Implementing Factors with Magnitude

Trevor BENCH-CAPON [a] [1], Katie ATKINSON [a]
[a] *Department of Computer Science, University of Liverpool, UK*

Much work on arguing with legal cases has represented cases using *factors*, stereotypical fact patterns which are legally significant and which can be seen in Boolean terms, as either present or absent [5]. Factors were introduced in the CATO program [3], which supported argumentation in US Trade Secret law. In section 757 of the *Restatement of Torts*, on which CATO is based, however, we read:

> Some factors to be considered in determining whether given information is one's trade secret are: the *extent* to which the information is known outside of his business; the *extent* to which it is known by employees and others involved in his business; the *extent* of measures taken by him to guard the secrecy of the information; the *value* of the information to him and to his competitors; the *amount* of effort or money expended by him in developing the information; and the *ease or difficulty* with which the information could be properly acquired or duplicated by others. (http://www.lrdc.pitt.edu/ashley/RESTATEM.HTM *italics ours*)

From this it is clear that concepts which are treated as factors in CATO, are not really Boolean but have *magnitudes* (extents, amounts, degrees of difficulty and the like). Consequently current work in AI and Law attempts to extend the modelling of case law to accommodate factors with magnitude ([8], [9] and [6]). While [8] and [9] provide formal accounts, the demonstrated program draws on [6] to provide a practical illustration. It does so by re-implementing the CATO analysis of [3], but using factors with magnitude.

The knowledge is represented following the ANGELIC methodology [2], specifically using the 2-regular structure of [1], in which statements (issues, intermediate concepts and factors) are represented as nodes and non leaf nodes have exactly two children. Each non-leaf node is associated with acceptance conditions, expressed in terms of its two children. Like [6] the statements have *degrees of acceptance*, in range [0,1]: 0 representing total rejection and 1 total acceptance. Some statements are genuinely Boolean and they use only 0 and 1. The acceptance conditions are then a set of conditions for attributing particular degrees of acceptance, together with a default value. Of the 26 factors in [3], 17 turned out to be adequately modelled as Boolean, but 9 were modelled with magnitudes. The program was implemented using SWI Prolog (Windows version).

Acceptance conditions can take a variety of forms, depending on whether the children have magnitude, and how they combine to determine the degree of acceptability of the node. The program uses the following acceptance condition types:

- Boolean Disjunctions (13 nodes) and Boolean Conjunctions (4 nodes).
- Fuzzy Disjunctions (7 nodes), where one or both children have magnitude.
- Preference Based Exceptions (7 nodes). Preferences are based on values, taken from [7]. For example, information is considered valuable if it is useful in developing the product, *unless* it is known to others or available elsewhere. This ex-

presses a preference for the value of holding the plaintiff responsible for protecting the secret over the value of its material worth.

- Defeasibility Based Exceptions (2 nodes). The exceptions are based on the meaning of terms: thus secrecy was not maintained with respect to the defendant if the information was disclosed in negotiations *unless* the defendant had agreed not to disclose the information.
- Comparison with a threshold (3 nodes): this exploits the magnitude of a factor: the threshold determining whether the magnitude is *sufficient* for acceptance. The threshold may vary according to the importance of the factor and its related value.
- Balancing two factors using weights (1 node). The nodes in question in our program are *questionable means* and *legitimately obtained*, and the balance needs to be struck according to our view of the weights of the values promoted by these factors. Such arguments are discussed in [4] and [6].

Fuzzy Conjunctions are possible, but no instances were needed. Thresholds and weights are specified in separate predicates, to allow different thresholds and different weights to be used to express different preferences. The program is completed by supplying degrees of acceptance for the base level factors, according to the facts of particular cases.

The program was tested using the 32 publicly available cases from CATO collected in [7] and used in [2]. In [2] several cases exhibited problems which require questioning the original analysis of those cases, or modifying the representation. The use of factors with magnitude can address these problems. In the case of *reverse engineerability*, which was problematic in [2], we can now reflect what the *Restatement* calls the "ease or difficulty with which the information could be properly acquired or duplicated", and set the threshold higher or lower as required by decisions in these cases. Further, the balance between factors can be affected by the choice of weights, and so be an explicit part of the program rather than implicit in the analysis. Importantly this enables us to test the effect on other cases of using weights to strike the balance so as to include or exclude a given case. The program is a step in our current exploration of the use of factors with magnitude in practical applications.

References

[1] L. Al-Abdulkarim, K. Atkinson, and T. Bench-Capon. Factors, issues and values: Revisiting reasoning with cases. In *Proceedings of the 15th International Conference on AI and Law*, pages 3–12. ACM, 2015.

[2] L. Al-Abdulkarim, K. Atkinson, and T. Bench-Capon. A methodology for designing systems to reason with legal cases using abstract dialectical frameworks. *Artificial Intelligence and Law*, 24(1):1–49, 2016.

[3] V. Aleven. *Teaching case-based argumentation through a model and examples*. U. of Pittsburgh, 1997.

[4] T. Bench-Capon. Arguing with dimensions in legal cases. In *Proceedings of the 17th Workshop on Computational Models of Natural Argument*, pages 2–6, 2017.

[5] T. Bench-Capon. HYPO's Legacy: Introduction to the virtual special issue. *Artificial Intelligence and Law*, 25(2):205–250, 2017.

[6] T. Bench-Capon and K. Atkinson. Dimensions and values for legal CBR. *Proceedings of JURIX 2017*, pages 27–32, 2017.

[7] A. Chorley and T. Bench-Capon. An empirical investigation of reasoning with legal cases through theory construction and application. *Artificial Intelligence and Law*, 13(3-4):323–371, 2005.

[8] J. Horty. Reasoning with dimensions and magnitudes. *Proceedings of the 16th International Conference on Artificial Intelligence and Law*, pages 109–118, 2017.

[9] A. Rigoni. Representing dimensions within the reason model of precedent. *Artificial Intelligence and Law*, 26(1):1–22, 2018.