a derivative work.³ According to the lawyers of the Free Software Foundation, linking does create a derivate work. Lawrence Rosen, a legal expert on open source licensing issues [7] takes the opposing point of view and argues that linking per se is not sufficient to create derivate works.

rule FSFTheoryOfLinking

```

SE1 is derived from SE2

```

given

```

SE2 is a software library
SE1 is linked to SE2
The FSF theory of linking is valid

```

rule RosenTheoryOfLinking

```

not: SE1 is derived from SE2

```

given

```

SE2 is a software library
SE1 is linked to SE2
The Rosen theory of linking is valid

```

The last condition of each of these rules requires that the interpretation of copyright law represented by the rule is legally valid. Making this condition explicit enables us to argue about which theory of linking is correct and to compare the effects of these two theories on particular cases.

6. Arguments

Now let's use the theory of open source licensing issues we have constructed with OWL and LKIF rules to analyse a licensing issue of the hypothetical software project. Recall that in the example, the Carneades engine is implemented using the Clojure

³ http://en.wikipedia.org/wiki/GNU_General_Public_License#Linking_and_derived_works

programming language and links to the Clojure library, as well some others, including the Pellet library. The Clojure compiler and library are licensed using the EPL. The Pellet library uses the AGPL variant of the GPL. Both the EPL and AGPL are reciprocal.

Suppose we want to analyse whether the Carneades engine may be licensed using the Eclipse Public License (EPL). Using the Carneades editor, an interactive argument mapping tool integrated with an assistant which helps users to find and construct arguments from ontologies and rules, we can create an argument graph about this issue, as shown in Figure 1.

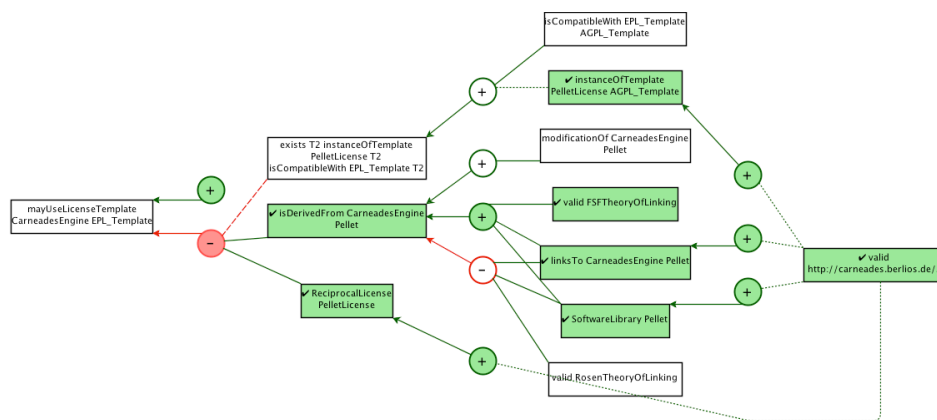


Figure 1. Example Argument Graph

Carneades argument maps are visualizations of argument graphs. An argument graph is a bipartite graph, with statement and argument nodes. Statement nodes are displayed using boxes; argument nodes with circles. The propositional content of a statement node is shown as text inside the box, in either natural language or, as in this example, in some formal language. Here statements are displayed as RDF triples of the form *predicate subject object*. For example, the main issue, about whether the Carneades engine may use the EPL license template, is shown in the box at the far left of the map, with the text:

CarneadesEngine mayUseLicenseTemplate EPL_Template

The map includes two arguments about this issue, a pro argument and a con argument. Pro arguments are visualized by circles containing a large plus sign; con arguments by circles containing a minus sign. In the example, the pro argument was constructed from the DefaultLicenseRule. Recall that this general rule states that, by default, a software entity may be licensed using any template license, with no limitations on the copyright owner. Since this rule has no conditions, the argument which has been constructed from it has no premises. Arguments are applicable when all of their premises hold. Arguments with no premises, such as this one, are always applicable. In the figure, applicable arguments and acceptable statements are visualized by filling their circle or box, respectively, with a gray background.

Arguments are linked to their premises and conclusion uses various kinds of edges between the statement and arguments nodes in the diagram. Solid lines denote ordinary premises; dotted lines denote exceptions.

The con argument in Figure 1 is not (yet) applicable, and thus shown with a white background in the figure, because it is unresolved whether the Carneades engine is derived from the Pellet library. (The last premise of the con argument, about whether the EPL is compatible with the AGPL, is an exception, and thus need not hold for the con argument to go through.)

As can be seen in the figure, the FSF and Rosen theories about whether or not linking creates a derivative work have both been applied, to construct pro and con arguments about whether or not the Carneades engine is derived from Pellet. The open question is whether either of these two legal theories is valid. If the FSF theory of linking is valid, then the Carneades engine is derived from Pellet and the con argument from reciprocity would be applicable. This con argument would then rebut the argument from the default license rule, with the result that it would no longer be acceptable to use the EPL license.

When analysing a case and constructing arguments, it is important to focus your efforts on relevant issues. In a recent conference paper [1], Stefan Ballnat and I presented a model of abduction for Carneades and show how it can be used to support goal selection. The model of abduction computes minimal sets of statements which, if true (accepted), would make a given statement *in* (acceptable or accepted) or *out* (not in). In our example, if the aim is to rebut the conclusion that the EPL may be used, one of the minimal positions computed by abduction is the singleton set:

{FSFTheoryOfLinking is valid}

7. Conclusion

This article is a summary of a technical report [5] written in the Qualipso project. Building on our prior theoretical and practical work on the Legal Knowledge Interchange Format (LKIF) and the Carneades argumentation system, we have developed a proof-of-concept, prototype system for helping developers to construct, explore and compare legal theories when analysing open source licensing issues in particular cases. In the future we hope to find an opportunity to develop the ontology and rulebase further and to validate in pilot applications the suitability of Carneades as a tool for constructing and evaluating alternative legal theories.

8. References

1. Ballnat, S. and Gordon, T.F. Goal Selection in Argumentation Processes — A Formal Model of Abduction in Argument Evaluation Structures. Proceedings of the Third International Conference on Computational Models of Argument (COMMA), IOS Press (2010), 51-62.
2. ESTRELLA Project. The Legal Knowledge Interchange Format (LKIF). 2008.
3. Gordon, T.F. and Walton, D. Legal Reasoning with Argumentation Schemes. 12th International Conference on Artificial Intelligence and Law (ICAIL 2009), ACM Press (2009), 137-146.
4. Gordon, T.F., Prakken, H., and Walton, D. The Carneades Model of Argument and Burden of Proof. Artificial Intelligence 171, 10-11 (2007), 875-896.
5. Gordon, T.F. Report on a Prototype Decision Support System for OSS License Compatibility Issues. Qualipso (IST- FP6-IP-034763), Deliverable A1.D2.1.3, 2010.
6. McGuinness, D.L. and van Harmelen, F. OWL Web Ontology Language Overview. 2004.
7. Rosen, L.E. Open source licensing: Software freedom and intellectual property law. Prentice Hall Professional Technical Reference, Upper Saddle River, New Jersey, USA, 2004.

Emotional Argumentation Schemes in Legal Cases

(Short Position Paper)

Martyn LLOYD-KELLY¹, Adam WYNER, Katie ATKINSON
University of Liverpool

In this short position paper, we outline points at which arguing about emotions is relevant in legal cases, a brief overview of related discussions, our position that emotions can be the object of argumentation, the analytic framework we adopt, and a case which we intend to model.

As emotions are a widespread, salient experience of our lives and in our social encounters, it is unremarkable that they are found the subject of legal proceedings, where human experience and behaviour is reasoned about and regulated. This is realised in a variety of ways. In common law, among the *causes of action* we find *intentional or negligent infliction of emotional distress*, for which a plaintiff may be due compensation. Over the course of litigation, there may be arguments as to whether distress was caused, the extent of distress, along with supporting evidence or expert testimony. In *hate crimes*, there may be focus on the emotional disposition of the perpetrator. Lawyers arguing a case might appeal to a jury with respect to pity, fear, or sympathy in an effort to sway a decision on behalf of their client. In such a circumstance, a jury might receive instructions to reason strictly about the facts of the case with respect to the law, leaving aside emotional appeals. In cases of particularly heinous crimes, the degree of outrage to the sensibilities is relevant in meting out punishment. The difference between *murder* and *voluntary manslaughter* can hinge on the emotional state of mind of the perpetrator and bear, among other issues, on 'emotional decay over time'. Finally, in coming to a decision, the judges may seek any relevant mitigating factors which warrant *mercy* and counterbalance an otherwise harsh decision. Following [13], we believe that making emotional argumentation explicit and formal, we can present better, clearer, and fuller representations of legal case arguments and decision making.

Recent work on formalisation of emotions is relevant to our discussion. There are broad claims about a distinction between *good, rational* argumentation forms (e.g. *All men are mortal, Socrates is a man, therefore Socrates is mortal*) and *fallacious, adjunct* argumentation forms (e.g. *ad hominem* or *appeals to pity*). In [11,12], fallacious arguments are conversational moves that, while appearing to contribute to the purpose of a conversation, interfere with it. In this view, emotional arguments have an adjunct status: "good" emotional arguments can be used to direct an agent towards a prudent course of action to achieve a desired

¹Corresponding Author: Department of Computer Science, University of Liverpool, L69 3BX, UK. Tel.: +44 (0)151 795 4271; E-mail: mlk5060@liverpool.ac.uk.

goal, while “poor” emotional arguments can detract from it. This is along the lines of related *teleological* reasoning as found in AI and Law [2]. Thus, normatively, one should only use good and avoid fallacious argument forms. While there are emotional argumentation schemes, the emphasis is on filtering “poor” arguments from the otherwise “rational” discussion rather than reasoning with them. Moreover, emotions are not integrated into a formal argumentation framework. [8] presents the Ortony, Clore, and Collins (OCC) model of the emotions for modelling agents: emotions are decomposed into reactions to events, agents, and objects; beliefs about the events may affect the emotion; formal lexical semantic representations of the emotion terms are provided; *intensity* is represented in terms of *unexpectedness* and *decay* [10]; as emotions are related to circumstances which give rise to them, they are defined in terms of preconditions and postconditions. However, the model does not itself support argumentation about the emotions. [14] integrates the OCC model into a decision-making model that uses an action formalism with the *Practical Reasoning Argumentation Scheme* [3], argumentation frameworks, and value-based argumentation [4]. In this analysis, emotions play an adjunct role of influencing agents’ decision-making with respect to what course of action to follow; emotions can increase or decrease the priority given to alternative value rankings, thereby influencing the choice of action. However, it is unclear to us whether this model serves in the legal arena since while practical reasoning does appear to be a primary mode of argument in legal proceedings, it is not clear how the emotional responses of various participants in a legal setting bears on the relevance to the legal decision.

Our position is that we can fruitfully integrate emotional argumentation schemes into a formal model of argumentation, making emotions the very *objects* of argumentation [6], rather than relegating them to an adjunct role. Thus, rather than ruling out, filtering out, or assigning a subordinate function to rational argument, emotions can be “first class” citizens of argumentation. Formally, we propose to use *emotional argumentation schemes* [5,11,8] that are represented as instantiations of arguments in the sense of ASPIC+ [9] and reasoned with in *argumentation frameworks* [1]. Representations of emotions in the OCC are represented as additional emotional argumentation schemes. Thus, as with other defeasible arguments, we can argue for or against emotional arguments, and we can use emotional arguments to attack other arguments. Where there are “preferences” among classes of arguments, we represent this explicitly: “fallacious arguments” can attack and be attacked by “rational arguments”; however, given a preference ranking, “rational arguments” can defeat, but not be defeated by, “emotional arguments”; we introduce a means to change the rankings where necessary.

