Checking the validity of rule-based arguments grounded in cases: a computational approach

Heng ZHENG [a,1], Minghui XIONG [b] and Bart VERHEIJ [a]
[a] *Artificial Intelligence, University of Groningen, The Netherlands*
[b] *Institute of Logic and Cognition, Sun Yat-sen University, Guangzhou, China*

Abstract. One puzzle studied in AI & Law is how arguments, rules and cases are formally connected. Recently a formal theory was proposed formalizing how the validity of arguments based on rules can be grounded in cases. Three kinds of argument validity were distinguished: coherence, presumptive validity and conclusiveness. In this paper the theory is implemented in a Prolog program, used to evaluate a previously developed model of Dutch tort law. We also test the theory and its implementation with a new case study modeling Chinese copyright infringement law. In this way we illustrate that by the use of the implementation the process of modeling becomes more efficient and less error-prone.

Keywords. Artificial Intelligence and Law, Rule-based Reasoning, Case-based Reasoning, Argumentation Modeling, Prolog

1. Introduction

The recent case model formalism [1] is a hybrid theory showing connections between cases, rules and arguments [2]. The formalism defines different ways in which rule-based arguments can be valid in cases: arguments can be coherent, conclusive or presumptive. The formalism has been applied to model Dutch tort law, showing how a rule-based legal domain can be grounded in legal cases. In this way, a formal connection is established between the civil law tradition focusing on rules and the common law tradition focusing on cases.

The present paper provides a computational version of the case model formalism. A Prolog program is presented that can computationally check whether a case model is correct, whether rule-based arguments are valid (in the three kinds of validity coherence, conclusiveness and presumptiveness), and whether defeating circumstances are rebutting, undercutting or undermining. The computational tool can be used to support the manual modeling of a complex legal domain, making that more manageable. As an example, we provide a new domain model, namely Chinese copyright infringement law, both formally (as a case model) and computationally (in Prolog).

[1] Corresponding Author: zhengh48@mail2.sysu.edu.cn.

```
case(model_num(1),case_num(101),[not(dmg)]).
...
case(model_num(1),case_num(104),[not(dut),dmg,unl,imp,not(cau)]).
case(model_num(1),case_num(105),[dut,dmg,unl,imp,cau,vrt,not(vst),not(vun),ift
    ,not(ila),not(ico),not(jus),prp]).
...
case(model_num(1),case_num(114),[not(dut),dmg,not(unl),vrt,not(vst),jus]).
...
case_order(model_num(1),[case_num(101),case_num(102),case_num(103),case_num
    (104),[case_num(105),case_num(106),...,case_num(113)],...]).
```

Listing 1: Definition of the Dutch tort law case model in Prolog

2. The implementation of case model formalism in Prolog with case studies

We have implemented the case model formalism in Prolog. We use the previously developed model of Dutch tort law [2] as an illustration. Cases are represented as Prolog lists, of which the elements consist of strings and their negations (represented using not/1). Case models are represented as lists of cases and their ordering, where case models and cases are referred to using identifiers.

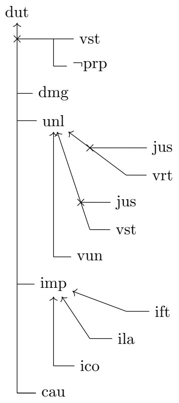
Listing 1 provides a part of the representation of the case model for Dutch tort law (model_num(1)). The full model consists of the 16 cases discussed in [2]. The ordering is represented as a list of cases and lists of cases, each representing an equivalence class of the total preorder, in decreasing level of preference. Hence the first element of this list represents the case or cases that are maximal in the total preorder. Here there is one maximal case case_num(101): not(dmg), representing that there are no damages. When an equivalence class consists of several cases, such as [case_num(105),...,case_num(113)], it is represented as a list of cases (actually: of case identifiers).

The predicate case_model_valid/1 (with a case identifier as single argument) checks whether all cases are consistent, incompatible and different. Checking whether the ordering of cases is total and transitive is not directly represented since we use an explicit representation of the total preorder as a list of equivalence classes. In this representation another validity check is helpful, namely whether each case of which an identifier appears in the ordering is defined and whether each case identifier of a defined case appears in the ordering exactly once. This check has been implemented using the predicate ordering_valid/1.

Arguments are represented by a list of premises and a list of conclusions. The three kinds of validity of arguments can be checked using the predicates coherent/1, conclusive/1 and presumptively_valid/1, each taking an argument (represented using argument/2) as input. Coherence is checked by first determining the case made by an argument, found by appending the list of premises to the list of conclusions, and then checking whether there is a case in the case model that contains all elements of the case made by the argument.

The conclusiveness of an argument is checked by first checking whether the argument is coherent and then checking, if all cases in the case model that contain the argument's premises also contain its conclusions.

An argument's presumptive validity is checked by determining a maximally preferred case that witnesses the argument's coherence (i.e., finding a case in



```
presumptively_valid([dmg,unl,imp,cau],[dut]).
    rebutting_attack(argument([dmg,unl,imp,cau],[dut]),[vst,not(prp)]).


    rebutting_attack(argument([vrt],[unl]),[jus]).
presumptively_valid(argument([vrt],[unl])).
    rebutting_attack(argument([vst],[unl]),[jus]).
presumptively_valid(argument([vst],[unl])).
conclusive(argument([vun],[unl])).

conclusive(argument([ift],[imp])).
conclusive(argument([ila],[imp])).
conclusive(argument([ico],[imp])).
```

Figure 1. The Dutch tort law model: argument structure (left); in Prolog (right)

which both the premises and conclusions of the argument hold and that is maximal in the ordering with this property) and then checking for each case in which the argument's premises hold whether that case is of equal or lower ordering.