Our analysis is particularly salient in legal contexts, where we propose to formally model the role and function of emotions in court cases. We only consider explicit, verbal representations of and arguments about emotional content. The analysis is illustrated with a discussion of some particular cases concerning domestic violence: *People v. Rupe* [(1998) 206 Cal.App.3d 1537 [256 Cal.Rptr. 126]] is a murder case concerning *the heat of passion*; and cases cited in [7], which represent a shift in values associated with women’s rights.

References

- [1] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and nperson games. *Artificial Intelligence*, 77:321–357, 1995.
- [2] C. Hafner and D.H. Berman. The role of context in case-based legal reasoning: teleological, temporal, and procedural. *Artificial Intelligence and Law*, 10:19–64, 2002
- [3] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10-15):855–874, 2007.
- [4] T. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
- [5] B. Manolescu. A normative pragmatic perspective on appealing to emotions in argumentation. *Argumentation*, 20:327–343, 2006.
- [6] R. Micheli. Emotions as objects of argumentative constructions. *Argumentation*, 24:1–17, 2010.
- [7] R. Moles. *Definition and Rule in Legal Theory: a reassessment of H.L.A. Hart and the positivist tradition*. Basil Blackwell, 1987.
- [8] A. Ortony, G L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.
- [9] H. Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
- [10] Juan D. Velásquez. When robots weep: Emotional memories and decision-making. In *Proceedings of American Association for Artificial Intelligence 1998*, pages 70–75, 1998.
- [11] D. Walton. *The Place of Emotion in Argument*. Pennsylvania State University Press, 1992.
- [12] D. Walton. *A Pragmatic Theory of Fallacy*. The University of Alabama Press, 2003.
- [13] D. Kahan. The Anatomy of Disgust in Criminal Law. In *Faculty Scholarship Series*. Paper 112.
- [14] F. Saud Nawwab, T. Bench-Capon, and P. E. Dunne. Emotions in Rational Decision Making. In P. McBurney et al. (Eds) *Proceedings of ArgMAS 2009*, LNAI 6057, pages 273-291, 2010.
- [15] A. Wyner, T. Bench-Capon, and K. Dunne. Arguments, values and baseballs: Representation of Popov v. Hayashi. In A. Lodder and L. Mommers (editors): Legal Knowledge and Information Systems. JURIX 2007: The Twentieth Annual Conference, LNAI 6057, pages 273-291, 2010. A. Wyner, T. Bench-Capon and K. Atkinson (2007): Arguments, values and baseballs: Representation of Popov v. Hayashi. In A. Lodder and L. Mommers (editors): Legal Knowledge and Information Systems. JURIX 2007: The Twentieth Annual Conference, Leiden, The Netherlands, pp. 151-160. *Frontiers in Artificial Intelligence and Applications*, Vol. 165. IOS Press, Amsterdam, The Netherlands.

MOSONG, a Fuzzy Logic Model of Trade Mark Similarity

Anna RONKAINEN¹

Department of Modern Languages, University of Helsinki, Finland

Abstract. This paper describes MOSONG, a knowledge-based system prototype implementing a model of an open-textured legal rule using fuzzy logic. The legal domain modelled by MOSONG are the relative grounds of refusal of registration of European Community trade marks. Second-order fuzziness is used in MOSONG as a representation of uncertainty. MOSONG was constructed to resolve a training set of 119 cases correctly, and in subsequent tests with a validation set of 30 cases, a correct solution was produced for all of them as well. In questionnaire-based follow-up tests with non-expert users (n=75), however, only $54.6 \pm 6.5\%$ of all cases received the correct answer, and $25.9 \pm 7.5\%$ an incorrect one.

Keywords. trade mark law, Community trade mark, trade mark similarity, uncertainty, vagueness, fuzzy logic

1. Introduction

Fuzzy logic is an obvious tool of choice for modelling knowledge with an element of vagueness, such as any type of knowledge based on natural language [1,2]. Vagueness is of course also a central issue in legal theory [3,4]. In artificial intelligence and law, models using fuzzy logic are well established and have been documented in at least one monograph [5] as well as numerous articles, such as [6,7,8]. So-called type-2 fuzzy logic systems, that is, systems in which the individual values used in the fuzzy logic rules are fuzzy sets themselves [9], however, have, to the best of the author's knowledge, not been previously used within the legal domain.

The main advantage of type-2 fuzzy logic is that it allows for the representation of second-order vagueness. That is, both the vagueness of a concept and the uncertainty associated with its application can be included in a model. The most simple variant of type-2 fuzzy logic is interval-type fuzzy logic, in which the membership value of a fuzzy concept can be either a singleton value representing vagueness without uncertainty or an interval with a lower and an upper bound representing vagueness with a range of uncertainty. In simple terms, traditional fuzzy logic allows us to say that *John is 0.9 TALL* (whatever that means), whereas with type-2 fuzzy logic we can also say that *John is between 0.85 and 0.95 (0.90 \pm 0.05) TALL*, in which the uncertainty or margin of error may stem from any source, anything from potential measurement errors to intrinsic design factors within the model. Instead of just a simple interval, the second-order membership

¹Corresponding Author: Anna Ronkainen, University of Helsinki, Department of Modern Languages, PO Box 24, 00014 Helsinki yliopisto, Finland; E-mail: anna.ronkainen@helsinki.fi.

function could of course also be something more complicated, such as the bell curve of a normal distribution.

In order to examine the applicability of type-2 fuzzy logic within the legal domain, MOSONG (Tok Pisin for 'fuzz') was constructed as a proof-of-concept prototype system representing a model of a single rule. MOSONG was implemented from scratch using Common Lisp, and as a minimal prototype system its features are limited to the bare essentials, there is for instance no user interface whatsoever apart from calling the functions implementing the model directly.

The jurisdiction covered by MOSONG is the European Union, and all references made in this paper are to EU statutory and case law and institutions, unless otherwise specified.

2. Domain

The Community trade mark (CTM) was established in 1996 to provide the possibility of registering a trade mark in all the member states of the European Community at the same time. Its statutory basis is the 1993 Community trade mark Regulation (40/94/EC). The CTM is administered by the Office for Harmonization in the Internal Market (OHIM), based in Alicante, Spain. The CTM is a parallel system alongside the national (and Benelux) trade mark systems, which were not changed by the establishment of the CTM [10].

The legal domain implemented by MOSONG are the so-called relative grounds for refusal of registration, that is, the relationship between the applicant's new CTM and preexisting CTMs and national marks, and whether the new mark's similarity to older marks is such that it can be considered to infringe on them or there is a risk of confusion between them.

The statutory provision modelled by MOSONG is Article 8, Paragraph 1 of the CTM Regulation:

Article 8

Relative grounds for refusal

1. Upon opposition by the proprietor of an earlier trade mark, the trade mark applied for shall not be registered:

(a) if it is identical with the earlier trade mark and the goods or services for which registration is applied for are identical with the goods or services for which the earlier trade mark is protected;

(b) if because of its identity with or similarity to the earlier trade mark and the identity or similarity of the goods or services covered by the trade marks there exists a likelihood of confusion on the part of the public in the territory in which the earlier trade mark is protected; the likelihood of confusion includes the likelihood of association with the earlier trade mark.

The crucial part here are the words 'identity or similarity' and 'likelihood of confusion'. Since there are no further provisions in the Regulation concerning the interpretation of these words, the rule is obviously quite open-textured. In Hart's terms, we could consider full identity as the core of the area of applicability of the rule, and lesser grades of similarity as its penumbra. As far as legal institutions go, the CTM is still fairly recent, but of course it was not a total innovation built on top of no previous jurisprudence, but rather it had a substantial foundation in the case law of the individual member states, in

all of which in general similar and equally open-textured provisions had become to be interpreted in a reasonably uniform fashion over the course of the past century or so.

In order to keep the system as simple as possible, the domain covered by MOSONG was restricted to word marks and combination marks (words as such or with a particular graphic design), as word marks clearly are the easiest ones to compare for similarity, both through a raw similarity metric and with descriptive options to be chosen by the end user. This is by no means a fixed characteristic of the system, with compatible additional (but naturally much more complex) similarity metrics the prototype could be extended to cover other types of marks as well.

It should be pointed out that MOSONG was originally implemented in 2002 and reflects the law in force as of that time. As such, it may already be an exercise in computational legal history rather than an accurate model of the law at the present time. The original Regulation has recently been replaced by a new one (207/2009/EC), but Article 8 remains essentially unchanged.

3. Implementation

The text of the provision offers an obvious formulation for the rule covered by MOSONG, as long as we consider identity simply as a limiting case of similarity:

REFUSAL = MARKS-SIMILAR and GOODS-SIMILAR

3.1. Case Law Analysis for the Training Set

By itself this is of course not very helpful, and so a more precise meaning of the two similarity conditions must be found through the use of case law. For MOSONG, an obvious source of case law providing a comprehensive coverage of the domain were all the relevant cases cited in a trade mark law textbook [11]. The cases concerned were decided either by the OHIM Opposition Division as the first instance, or the OHIM Boards of Appeal as the second instance.² Only cases in which a decision was made as to the presence or absence for relative grounds of refusal of registration were included. A number of Opposition Division cases were for example excluded because the opposition failed due to formal grounds such as lack of documentation for the existence of opponent's earlier trade mark right rather than the grounds covered by the rule being modelled. Out of a grand total of approximately 300 cases cited in the textbook, 119 were such that they could be included in the training set: 107 from the Opposition Division and 12 from the Boards of Appeal.

3.2. The Training Set

The cases in the training set were initially read summarily by the author in search of a preliminary set of factors for classifying the cases in terms of input conditions for the provision being modelled. After this they were read in greater detail and classified ac-

²Further appeal is possible to the Court of First Instance and finally to the Court of the European Union, but because of the complexity and one-of-a-kind nature of cases that reach so far, such cases were not considered for inclusion in MOSONG.

cording to a template created after the first round. The classification scheme was however not frozen at this point, several further alternatives were added during the classification stage and a few also during the weighting and adjustment stage. Because of the type-2 fuzzy design, it was not necessary to settle for a single choice for factors of cases that were difficult to reconcile with the classification scheme, but rather capture the potential ambiguity of the case law in its representation in the training set as well. In the final version of the training set, this possibility was used in almost one fourth (28) of all the cases. Similarly, a few of the weights used for the individual factors were also designated using intervals rather than singleton values as well.

Once the cases had been categorized, initial weights were assigned for all the factors. Subsequently the outcomes predicted by the model were calculated and compared against the actual outcomes of the cases. After a fairly modest number of iterations with appropriate changes to both the categorization scheme, the weights used for the individual factors, the fuzzy logic expression of the rule itself and the defuzzification parameters used, MOSONG was able to produce the correct answer to all the cases in the training set.

3.3. The Final Model

The final version of all the parameters used in the case vectors is as follows:

CASE-ID	OHIM case identifier, eg "R109/1999-2"
OPPONENT-MARK	opponent's (earlier) trade mark, eg 'SUNRISE'
APPLICANT-MARK	applicant's (newer) trade mark, eg 'SUNSET'
OPPONENT-COMMON	the fraction of letters maximally shared (ignoring intervening letters) by both marks divided by the length of OPPONENT-MARK (also ignoring whitespace, accents and punctuation), eg 0.714 for the two marks above
APPLICANT-COMMON	as above, but divided by the length of APPLICANT-MARK, eg 0.833 for the two marks above
BEGINNING-IDENTICAL	whether both marks are identical in the beginning of the mark
1	yes (at least 40 % of the shorter mark, single-character difference permitted)
0.8	identical except for an article (or other non-dominant word)
0.6	in the first word only the first character is different
0	no
DIFFERENCE-DISTINCTIVE	whether the difference of the marks is distinctive
1	yes
0.8	(pseudo-)morpheme-like difference with non-descriptive (or no) meaning, or product used by specially qualified personnel only
0.7	visually or phonetically distinctive signs
0.3	descriptive difference (eg place)
0.1	single-character/article difference; visually or phonetically similar
0	no difference, or differing parts descriptive and synonymous

COMMON-PART-DISTINCTIVE	whether the common part shared by the marks is distinctive
1	unrelated to the goods/services, dominant, or no difference
0.7	related to the goods/services, somewhat descriptive
0.4	trivial in the context
0	common part non-transparent or subordinate to the visual appearance
OPPONENT-GOODS-SPECIFICITY	whether the list of goods/services of the opponent is specific (few well-individualized products) or not (entire trade mark classes)
1	very specific
0.8	somewhat specific
0.4–0.5	whole trade mark class(es)
APPLICANT-GOODS-SPECIFICITY	as above, for the applicant
GOODS-OVERLAP	the relationship between the lists of goods and services for both marks
1	both are approximately the same at least within the classes covered by the applicant
0.8	applicant's list is a subset of that of the opponent
0–1	opponent's list is a subset of that of the applicant
0.2	the lists are different but within the same class
0.4	there is a whole-component relationship between the two
0.5	the lists are otherwise somewhat related (even if they might be in different classes)
0	no overlap
SPECIAL-FACTORS	various factors overriding normal consideration
1	high reputation: marks such as 'NIKE', 'BOSS', 'OLYMPIC', 'SEMCO', 'RETROVIR'
0.7	visually identical marks
OPPOSITION-UPHELD	whether relative grounds for refusal of registration exist according to the original decision being modelled
1	yes
0	no
0–1	the opposition is only upheld for the goods and services registered for the opponent; for the other goods and services the opposition is rejected

In the final version, the rule had evolved considerably and was expressed as follows:

REFUSAL = MARKS-SIMILAR and GOODS-SIMILAR;

in which

MARKS-SIMILAR = ((APPLICANT-COMMON or OPPONENT-COMMON or
 BEGINNING-IDENTICAL or
 (dil(APPLICANT-COMMON) and
 BEGINNING-IDENTICAL))
 and
 COMMON-PART-DISTINCTIVE
 and
 not(DIFFERENCE-DISTINCTIVE))
 or
 (OPPONENT-COMMON and SPECIAL-FACTORS);

and

GOODS-SIMILAR = (GOODS-OVERLAP and OPPONENT-GOODS-SPECIFICITY and
 APPLICANT-GOODS-SPECIFICITY)
 or
 (dil(GOODS-OVERLAP) and
 conc(OPPONENT-GOODS-SPECIFICITY) and
 conc(APPLICANT-GOODS-SPECIFICITY))
 or
 (conc(GOODS-OVERLAP) and
 dil(OPPONENT-GOODS-SPECIFICITY) and
 dil(APPLICANT-GOODS-SPECIFICITY))
 or
 SPECIAL-FACTORS;

The fuzzy and and or operators were implemented as the minimum and maximum operators. The concentration and dilation operators were defined as square and square root, respectively.

As can be seen from the final version of the rule, a considerable number of factors were necessary in order to represent all cases accurately. With a minimal version of the rule, in which

MARKS-SIMILAR = APPLICANT-COMMON or OPPONENT-COMMON;

and

GOODS-SIMILAR = GOODS-OVERLAP and OPPONENT-GOODS-SPECIFICITY and
 APPLICANT-GOODS-SPECIFICITY;

only 73 out of the 119 cases would receive the correct result.

3.4. Legal Analysis of the Model

A number of conclusions relevant to the legal analysis of the rule can be drawn from the model. Crude similarity of the marks as measured through the proportion of common characters in both marks is a reasonably good predictor, the dividing line being somewhere in the vicinity of the 50% mark, but other factors concerning the location and type of the common part had to be considered as well. The beginning of the mark is especially

significant, which is appropriate considering its particular importance for human word recall using the availability heuristic [12,13]. The perceived morphosyntax of the mark is important as well, as are the semantic meanings of both the shared and the distinctive parts.

The judgement of similarity of goods and services for the two marks was discovered to depend on two factors: to what extent the lists overlap or at least approach each other and how specific they are. Specificity is assessed relative to entire trade mark classes of the Nice classification. Claims including an entire class (typically by repeating the Nice classification description verbatim) are considered unspecific, a short list of individual products (usually something that really could be considered to be sold under the same brand in practice) are considered specific. In terms of overlap, unspecific marks must match each other closely, whereas specific marks may only have to be related in some respect, as captured in the model through the two versions of the similarity assessment using dilation and concentration.

The SPECIAL-FACTORS parameter was included to cover a few borderline cases regarding the applicability of the model. This concerns two different categories of cases. The first category concerns the limits of the implementation of the model by design through limiting it to word marks only. For graphically similar combination marks, the graphic design of the marks may be decisive and similarity of the marks as words may be of less importance. The second category concerns the limits of the applicability of the modelled provision itself. In accordance with the so-called Kodak doctrine, marks of high reputation receive a wider range of protection than other marks. In terms of CTM law, such cases are decided based on Paragraph 5 rather than Paragraph 1 of Article 8 of the CTM Regulation. In a more comprehensive system, both categories of cases would of course be handled by separate modules implementing a comparison of graphic similarity and the rule in Paragraph 5, respectively.

The model is intended as predictive and thus mainly descriptive, but by virtue of being a description of courts' (or comparable instances') reasoning, it can at the same time be considered normative.³ Most importantly, the model contains a transparent reproduction of the statutory provision itself. Of course, the result produced by the model is not in itself sufficient to show that a case could be incorrectly decided, but the two core similarity dimensions would allow for easy retrieval of nearest neighbours from the knowledge base for critical comparison. Even if all of the most similar cases agree with MOSONG, it is however still possible that the decision is correct if it is for instance based on factors not included in the model. On the other hand, it is of course also possible that MOSONG draws the line slightly in the wrong place, but the safety margins in defuzzification make this unlikely.

3.5. Type-Reduction and Defuzzification

Based on the parameters of the case vector and the logic rule used in the model, MOSONG computes an initial result, which must subsequently be processed and interpreted in order to see whether the result is correct. Because of the type-2 design used in MOSONG, the initial result may be either a singleton fuzzy value or an interval. In case of intervals the initial result must first be type-reduced into a singleton value in order to get a comparable result. Intervals were type-reduced by calculating the average of the upper and the lower

³At least for a legal realist, whether American [14] or Scandinavian [15].

bounds, except in cases where the lower bound was below 0.25 and the upper bound above 0.75, in which cases the interval was preserved to the next stage without type reduction.

The output is then defuzzified to yield a crisp (yes/no) result. In order to make the model more robust in handling ambiguity, a double threshold was used. After type-reduction, singleton output values of 0 to 0.48 were defuzzified to a 0 (opposition rejected), values of 0.63 to 1 to a 1 (opposition upheld). Values remaining as intervals after type-reduction were interpreted as the special case in which the application may only proceed for some of its goods and services, provided the GOODS-OVERLAP parameter would remain an interval after same type of type-reduction as the initial result. Other intervals and singleton values between 0.48 and 0.63 were interpreted as an unclear result. As pointed out earlier, it was possible to make MOSONG output a correct result for each case in the training set relatively easily. A result is deemed correct when it matches exactly the OPPOSITION-UPHELD value encoded in the case vector.

4. Validation

Since MOSONG was designed so as to give the correct answer to each of the cases of the training set, it was also necessary to test its performance independently with a different set of cases. For this purpose, a validation set was collected, with the most recent decisions published at the time by both the OHIM Opposition Division and the OHIM Boards of Appeal representing a sufficiently random and varied cross-section of cases intended to be handled by the system.

4.1. Validation Set

The validation set consists of 10 (presumably more difficult) cases from the Boards of Appeal and 20 (presumably easier) cases from the Opposition Division. The initial target was set to a higher number of cases, something comparable to the training set, but since only approximately 10% of all cases reviewed were such that they could be used for this purpose, this target could not be met in practice. The validation set was categorized by the author using the same procedure as for the training set, except that the model itself was of course kept unchanged during the process.

A test run with the coded validation set resulted in a perfect score: all cases were decided correctly. As such, the model could certainly be considered a success.

4.2. End-User Validation

On the other hand, the perfect score did seem perhaps a bit too good to be true. Since the categorization of cases in the validation was done by a domain expert (the author), it could certainly be argued that perhaps all the expertise supposedly exhibited by the system was actually simply located in the user rather than the system itself.

In order to test this hypothesis, MOSONG has since also been tested on non-expert users through questionnaires as an assignment on a course on intellectual property law aimed at non-law students at the University of Helsinki. A final round of tests of this type will be carried out during the spring term of 2011, after which a more comprehensive analysis of the results will be carried out and published in a subsequent paper. Already

at this point it can however be pointed out that the hypothesis does indeed seem to be true, at least to some degree. Using the 30-case validation set on 75 test subjects thus far, only $54.6 \pm 6.5\%$ of all cases produced the correct result. On the other hand, an incorrect result was only produced in $25.9 \pm 7.5\%$ of the cases, and in the remaining $19.5 \pm 5.2\%$, MOSONG was unable to produce a conclusive answer due to excessive ambiguity present in the input.

The results of the end-user validation do however provide a sufficiently varied set of data for evaluating the impact of potential changes to the system compared with the training set and the original (100% correctly coded) validation set. The most obvious of the potential changes would be an adjustment of the defuzzifier using a wider range for the threshold, thus changing at least some of the results from incorrect to inconclusive. The number of inconclusive answers in turn would certainly be lower in an interactive system, in which users would be allowed to reconsider their answers and see how changing any of them might affect the result. Quite to the contrary, in this questionnaire-based setup, on the other hand, ambiguous answers (multiple choices for the same parameter) were actually even encouraged in the instructions.

5. Discussion

There is at least one previously published model of the same legal issue, CATMiNE by Renaux and others [16]. It is easy to point out a considerable number of superficial differences between CATMiNE and MOSONG: CATMiNE is based on neural networks and machine learning and it is self-contained and intended for production use as an advisory system for trade mark lawyers, whereas MOSONG is constructed manually using fuzzy logic, 'outsourcing' some of the more difficult reasoning tasks from the user by framing them as non-legal questions, and only intended as a simple prototype for research purposes. Perhaps the most striking difference however is in performance: CATMiNE produces an average error rate of 28%, compared against MOSONG's 0% for both the training and validation sets with an expert user (with non-expert users, MOSONG is approximately on par with CATMiNE). However, there appears to be a simple explanation for this. In the data structure used by CATMiNE, goods and services are only represented at the trade mark classification level, and at this level of granularity the kind of comparison used in MOSONG remains impossible. In terms of the underlying knowledge representation of the similarity assesment used in the two models, they do nonetheless resemble each other considerably, and the more sophisticated components used for comparison in CATMiNE could probably be used directly as replacements for some of the factors MOSONG now has to get from the user.

6. Conclusions

This experiment shows that it is at least within this particular domain indeed possible to represent a substantial body of case law as a more accurate description of a vague statutory provision with a relatively simple model using fuzzy logic. With an interval type-2 system, it is also possible to represent uncertainty in an easily understandable fashion.

The validation part of the experiment shows on the one hand that the model works correctly also for cases outside the training set, but on the other hand, testing with non-experts also highlights the role of the user when a knowledge-based system is being evaluated or used in general.

It should be pointed out that neither of the two textbooks cited provides anything nearly as concrete in their descriptions of the rule, and already as such the model may provide some valuable input for the study of trade mark law in general and not just in an AI & law context.

Acknowledgements

This research has been funded in part by a grant from the Finnish Cultural Foundation.

References

- [1] Lotfi A. Zadeh. Fuzzy Logic = Computing with Words. *IEEE Transactions on Fuzzy Systems*, 4:103–111, 1996.
- [2] Timothy Williamson. *Vagueness*. Routledge, London, 1994.
- [3] H.L.A. Hart. *The Concept of Law*. Clarendon Press, Oxford, 1961.
- [4] Timothy A. O. Endicott. *Vagueness in Law*. Oxford University Press, Oxford, 2000.
- [5] Oliver Munte. *Die Konkretisierung unbestimmter Rechtsbegriffe mit Hilfe von Fuzzylogik am Beispiel des § 1610 BGB*. Schöningh Verlag, Münster, 2000.
- [6] Leo Reisinger. Über die Anwendungsmöglichkeiten der Theorie unscharfer Mengen im Recht. *Datenverarbeitung und Recht*, 4:119–157, 1975.
- [7] Lothar Philipps. Vague Legal Concepts and Fuzzy Logic: An Attempt to Determine the Required Period of Waiting After Traffic Accidents. *Informatica e diritto*, pages 37–51, 1993.
- [8] Jacky Legrand. Some Guidelines for Fuzzy Sets Application in Legal Reasoning. *Artificial Intelligence and Law*, 7:235–257, 1999.
- [9] Jerry M. Mendel. *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice-Hall, Upper Saddle River, NJ, 2001.
- [10] Ruth E. Annand and Helen E. Norman. *Guide to the Community Trade Mark*. Blackstone Press, London, 1998.
- [11] Harri Salmi, Petteri Häkkänen, Rainer Oesch, and Marja Tommila. *Tavaramerkki*. Kauppakaari, Helsinki, 2001.
- [12] Amos Tversky and Daniel Kahneman. Availability: A Heuristic for Judging Frequency and Probability. *Cognitive Psychology*, 4:207–232, 1973.
- [13] Rebecca Tushnet. Gone in 60 Milliseconds: Trademark Law and Cognitive Science. *Texas Law Review*, 86:507–568, 2008.
- [14] Oliver Wendell Holmes. The Path of Law. *Harvard Law Review*, 10:457–478, 1897.
- [15] Alf Ross. *On Law and Justice*. University of California Press, Berkeley & Los Angeles, 1959.
- [16] Pierre Renaux. CATMinE – Computer Assisted Trade-Mark Infringement Evaluation. In Danièle Bourcier, editor, *Legal Knowledge and Information Systems. JURIX 2003. The Sixteenth Annual Conference*, pages 61–70, Amsterdam, 2003. IOS Press.

Say “cheese”: Natural kinds, deontic logic and European Court of Justice decision C-210/89

Burkhard SCHAFER^{a,1}

^a*University of Edinburgh, SCRIPT*

Abstract. The paper develops a semi-formal framework to analyse the semantic of the European Court of Justice decision C-210/89 on Geographical Indications. It is shown that the debate regarding the appropriateness of giving the term “Feta cheese” legal recognition mirrors discussions in analytical philosophy about the semantics of natural kind terms. A logic with formal quotation function is introduced that allows to represent the different arguments that the competing sides presented, and allows to see the philosophical and linguistic intuitions that inform them..

Keywords. Possible world semantics, Quotation logics, EU case law, rigid designation

Introduction

The example that is at the centre of this paper are the attempts by the European Union to create legal protection for geographical indications (GI), and in particular the debate concerning the designation “Feta”, contested between Greece, Germany and Denmark which the European Court of Justice had to adjudicate in its decision C-210/89 [2,8,13]. We develop a simple formalism that displays possible disambiguation of the semantic of the relevant norms and the court decision that settled the debate. This will allow us to show that the court in its reasoning took up themes and ideas that are also discussed in analytical philosophy, in particular the discussion on the semantics of natural kind terms following the work of Saul Kripke[9].

1. Natural kinds – from language to law

In the case that is at the centre of this analysis, Germany and Denmark challenged the decision of the European Commission to award the designation “Feta” exclusively to sheep’s cheese produced in Greece. If unsuccessful, this would in effect prohibit to market cheese produced in Germany and Denmark whose taste was undistinguishable from the Greek produce as “Feta”. The case is unusual in that it requires to analyse and formally represent norms about word usage, in a multi-language environment. Traditionally in legal AI, the analysis is restricted to decisions where the language of

¹ Corresponding Author.

the court, and the language of the parties, is the same. This removes from the beginning certain issues of word meaning, and allows to disambiguate contested terms sufficiently to create unique formal representations of the norms in question. That is to say, the analyst or AI researcher can use his favourite theory of meaning as a tool to prepare the decision for formalisation, without having to fear to prejudice in this way the legal discussion with extraneous factors that have no legal standing (like philosophical theories of semantics). In our case, the situation is different. What is argued between the parties are not just norms and their meaning, but the meaning of meaning itself. Without realising it, the parties use as arguments (fragments of) theories, intuitions and ideas that are also prominently discussed in the philosophy of natural language. The task, for the analyst, is therefore to find a mode of representation that allows to express these different intuitions, ideas and arguments faithfully, without letting our own preferred theory of meaning pre-empt the legal argument.

Before we begin with the analysis though, some preliminary discussions about the background of the discussion within the logic community are necessary. Classical logic, the logic that is still in one way or the other builds the foundation of most AI systems, is an extensional language. That means that two terms that refer to the same object (have the same extension) can be replaced in all contexts without changing the truth value of the sentence. If Peter is taller than Paul, and Paul = The serving president, the Peter is also taller than the serving President. While in mathematics and the natural sciences, most contexts that are of interest for AI researchers are extensional and can be adequately modeled using classical logic, there are numerous examples in natural language that violate that principle, many of them of particular interest for law. If Peter e.g. attempts to kill Paul, we may not be entitled to replace “Paul” by “the serving president”, since Peter may not be aware of the fact that the two are identical. The legal implications are obvious: depending on the legitimacy of the substitution, we may be able to charge Peter for both attempted murder and treason, or only for attempted murder. Modal terms such as “believes that”, “intends to” or “promises to” introduce referential opacity into language, and mere co-extensionality is not enough to guarantee that the truth value of a statement remains the same after substitution of co-referential terms. Since the 1950s and the work by Saul Kripke, intensional contexts like this are normally analysed within the framework of a “possible worlds semantics”. To permit substitution, it is not sufficient that two terms contingently refer to the same object in this world, they also have to have the same referent in all relevant alternative worlds. In the case at hand, if Peter knows that Paul is the serving President, then Paul and “the serving president” refer to the same individual in all possible worlds that are consistent with everything Paul believes. In this case, substitution is permissible – if he attempts to kill Paul, he also attempts to kill the serving President. Similarly, if the context is an alethic modality, the substitution must be valid in all possible worlds simpliciter, and if it is a deontic modality, in all worlds that are deontologically perfect.

This elegant and intuitive solution raised however immediately several new problems. One of them is the issue of transworld identity: how do we “pick out” Paul and “the serving President” in other possible worlds to see if they refer indeed in all of them to the same individual? Saul Kripke, in *Naming and Necessity*, invites us to consider the following thought experiment [9]: Imagine a French citizen, Pierre. Pierre, proudly French and monolingual, believes on the basis of some films he saw the following: “Londres est joli.” (“London is beautiful.”) Much later, Pierre moves to London without realizing that London = Londres. In England, he immerses himself in British culture and learns English the way a native speaker would learn it, without

translating from French. Pierre lives at this time in a very unattractive part of London, which he now associates with the word “London”, so he comes to believe that London is not beautiful. According to Kripke, Pierre now believes both a) that London is “joli” and b) that London is not beautiful. Are there now two Londons in our possible worlds? Or just one – but then which one? Is Pierre holding a contradictory belief? Part of Kripke’s answer was to argue that proper names behave in an interesting way differently from other linguistic entities. They are “rigid designators” that pick out in every possible world the same object [3]. While the referent of “the British capital” might change (it could have been York, or Lancaster) and hence pick different objects in different worlds, “London” and “Londres” pick in all possible worlds the same object, that is London. At this point, we should note a certain asymmetry in access to information that makes this story intelligible for us. We know, from an outsider perspective, that the terms “London” and “Londres” refer to the same object, London. Pierre, from the insider perspective, has no access to this information. This allows us to judge if Pierre’s believes de-dicto – that is under the expressions which he uses in his mental language, refer to the same object de-re, in the outside world. This de-re and de-dicto distinction, and the outside and inside distinction, are crucial to see the example later as an analogy for legal reasoning. Courts, so we will argue, assumes the de-re position of the reader in Kripke’s example, while the plaintiffs are like Pierre, limited to de-dicto beliefs. The logic that we propose later on will be able to express this difference, but will do so somewhat differently from what the reader familiar with possible world based modal logic might expect.

As an obiter, Kripke also considered that terms other than proper names may be candidates for rigid designation, in particular natural kind terms such as “gold” or “water”, and the list has been expanded by other analytical philosophers [4,6]. Defining what natural kinds actually are is not straightforward, and often we find a mixture of examples and diagnostic tests. The term was brought into contemporary philosophy by W. V. Quine in his essay “Natural Kinds”, where any set of objects forms a kind only if it is “projectible”, meaning judgments made about some members of that set can be extended by scientific induction to other members. [11]

Hilary Putnam developed Kripke’s idea further, again in the form of a thought experiment, the “Twin earth” experiment [4]. Assume that there is a planet exactly like earth in virtually all respects, our ‘Twin Earth’. On that Twin Earth, we can find a Twin equivalent of every person and thing here on Earth. However, there is one small difference. In place of water, on Twin Earth there is a liquid that while superficially identical to H₂O, is chemically different, which we express by assigning it the chemical formula ‘XYZ’. The Twin Earthlings, in a language they call ‘English’ call XYZ ‘water’. Let us now assume in addition that neither earthlings nor twin-earthlings have yet discovered the chemical make-up of their respective liquids, that is they can’t of know that the liquids they called ‘water’ were H₂O and XYZ respectively. The way earthlings experience H₂O, and twin-earthlings experience XYZ is hence identical. For Putnam, this raises the following question: when an earthling and his twin on Twin Earth say ‘water’ do they mean the same thing?

Again, part of the argument is the intuition that water is a rigid designator, and therefore picks the same object in each possible world, and that it is hence necessarily H₂O [12,7] Umberto Eco finally gave this thought experiment a twist, that makes its relevance for legal discourse, and European law in particular, abundantly clear [5]. He sends astronauts on a mission to boldly go where no man has gone before. During their journey, they reach Twin Earth, and, firmly believing in the doctrine of rigid

designation, drink a liquid which the natives of the planet call “water”. Unfortunately, it turns out to be hydrochloric acid, and the explorers find an untimely end.

Does this thought experiment show that Putnam’s analysis is so counterintuitive that it should be abandoned? Whatever we think of Eco’s planet, it is for sure not a “deontologically ideal” alternative to our world. Unfriendly natives fraudulently bringing the lives of heroic explorers at risk? We surely can’t have that! And so, we would expect in “real” worlds the law to prevent this kind of disaster. Concerns like this, if a bit less drastic, are indeed the objective of the European Commission regulation 1829/2002 and a whole family of related legislation, administrative decrees and court judgments of a similar kind [1,13]. If we now think of possible worlds in a deontic context as jurisdictions, then we can make a direct connection between Kripke’s and Putnam’s thought experiments, and the structure of legal rules that prescribe a uniform word usage in a European context. What these norms have in common is that they “rigidify” the use of words in the labeling of foodstuff, that they prescribe specific expressions as names for specific products. Commission regulation 1829/2002 contains a specific kind of normative statements: norms which prescribe or prohibit the use of specific expressions for certain products. It *inter alia* prohibits the use of the name “Feta” for cheese not made from goat’s and sheep’s milk. What distinguishes this norm from more conventional legal regulations is that they do not prescribe so much an action (under a description?) but rather a description (for an action?). The claim that is made here is that these norms have the effect of making reality, and in particular language usage, closer to the ideal, or theories of meaning expressed in analytic philosophy of language. Where formal logics for intensional contexts require idealized assumptions of rationality, law posits normative rules that sanction certain type of language use over others.

2. Rigidifying “Feta”

Before we give a formal account, we need a bit more background information about the problem that the court had to address [see 2;13]. In the 1990s, Greek Feta producers realised that their export of Feta cheese to other European countries was plummeting. When they investigated the reason, they found that this was not because of a drop in demand for Feta cheese. Rather, much more products were sold under the label “Greek Feta cheese” than produced in Greece at that time. Even more outrageous, more cheese was sold than the production of sheep’s milk in Europe could possibly allow. In fact, most of the Greek Feta cheese in circulation was produced in Germany and Denmark, using cow’s milk instead of the original recipe. If Putnam’s fictitious explorer had instead of looking for the stars decided to explore the nearby neighbourhood travelling through Europe, what would their findings have been? One answer could have been: “Feta is not necessarily sheep’s cheese”, as an empirical finding. This was indeed the opinion of the German Government, presented in court as argument against the designation “Feta”. They argued against the Greek claim that far from using an inappropriate expression to refer to cow’s cheese, the Greek had become victim of an ambiguity. As good analytical philosophers, the German representatives proposed an analytical distinction: “Feta” as a word of the German language would refer to both sheep and cow’s cheese, as long as it had the same flavour. “Φέτα” on the other hand might well be an expression of the Greek language, and designate sheep’s cheese

only. The relevant theory of meaning is meaning as usage, and this allows for German and Greek users of the term to develop different associations.

The commission was not impressed. Arguably, the fact that on the packaging of “German Feta”, people in Greek costumes would dance in front of interested looking sheep did not help their case. Consequently, the Commission maintained that a) Feta is (de-re) sheep’s cheese and that b) this was the case in all member states. German manufacturers as a result simply lied when selling their product as Feta.² The Greek Government applied as a result of their findings for a “protection of origin order”, a means available to them under European consumer protection law. To gain the required protection, they had to show that consumers buying “Greek Feta” would normally assume that they would get indeed Greek Feta, and with that a product that a) is physically different from other products on the market and b) owes this difference to a form of production/raw materials specific to a geographical region. In that case, they were (initially) able to argue that both conditions were met, and the registration of Greek Feta as a sheep milk product from Greece was approved. This created a new norm: You ought not to sell something as “Feta”, unless it is Feta. What logical form does this norm have? Let us first contrast it with a more familiar form of norm, a norm requiring a simple action, for instance:

1) You ought to sell (some) Feta

as opposed to

2) You ought to sell as Feta only those cheese which are Feta

We might formalise the first in classical possible world systems with an OUGHT operator as

$$1a) \exists x Fx \text{ Ought}(Sx) \quad F: \text{Feta}, \quad S: \text{Sell}$$

But to analyse the second sentence, we need to find a way to represent the syntactical structure of the “as”-qualification, and give it a semantic interpretation. One might be tempted to use again modal logic

$$\#2b) \forall x \text{ Ought}(S(f,x) \rightarrow Fx) \quad f: \text{feta}, \quad F: (\text{being}) \text{Feta} \quad S: \text{sell as}; \quad f: \text{feta}, \quad F: (\text{being}) \text{Feta} \quad S: \text{sell as}$$

But this is in every respect a poor solution. It departs considerably from the syntactic structure of the natural language sentence, and introduces a strange “Feta-object”, which furthermore has no logical connections to the property of “being Feta”. But even if we disregard this for the moment, it could not work. Assuming that “Feta” is an adequate translation of the German “Schafskäse” (and hence Feta = Schafskäse a necessary analytical truth), it would follow from 2b) that if you sell something as “Schafskäse”, it has to be Feta. But this is *not* what the regulation requires from you. To make this clearer, let’s go back to Putnam’s explorers. If you insist on producing mineral water containing sources of a dangerous substance, e.g. hydrochloric acid, your customers might want to know. Likewise, to insert a label in Gaelic if you intend to sell them in Italy won’t do. Indeed, consumer protection regulation of the European Union

would rule out something like this as well.² This consumer protection norm has the same surface structure as the regulation under discussion so far. But if the Gaelic word and the Italian are (ex hypothesis) translations of each other, then they designate the same individuals in *all* possible worlds. This however means that they are substitutable in classical modal logics.

As mentioned above, we are dealing here with the problem that legal norms are typically language-relative. To preserve the meaning of the norm, and to preserve the syntactical structure of the natural language expression, we have to enrich our logic. A very natural way of doing this is to introduce a formal quotation operator. We add to our vocabulary a new function constant, the quotation function “”, and require syntactically that

if p is a term, then “p” is an object parameter

As long as we do not allow quantifying into the brackets, this should not pose any problems. (This becomes relevant if we think of the obligation for member states *to develop* a classification system e.g. for dangerous substances. This might well create the obligation, say for Germany, to introduce *some* single expression for HCl (e.g. “HCl”, “Salzsäure”, “Echt giftiges Mistzeug” etc.) so that every producer is under the obligation to use this and only this word in labelling his product)

For the time being, we might then re-formulate 2b)

$$2\beta) \forall x \text{ Ought}(S(x, \text{“F”}) \leftrightarrow Fx) \qquad \text{S: sell-as}$$

This quotation function seems to be a necessary addition to our formal instruments in any case, and it has applications above and beyond the very specific problem discussed here. Are we now safe? This depends on some crucial semantic choices. The problem now shifts to the interpretation of the sentence “Fx”. Let’s assume we have opted for a semantic with “nested domains”. This would mean that “Feta” is not a rigid designator, that there is no non-trivial necessary truth connected with it. If this were the case, it would mean that “Feta” can have different referents in different possible worlds, and from the outset, this seems to be a rather innocent assumption. “The Greeks could have used cows instead of sheep” seems to be a rather innocent statement about possible events in the past. However, this would mean that we can think of a possible world where 2β is fulfilled (and hence, this world is a deontologically preferable alternative to one where it is violated), and nevertheless, what they call “Feta” in this world is cow’s cheese. This however is exactly what the regulation prohibits. It seems that we must assume that “Feta is sheep cheese” is a necessary truth, and hence, that Feta is a rigid designator.

With the quotation function available to us, we can avoid this problem, *if* we are prepared to treat legal norms as linguistic entities. The deontic operators in this approach become predicates that take specific objects, quoted expressions, as arguments. To see norms as linguistic entities comes rather natural to the lawyer, or so I would argue. We typically deal with books, court reports, statutes, and learn to present cases in court, and to draft contracts and submissions. Not only that, but legal norms are typically “linguistically relative”. That is to say, two norms of different legal systems can be perfect translations of each other, and are nonetheless not identical.

² Commission Communication 93 (456)

“A contract made under duress is void” is a norm of Scots contract law, but not of German law, even though its perfect translation: “ein unter Drohung gemachter Vertrag ist nichtig” is. But since there is a norm that states that all norms in German law have to be in German, the two must be distinguished.

If we can therefore accept that legal norms are linguistic entities, we can introduce formulations of the kind:

3) Ought "S ("F", x) ↔ Fx"

“Ought” is a one place predicate, which takes a linguistic object, the quotes sentence “S(“F”, x) ↔ Fx” as argument. Note that “ought” is not a deontic operator, and there are no semantic rules or axioms governing its use. The logic is a conservative extension of first order logic, not a logic of norms. The norm is de-dicto in the sense that it does not presuppose the existence of Feta, nor does it allow substitution of co-referential or even synonymous expressions. Normally, we would assume from our discussion that this is the structure of legal norms proper, expressing the idea that they are language-relative in the sense discussed. The relative German and Greek norms about Feta and φετα are no less contradictory then Pierre’s de-dicto believes about Londres and London

4) Ought “S(“F”,x) ↔ Fx” S: sell as. F: Feta

5) Ought “S(“φ”, x) ↔ φx” S: Sell as, φ: φετα

However, European law is different from these de-dicto norms of the individual legal systems, as it operates in a multi-language environment. EU law is not only higher in hierarchy of norms - the legal aspect of transnational law - but also speaks about the individual legal systems which it comprises in a uniform language - the meta-linguistic or philosophical aspect of transnational law. In taking an outsider position towards individual legal systems, it is able to “see through” the referential opacity created by the quotation marks, in the same way as we as observers can see that Pierre means “in reality” the same city. . The Commission decreed that feta-de-re is sheep cheese, and that therefore labels like Feta and φετα must be used only to name sheep cheese. The regulation therefore turns out to be not one, but a number of norms, one for each language in the EU, decreeing that the respective translation of “Feta” must be used as label for sheep cheese only. This interpretation of the Directive as a bundle of norms, necessitated by the way we used quotation marks to analyse the norm, is indeed an accurate description of EU law making: EU norms are translated into the various different legal system of the member states, and can in that process subtly change meaning. The implementation of one and the same EU directive can and will vary to some extent between member states. But in all of these cases, the norm will be de-re, about Feta as an object, not the word or description “Feta”. We can achieve this result by moving the definition of Feta outside the “” bracket. Technically, that makes it necessary to quantify into the brackets, which allows us to express sentences such as : Peter thinks of Paul(de-re) under the description “my wife’s lover” that his shot will kill him, but he does not believe of Paul under the description “serving President” that his shot will kill him. Similarly, we can express the idea that it is illegal to sell something that is not Feta (de-re) under the expression “Feta”. To be able to express this formally, we need two different types of variables, de-re and de-dicto variables x^d and x^o . The first type behaves just like the variables of classical logic and takes only

objects-de-re as arguments, the other take linguistic entities as object. Consequently, a sentence of the form $\exists x^o P^o x^o$ is true if we can find a name which when substituted for the bound variable yields a true sentence. Similarly, each predicate carries a type $\tau = \bullet$ or o , indicating whether its argument position is interpreted referentially (\bullet) or substitutionally (o).

$$6) \forall^{\bullet} x (F^{\bullet} x \rightarrow SM^{\bullet} x \wedge \text{Ought}^o \text{“S}^o \text{ (“F”, } x) \leftrightarrow F^{\bullet} x \text{”}) \quad F: \text{Feta}; SM: \text{Sheep’s milk}$$

With 6 and 5, we now have a de-re and a de-dicto version of norms that prescribe the use of specific expressions. 6) in particular comes closest in our reconstruction to the logical form that represents the reasoning of the Commission when protecting as “Feta” cheese that , “in reality” or de-re, is Feta.

For the purpose of this paper, I will give only a very short version of the formal semantics, insofar it deviates from classical logic. It is a conservative extension of classical logic, inspired by Martin’s disquotational Logic and Blau’s three valued Logic of Quotations. [15]

As primitive symbols, we have the logical constants $\neg, \&, \forall, =$, (identity is of type \bullet), infinitely many de-re (object)variables x, y, z, z_1, \dots
 infinitely many de-dicto (expression) variables $x^o, y^o, z^o, z^o_1, \dots$
 infinitely many object parameter a, b, c, a_1, \dots
 and infinitely many predicate parameter for each type $t = t_1, \dots, t_r$ and $t_i = \bullet$ or o with “ \bullet ” marking de-re and “ o ” marking de-dicto argument positions

The *terms* of L in basic notation (BN) are the de-re and de-dicto variables, and the object parameters. The formulas of L in BN are either elemental or complex. Elemental formulas are

$$t_1 = t_2, P, X, P^{t_1, \dots, t_r} (e_1, \dots, e_r), X^{t_1, \dots, t_r} (e_1, \dots, e_r)$$

so that t_1, t_2 are terms in BN and for $I+ 1, \dots, r$:

- a) $t_i = \bullet \Rightarrow e_i$ is a term in BN
- b) $t_i = o \Rightarrow e_i$ is a (possibly defined) term without free de-re variables

Then $P e_1, \dots, e_r$ is called a formula which is free for exactly those variables where the e_1, \dots, e_r are free. (condition b expresses the idea that you can not quantify with de-re variables into de-dicto contexts)

The condition for complex formula of L is:

Let A, B, C be formulas in BN, v be a de-re or de-dicto variable. Then $\neg A, A \& B, \forall v C$ are formulas in BN (Sentences are as usual formulas without free variables)

We have the usual definition for $\exists, \rightarrow, \leftrightarrow, v$ and definite descriptions.

Semantics

A universe for L is an arbitrary set D . For each $d \in D$ we assume a fictitious object name $\underline{d} \in \underline{D}$ which is either an object parameter of L or a new symbol which is not element of L . \underline{D} be the set of these names. We get the formulas and terms of L_D from

the terms and formulas of L by substituting object names for one or more free object variables. (the additional names are not part of the language of L but of the interpreted language QDL ϕ . They are meta-theoretical tools to simplify the semantic rules for the quantifiers. Therefore, they do not appear in de-dicto positions, where the linguistic forms of the expressions *as used by a speaker* play a role) Expressions without object names are called pure expressions.

Unlike in standard interpretations, we do not interpret the predicate parameters, but the elemental de-re predicates.

Pe_1, \dots, e_r be a pure formula in basic notation, P predicate parameter of type t_1, \dots, t_r , so that for each $i = 1, \dots, r$:

$t_i = \bullet \Rightarrow e_i$ is the alphabetically first de-re variable different from e_1, \dots, e_{i-1}

$t_i = o \Rightarrow e_i$ is a pure description (parameter, definite description)

Let x_1, \dots, x_k be the alphabetically first free de-re variables in Pe_1, \dots, e_r ($0 \leq k \leq r$). Then we call this formula an *elemental predicate* in x_1, \dots, x_k and introduce the metatheoretical abbreviation E^k . From E^k we derive En_1, \dots, n_k by substituting terms n_i for x_i

An *interpretation* for L over D is a function f that fulfils the 4 conditions:

(I1) $f(\underline{d}) = d$ for all $d \in D$

(I2) $f(a) \in D$ for all parameters a

(I3) $f(E^0) \in \{W, F\}$ for elemental predicates E^0 of arity 0

(I4) $f(E^k) \subset U^k$ for each elemental predicate E^k of arity k

The truth-values of elemental sentences are determined by the following rules

(R=) $\{a_1 = a_2\}_f = \begin{matrix} t & \Leftrightarrow & f(a_1) = f(a_2) \text{ and both exist} \\ f & & \text{otherwise} \end{matrix}$

(R E^k) $\{Ea_1, \dots, a_k\}_f = \begin{matrix} t & \Leftrightarrow & \langle f(a_1), \dots, f(a_k) \rangle \in f(E^k) \\ f & & \text{otherwise} \end{matrix}$

From the rules for complex sentences, only the rules for quantification differ from the usual definitions, and are therefore given:

(R $\forall x$) $\{\forall x Ax\}_f = \begin{matrix} t & \Leftrightarrow & \{A\}_f = t \text{ for all } d \in D \\ f & & \text{otherwise} \end{matrix}$

(R $\forall x^0$) $\{\forall x^0 Ax^0\}_f = \begin{matrix} t & \Leftrightarrow & \{A\}_f = t \text{ for all pure descriptions (names and sentences)} \\ f & & \text{otherwise} \end{matrix}$

3. Conclusion

Assigning formal structures to European Union norms and cases provides a plethora of interesting challenges. Many of the preconditions for formal analysis that are taken for granted when analyzing the laws of individual nation states are absent, or in need of revision. This paper focused on but one of these aspects, the ability of EU norms to “talk about” the norms of member states in their respective languages. Not only that, while the present paper can only give a short first impression, the arguments of the parties and of the court in its reasoning replicate many of the themes found in the analytical philosophy of language. “Original baptism”, “dubbing” and experimental tests were all proposed, discussed and in varying degrees employed by the court, as was the debate between kind realists and conventionalists. We introduced a quotation logic that is loosely inspired by the formalisms of Martin and Blau, to be able to represent this reasoning without having to commit ourselves from the beginning to the theories that are under discussion by the court. Tentative conclusions that we can draw at this state include: Submission of parties and reasoning of the court can be understood as systematic shift in perspective, with the court being able to “see through” referential opacity and making de-re pronouncements. The most immediate effect of the court judgment is to “rigidify” the use of the term “Feta”, achieving through a norm what in logic is often achieved through “rationality postulates”. As a consequence, natural kinds, or even possibly artificial kind terms, are “made to” behave as if a realist theory of meaning that combines “original baptism” with a limited role for experimental tests were true.

References

- [1] BEIER, F., The need for protection of indications of source and appellations of origin in the common market. in: *Industrial Property* (35) 1977
- [2] DAVIS, A., Case report Denmark v. Commission. in: *Consumer Law Journal* (7) 1999 p. 243-244
- [3] DONELLAN, K., Reference and definite description. in: *Philosophical Review* 75 (1966)
- [4] DONELLAN, K., Kripke and Putnam on Natural Kind Terms. In: Ginet, C. (ed.) *Knowledge and Mind*. Oxford University Press, Oxford 1983
- [6] ECO, U., *Travels In Hyperreality*, Harvest Books, New York 1990
- [7] ELDER, C., Realism, Naturalism and culturally generated kinds. in: *Philosophical Quarterly* 39 (1989)
- [8] FUSTEGUERAS, A., On the semantics of natural kind words. in: *Theoria* (11) 1996 pp143-159
- [9] KNAAK, R., Case Law of the European Court of Justice on the protection of Geographical Indication and Designations of Origin Pursuant to the EC Regulation No 2081/92', in: *ICC* 32 (2001)
- [10] KRIPKE, S., *Naming and Necessity*. Basil Blackwell, Oxford 1980
- [11] QUINE, W.v.O.; Natural kinds. In W. V. Quine (ed.), *Ontological relativity and other essays* 114-138. Columbia University Press, New York.
- [12] PUTNAM, H., The meaning of “meaning”. in *ibid.* *Philosophical papers*. Cambridge University Press, Cambridge 1975
- [13] PUTNAM, H.: Natural Kinds and Human Artefacts. In: *Mind* (91) 1982
- [14] SCHWAB, B., ‘The Protection of Geographical Indications in the European Economic Community’, in: *E.I.P.R.* 17(1995).
- [15] BLAU, U: *Die Logik der Anführung und Quasianführung*. *Erkenntnis* 29 (1988), 227–268.

Visualising Legal Case-based Reasoning Argumentation Schemes

Adam WYNER¹, Trevor BENCH-CAPON

University of Liverpool

Abstract. The paper presents programs which together generate visual graphs for legal case based reasoning using argumentation schemes. Cases are represented using CATO factors, which appear in a factor hierarchy. For each case in a case base, the factors of the case are partitioned according to the role of the factor in arguing for or against a particular case decision. Inputting a case, a Prolog program generates the case factor partitions with respect to the input case and the cases in the case base. The Prolog program then instantiates the argumentation schemes with respect to the input case and partitioned cases in the case base. The output is sent to a graphics program written in C, which generates the output graphic as a tree in which nodes represent support or attack of a given proposition. Graphical representation of cases facilitates understanding of legal case-based reasoning.

Keywords. Legal case-based reasoning, implementation, argumentation schemes

Introduction

Legal case-based reasoning (LCBR) has long been a topic of interest in AI and Law ([2,1,5,7,8,6,3,4,11] among others), where a current undecided case is decided by comparing and contrasting factors in the current case against precedent cases in a case-base which have similar factors, then taking a decision in the precedent case as the decision into the current case. Several systems, notably CATO [1] which we focus on here, have been developed to teach and support reasoning about current cases with respect to a case base wherein the cases are analysed in terms of *factors*, which are prototypical fact patterns that are used to contribute to reasoning towards a decision in the case. Factors may predispose the decision in favour of the plaintiff or defendant in the case. Different precedents have different distributions of factors. Thus, finding and reasoning about precedents in arguing one's side in the current case is a complex task which requires one to examine the different combinations of factors in the precedent cases.

To find and reason with precedents, [3] introduced a case comparison method, wherein a case is analysed in terms of *partitions* of case factors. For example, if the current case and a given precedent case have exactly the same factors for the plaintiff and for the defendant (an *on point* case), and the decision in the precedent was for the plaintiff, then that precedent can be used to argue that the decision in the current case

¹Corresponding Author: Department of Computer Science, University of Liverpool, L69 3BX, UK.; E-mail: azwyner@liverpool.ac.uk.

ought to be for the plaintiff as well. Other distributions of factors in cases in the case-base can be used to support or undermine the plaintiff's argument that the current case should be decided in his favour. To support arguing about cases in a case base using partitions, [11] provided *argumentation schemes*, which are *defeasible reasoning patterns* and with reference to CATO factors and factor hierarchy [1]. For instance, while there might be a precedent case which contains all the factors of the current case and decided for the plaintiff, the precedent case may have additional factors in favour of the plaintiff. In such an instance, one could argue against using the precedent to support the decision in the current case, claiming that the additional factors in favour of the plaintiff are key in the decision in the precedent case and absent in the current case.

In this paper, we present our implemented tools which create the partitions, instantiate the argumentation schemes (by comparing a current case with respect to cases in the case-base), and visualise the resultant argument structure for a particular precedent. The visualisation is a graphical tree in which nodes represented the contribution of case factors in the decision, and relationships between nodes represent premises which imply a conclusion or attack of a given proposition. The visualisation makes more transparent the reasoning that goes into the case analysis and argument.

The paper is presented as follows. In section 1, we review the background of the proposal, including a discussion of CATO factors and factor hierarchy. In section 2, we outline the partitions and argumentation schemes. Section 3 discusses and exemplifies the implementation. We end with some discussion of issues and prospects for future development in section 4.

1. Background - Basic Elements of CATO

There are a variety of approaches to and implementations of LCBR, beginning with HYPO [2], and subsequently developing into CATO [1] and IBP [5] in one stream, and CABARET [7] and BankXX [8] in another. More theoretically oriented research appears in [6], [3], and [4]. Our implementation uses a CATO style representation of factors, which we discuss further below, where we outline the key elements and reasoning processes.

In this section, we present the basic elements of CATO, outlining first the factors then the factor hierarchy [1]. In CATO, cases are represented in terms of factors, stereotypical *fact patterns* that contribute to the decision in a case. Current cases, which are undecided, are compared with precedent cases in terms of common factors. The underlying idea is that a current case ought to be decided following the conservative legal reasoning principle of *stare decisis*. However, where cases are brought to court, there is usually some dispute concerning just what factors are relevant and which precedent decisions the current case decision ought to follow. Besides the comparison of the factors as they appear in a case, arguments can be made about factors which don't appear in the current case, but are nonetheless related by an *abstract factor* in a factor hierarchy; reasoning by comparison, given two factors which are children of an abstract parent factor, one in the current case and the other precedent, it can be argued that the differences between the cases are reconciled, and therefore, as it was decided in the precedent case so should it be decided in the current case.

In CATO, cases have factors and parties to cases; the factors are derived from trade secret law concerning trade secret misappropriation (Restatement of Torts First, Sec. 757

and the Uniform Trade Secret Act) (see [2,1]).² The factors associated with a case bias a decision in favour of one of the parties to the case. In Table 1, we represent *base-level* factors, which are those that are presumed (for our purposes) to be a way of characterising the relevant facts of the case. Table 1 gives factor names (giving an idea of what the factor is about), a factor ID (for reference), the party of the case which the factor favours (either defendant D or plaintiff P), along with the abstract factor which is the parent of the factor. For example, if a case has F1, we say that there is a factor which favours case being decided for D. The conceptual relation between the base-level factors and their parent is one of *support for* or *relevance of*; the presence of a base-level factor makes it more or less likely that a parent factor is supported or relevant to the case.

In general, a current case is decided for one party or another depending on the other factors of the case, the comparisons to factors in precedent cases (as partitions, discussed below), as well as the ways in which the current case factors counterbalance with the factors of a precedent case in relation to the parent factors. The argumentation schemes discussed later make explicit these reasoning patterns. The factors used here are a subset of the CATO factors. As it is not relevant to our current representation, other than the parent factors of base-level factors, we leave aside higher level factors (e.g. so-called intermediate legal concerns and legal issues).

Table 1. Base-level Factors: ID, Name, Side, Parent

ID	Name	Side	Parent
1	Disclosure-In-Negotiations	D	Efforts-To-Maintain-Secrecy-Vis-A-Vis-Defendant, Questionable-Means
6	Security-Measures	P	Efforts-To-Maintain-Secrecy
10	Secrets-Disclosed-Outsiders	D	Efforts-To-Maintain-Secrecy-Vis-A-Vis-Outsiders
12	Outsider-Disclosures-Restricted	P	Efforts-To-Maintain-Secrecy-Vis-A-Vis-Outsiders
15	Unique-Product	P	Info-Known, Info-Valuable
16	Info-Reverse-Engineerable	D	Info-Available-Elsewhere
18	Identical-Products	P	Info-Used
20	Info-Known-To-Competitors	D	Info-Known
21	Knew-Info-Confidential	P	Notice-Of-Confidentiality

2. Partitions and Argumentation Schemes

AS-CATO, proposed in [11] and further discussed in [10], justifies a decision concerning a current, undecided case on the basis of precedent cases using *argument schemes* (AS) which take into consideration *partitions* of factors between the cases as well as parent factors, which provide exceptions to arguments to *downplay* distinctions. We present the partitions, the argumentation schemes, and an example.

2.1. Partitions

AS-CATO reasons with respect to *case comparisons*, which are expressed as partitions of factors between the cases being compared. CC (Current Case) is a case in which

²[10] discusses issues concerning factor labels in the literature

Table 2. Partitions of Factors in CC and PCi

Partition	Biases Decision For	Set-Theoretic Definition
P1	P	P factors in both CC and PCi.
P2	P	D factors in both CC and PCi.
P3	P	P factors in CC not in PCi.
P4	P	D factors in PCi not in CC.
P5	D	D factors in CC not in PCi.
P6	P	P factors in PCi not in CC.
P7	U	Factors not in either CC or PCi.

the *outcome is undecided* and which we compare to a PC (Precedent Case) in which the *outcome is decided*. Our discussion will be in terms of finding an argument for the plaintiff: finding an argument for the defendant is similar, but the partitions are ordered differently. The fundamental question is whether, in comparing a CC to a precedent case (PCi being some arbitrarily chosen case), we should decide the CC for the same party as the decision in PCi. The partitions make the relevant case comparisons (in terms of the factors) explicit. In Table 2, we label the partitions P1,...,P7, indicate for whom the partition *biases* a decision, and describe the *set-theoretic definition* of the partition given CC and PCi.

Consider each of the partitions, and recall that we are considering cases in which P won in PCi.

- We say that P1 and P2 strengthen the case for P in CC because these are factors which appeared in PCi as well. PCi was a case in which P won. *Ceteris paribus*, as for PCi so for CC, so P should win in CC as well.
- P3 strengthens the case for P in CC since it says that there are additional P factors in CC than in PCi. Since P won in PCi without the additional P factors, then *a fortiori*, P ought to win in a CC which has the additional P factors.
- P4 strengthens the case for P in CC since it says that there are additional D factors in PCi that are not in CC. Since P won in PCi with the additional D factors, then *a fortiori*, P ought to win in a CC which as does not have those D factors.
- P5 weakens the case for P in CC because there are D factors in CC that are not in PCi. Perhaps if these factors had been present in PCi, P would not have won.
- P6 weakens the case for P in CC since there are P factors in PCi that are not in CC. P won in PCi, but it may have been these factors which are in PCi and not in CC that are key to the outcome in PCi.

As there is only rarely an exact match between a CC and a PCi (i.e. where P3-P6 are empty), we need to use our judgement, perhaps informed by other decisions, as to whether the differences are sufficient to affect the outcome.

To illustrate, suppose the factors in Table 1 and hypothetical cases described using the factors as in Table 3. Using these factors, cases, and case comparison method, we can provide selected case comparisons as in Table 4.

2.2. LCBR Argumentation Schemes

We provide six argumentation schemes which we use to argue about the outcome of a CC relative to a particular PCi. We broadly follow Walton [9] on argumentation schemes,

Table 3. Summary of Cases in Example

Case Name	P Factors	D Factors	Decision
CC	15, 21	10, 20	undefined
Mason	6, 15, 21	1, 16	P
Valco-Cincinnati	6, 12, 15, 21	1, 10	P
Televation	6, 12, 15, 18, 21	10, 16	P

Table 4. Selected Case Comparisons

CC/PCi	P1	P2	P3	P4	P5	P6
CC/Mason	15, 21	-	-	1, 16	10, 20	6
CC/Valco-Cincinnati	15, 21	10	-	1	20	6, 12
CC/Televation	15, 21	10	-	16	20	6, 12, 18

which are stereotypical, defeasible reasoning patterns comprised of a *conclusion* and *premises*. The particular statements constitute the scheme. For our purposes, we only consider that premises must hold and that exceptions must not hold for the conclusion to presumptively hold (See [11] for further discussion of other aspects of argumentation schemes not discussed here.)

The argumentation schemes are schematically represented as a cascade of related schemes which either support a premise or provide an argument for an exception, presented in Figure 1. All the leaf nodes of the tree are justified by inspection, which means that their content is determined by the case comparison; non-leaf nodes are justified with respect to nodes below them in the tree. The graphical language for expressing arguments is: arguments appear as circular nodes labelled *AS*; arguments are comprised of *premises*, represented with text boxes and pointed arrows pointing to the argument node, and a conclusion, represented with a text box that the argument node points to. Circular arrows indicate exceptions and point to argument nodes which are to be understood as *undercutters*, meaning that when the exception holds, the argument does not apply.

In **AS1**, the presumptive conclusion is that CC is decided for P since, while there are some factors for P and some for D in *both* cases (*P Factors Premise* and *D Factors Premise*), the decision in PCi (the precedent case) was decided in favour of P (*Factors Preference Premise*). The conclusion also presumes that the decision in PCi applies (*CC Weaker Exception*).

AS1: Main Scheme

- **P Factors Premise:** P1 are reasons for P.
- **D Factors Premise:** P2 are reasons for D.
- **Factors Preference Premise:** P1 was preferred to P2 in PCi.
- **CC Weaker Exception:** The priority in PCi does not decide CC.
- **Conclusion:** Decide CC for P.

P Factors Premise and *D Factors Premise* are leaf nodes which need no further justification other than holding in the case comparison. The *Factors Preference Premise* and *CC Weaker Exception* need justification to hold, which is done with two argumentation schemes.

AS2 justifies the *Factors Preference Premise*. Again, *P Factors Premise* and *D Factors Premise* must hold (since these are taken directly from partitions in the comparison

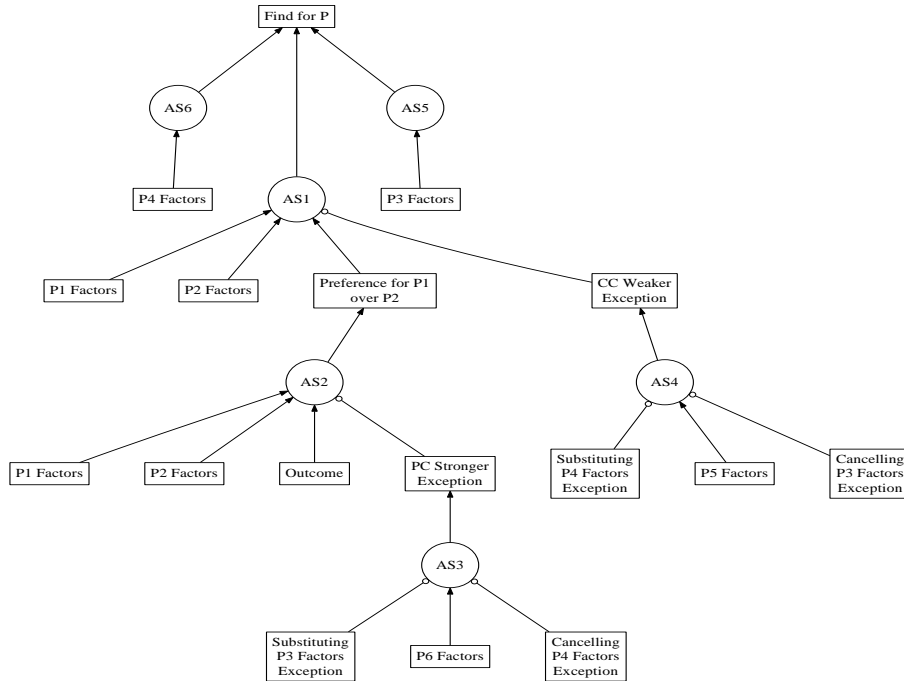


Figure 1. Visualisation of Argumentation Schemes for Legal Case Based Reasoning

of CC and PCi). The *Outcome* of the PCi must be for P, which can be determined by inspection of the case. Furthermore, the *PCi Stronger Exception* must not hold, which is a statement which will need further justification.

AS2: Preference-From-Precedent Scheme

- **P Factors Premise:** P1 are reasons for P.
- **D Factors Premise:** P2 are reasons for D.
- **Outcome Premise:** PCi was decided for P.
- **PCi Stronger Exception:** PCi cannot be used for P in CC.
- **Conclusion:** P1 was preferred to P2 in PCi.

To justify *PCi Stronger Exception*, the claim that PCi cannot be used for P in CC, we must *distinguish* PCi from CC in a way that the conclusion of AS2 does not follow. We see this in AS3, where if partition P6 holds, namely there are factors for P in PCi that do not appear in CC (*P6 Factors Premise*), then we can presume the exception in AS2 does hold, which blocks the conclusion of AS2. F_x , F_y , F_z are factor variables.

AS3: Precedent-Stronger Scheme

- **P6 Factors Premise:** $F_x \in P6$
- **Substituting P3 Factors Exception:** $F_y \in P3$ with the same parent as F_x .
- **Cancelling P4 Factors Exception:** $F_z \in P4$ with the same parent as F_x .
- **Conclusion:** PCi cannot be used for P in CC.

However, even though *P6 Factors Premise* may hold, there may be other reasons that the conclusion of this scheme, *PCi cannot be used for P in CC*, does not hold, which is given by two exceptions. These exceptions *downplay* the distinction suggested by *P6 Factors Premise*; essentially they claim that though there may be factors for P in PCi that do not appear in CC, the differences can be neutralised. For *Substituting P3 Factors Exception*, we compare a factor F_x for P in CC not found in PCi with a factor F_y for P in PCi not found in CC; if F_x and F_y have the same parent factor F which favours P, then we can understand that F_x strengthens the case for P in CC in the same way as F_y strengthens the case for P in PCi, thus lessening or eliminating the distinction. This supports the claim that the decision reached PCi applies as well to CC, despite the apparent difference between them. Where such common parents cannot be found, then the distinction stands. Somewhat similar reasoning applies to *Cancelling P4 Factors Exception* with respect to factors for D, where we have factors relating both positively and negatively to the same abstract factor.

An alternative means to defeat the presumptive conclusion of **AS1** is to argue for the exception that CC is weaker than PCi. This is given by **AS4**, where the conclusion presumptively follows where CC has *additional* factors for D than PCi.

AS4: Current-Case-Weaker Scheme

- **P5 Factors Premise:** $F_x \in P5$.
- **Substituting P4 Factors Exception:** $F_y \in P4$ with the same parent as F_x .
- **Cancelling P3 Factors Exception:** $F_z \in P3$ with the same parent as F_x .
- **Conclusion:** The priority in PCi does not decide CC.

As with **AS3**, the presumptive conclusion of **AS4** only holds where the exceptions do not hold. The exceptions hold, in turn, only where the distinctions between CC and PCi cannot be *downplayed*, reasoning much as with downplaying in **AS3**.

Finally, we have two argument schemes which effectively strengthen the conclusion for P in the CC. Where **AS5** is instantiated, we have more factors in the CC favour of P than in the PCi, thus strengthening decision in the case; similarly, **AS6** highlights that there are more D factors in the PCi than there are in the CC, also strengthening the decision.

AS5: Current-Case-Stronger Scheme

- **CC P Factors Richer:** P3 are reasons for P.
- **Conclusion:** Decide CC for P.

AS6: Precedent-Case-Weaker Scheme

- **PCi D Factors Richer:** P4 are reasons for P.
- **Conclusion:** Decide CC for P.

To this point, we have discussed the representation of cases in CATO using factors and a factor hierarchy, the analysis of cases in terms of partitions, and the use of argumentation schemes to argue about cases in relation to a case base and partitions. In the next section, we discuss our implementation of AS-CATO and provide sample visualisations of arguments.

3. The Implementation

The current version of our implementation has the following components:

- `lcbPartitions`: a Prolog program. Given a current case and a case base, where cases are expressed in terms of factors, it outputs a database of partitions of cases relative to the current case.
- `forGraph`: a Prolog program. Given a current case and a database of case partitions, it instantiates the argumentation schemes. The program can be used to produce results either for the plaintiff or the defendant.
- `genDot`: a C program which takes the instantiated argumentation schemes and generates the graph.
- `genGraph`: a shell script to run the `lcbPartitions`, `forGraph`, and `genDot` programs.

As the *forGraph* program is the most significant of these, we describe it in a bit more detail. As before, we only consider arguments relative to the plaintiff; we wish to report precedents which can be used or not. The *forGraph* program populates the portions of the arguments only and does not reason with the graph. The textual output below shows how the graph is interpreted to reason to an outcome.

- `go(CurrentCase)`: a procedure which finds precedents which can be used or not to argue for the plaintiff. The subgoals are to `getArguments` and `reportGraph`.
- `getArguments`: a procedure which has several subgoals. `caseComparison` instantiates variables relative to a case comparison database. It also has subgoals which evaluate to `true` for each of the first four main schemes, *AS1-AS4*, holds. These are *winner* if the CC wins, *preferred* if the precedent supports the preference of P1 over P2, *stronger* if the precedent stronger than the current case, and *weaker* if CC is weaker than the current case.
- `winner` and `preferred`: procedures which are `true` where P1 and P2 are non-empty.
- `stronger`: a procedure which is `true` where P6 is not empty and P factors in P3 or P4 are downplayed with respect to P6
- `weaker`: a procedure which is `true` where P5 is not empty and P factors in P3 or P4 are downplayed with respect to P5.
- `downplay`: a procedure which is `true` where the factors in the respective partitions have the same parent.
- `parent`: a functor which is given as a list of pairings of base level factors with their parent factor.

The result of `getArguments` is passed to `reportGraph`, which writes the results of `getArguments` in a form which can be used by `genDot`. `genDot` generates a dot file, which is input to the `graphViz` program, generating the visualisation. For example, Figure 2 is the result of testing whether Valco serves as a good precedent for the current case.

The output from `reportGraph` can also be used to populate a textual template, resulting in a correlated verbal output:

We should not find for the plaintiff in the Current Case on the basis of Valco. Although f15 and f21 are reasons favouring the plaintiff, and f10 is a reason favouring the defendant, and Valco was found for the plaintiff, f20 is present in CC and strengthens the case for the defendant.

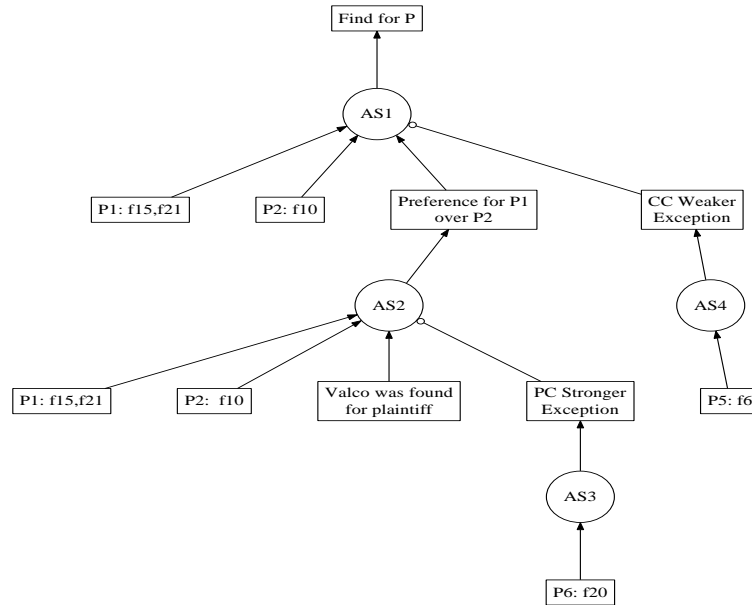


Figure 2. Visualisation of CC v. Valco

4. Discussion

In this paper, we have shown how we can process and visualise arguments about a decision for a current case relative to a case-base of precedents. In addition to the visualisation, we can provide the output as a textual representation, generated from the same file which is used to produce the visualisation.

In future work, we look to augment some of the user interface to allow, for instance, users to dynamically select which cases used for case comparisons. Similar to the claim lattice of HYPO and CATO, we want to generate and work with a case lattice, which is a hierarchy of case comparisons; such a graphical representation would display multiple, alternative cases. It would be useful to be able to graphically represent and make interactive the factor hierarchy so users can see the role of the hierarchy in reasoning about the cases. Similarly, we could have an interactive facility for adding or removing factors from cases and relative to the factor hierarchy in order to test 'hypotheticals' in a systematic, graphical manner. We will examine how to combine cases in into a single, larger visualisation which can be browsed dynamically. A main area of interest is to develop argumentation schemes and visualisations for dimensions as the current approach only works with factors. In the current implementation, the user must choose the case for comparison; we should like to add some rules to reason over the results of case comparisons, for example to find the most on point case or the case which best attacks the opposing position. Finally, we will examine how to integrate issues as in IBP.

Acknowledgements

During the writing of this paper, the first author was supported by the IMPACT Project (European Framework 7 project No. 247228). The authors wish to thank Kenneth Chan of the Department of Computer Science, University of Liverpool, for his help in writing the program to output the visualisation of the argumentation schemes.

References

- [1] Vincent Aleven. *Teaching case-based argumentation through a model and examples*. PhD thesis, University of Pittsburgh, 1997.
- [2] Kevin Ashley. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Bradford Books/MIT Press, Cambridge, MA, 1990.
- [3] Trevor Bench-Capon. Arguing with cases. In A. Oskamp et al., editors, *JURIX 1997*, pages 85–100, Nijmegen, 1997. Gerard Noodt Instituut.
- [4] Trevor Bench-Capon and Giovanni Sartor. A model of legal reasoning with cases incorporating theories and values. *Artificial Intelligence*, 150(1-2):97–143, 2003.
- [5] Stefanie Brüninghaus and Kevin D. Ashley. Predicting the outcome of case-based legal arguments. In Giovanni Sartor, editor, *ICAIL'03: Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pages 233–242, Edinburgh, United Kingdom, 2003. ACM Press: New York, NY.
- [6] Henry Prakken and Giovanni Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4(3-4):331–368, 1996.
- [7] Edwina L. Rissland and David B. Skalak. CABARET: rule interpretation in a hybrid architecture. *International Journal of Man-Machine Studies*, 34(6):839–887, 1991.
- [8] Edwina L. Rissland, David B. Skalak, and M. Timur Friedman. BankXX: Supporting legal arguments through heuristic retrieval. *Artificial Intelligence and Law*, 4(1):1–71, 1996.
- [9] Douglas Walton. *Argumentation Schemes for Presumptive Reasoning*. Erlbaum, Mahwah, N.J., 1996.
- [10] Adam Wyner. An ontology in OWL for legal case-based reasoning. In Katie Atkinson, editor, *Proceedings of the JURIX 2007 Workshop on Modelling Legal Cases*, Leiden, The Netherlands, December 2007.
- [11] Adam Wyner and Trevor Bench-Capon. Argument schemes for legal case-based reasoning. In Arno R. Lodder and Laurens Mommers, editors, *Legal Knowledge and Information Systems. JURIX 2007*, pages 139–149, Amsterdam, 2007. IOS Press.