Attack of arguments has been implemented using the predicates successful_attack/2, rebutting_attack/2, undercutting_attack/2 and undermining_attack/2. The predicate successful_attack/2 takes an argument and defeating circumstances (as a list) as input. The predicate checks whether the argument is presumptively valid and whether the argument to the same conclusions but from the premises with the defeating circumstances appended is not presumptively valid.

The three kinds of attack—rebutting, undercutting and undermining—are defined in terms of successful attack as follows. Rebutting attack requires a successful attack of an argument such that also the argument from the premises to the opposite of the argument's conclusion is presumptively valid. Note that this only makes sense for arguments with a single element as conclusion as we use no Prolog expression for the negation of a series of conclusions. Successful attacks that are not rebutting attacks are undercutting. Undermining attack is a special kind of successful attack, namely an attack of an argument with a tautology as premise. In the program, a tautology is represented as an empty Prolog list [ ].

The Prolog program can be used to validate hand-made domain models, such as the case model for Dutch tort law of [2]. In Figure 1, we show the argument structure that is valid in the hand-made formal case model of that paper (left). On the right, we show Prolog clauses that all evaluate to true given the Prolog version of that case model (partially shown in Listing 1). In other words, the model is computationally validated.

Another case study is about Chinese copyright infringement. The article of Copyright Infringement in Chinese Criminal Law [3] is below:

> **Article 217** Whoever, for the purpose of making profits, commits any of the following acts of infringement on copyright shall, if the amount of illegal gains is relatively large, or if there are other serious circumstances, be sentenced to fixed-term imprisonment of not more than three years or criminal detention and shall also, or shall only, be fined; if the amount of illegal gains is huge or if there are other especially serious circumstances, he shall be

> sentenced to fixed-term imprisonment of not less than three years but not more than seven years and shall also be fined:
> (1) reproducing and distributing a written work, musical work, motion picture, television programme or other visual works, computer software or other works without permission of the copyright owner;
> (2) publishing a book of which the exclusive right of publication is enjoyed by another person;
> (3) reproducing and distributing an audio or video recording produced by another person without permission of the producer; or
> (4) producing or selling a work of fine art with forged signature of another painter.

According to the articles related to Art. 217 in Chinese criminal law [3], and relevant official judicial interpretations [4], there is a defeating circumstance: the action was not belong to “without permission of the copyright owner”.

In the light of Art. 217 and the judicial interpretations related to it, a case model based on Verheij's case model formalism with a similar modeling approach to the Dutch tort law model in [2] can be built. We use the elementary propositions in Table 1, shown with their formal abbreviations. The full model has 30 cases. A selection of the cases is shown in its case list version in Table 2. The model has identifier model_num(2). In the text below, cases are numbered 1, 2, 3, ... corresponding to cases 201, 202, 203, ... in the Prolog version.

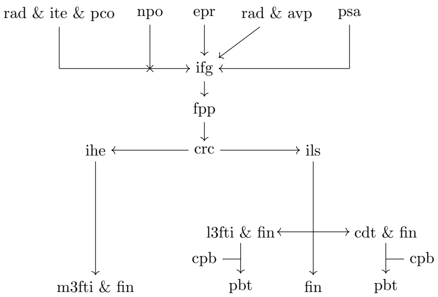
From Chinese copyright infringement, we can analyze the argument structure as in the diagram in Figure 2 (left). This argument structure shows multiple

Table 1. Elementary propositions in the copyright infringement model with abbreviations

ifg	there is a copyright infringement
fpp	the act was for the purpose of making profits
rad	the act was reproducing and distributing something
ite	the act concerned the items in Art. 217:1
pco	the act was without permission of the copyright owner
npo	the act was not belong to ”without permission of the copyright owner”
epr	a book is published of which the exclusive right of publication is enjoyed by another person
avp	a audio or video recording is produced by another person without permission of the producer
psa	a work of fine art with forged signature of another painter is produced or sold
ils	the amount of illegal gains is large or other serious circumstances
ihe	the amount of illegal gains is huge or other especially serious circumstances
crc	the person commits the crime of copyright infringement
l3fti	the person shall be sentenced to fixed-term imprisonment of at most three years
cdt	the person shall be sentenced to criminal detention
fin	the person shall be fined
m3fti	the person shall be sentenced to fixed-term imprisonment of not less than three years but not more than seven years
cpb	the defendant satisfies the conditions of probation
pbt	the defendant will be put on probation

Table 2. The Chinese copyright infringement case model (selection)

Case 1	¬rad,¬ite,¬pco,¬epr,¬avp,¬psa,¬ifg
Case 2	rad,ite,pco,npo,¬ifg
Case 3	rad,ite,pco,¬npo,¬epr,¬avp,¬psa,ifg,¬fpp
Case 4	rad,ite,pco,¬epr,¬avp,¬psa,ifg,fpp,ihe,¬ils,crc,m3fti,¬l3fti,¬cdt,fin
Case 5	rad,ite,pco,¬epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,¬cdt,fin
Case 6	rad,ite,pco,¬epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,cpb,pbt
Case 7	rad,ite,pco,¬epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,cpb,pbt
Case 8	rad,ite,pco,¬epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,¬cpb,¬pbt
Case 9	rad,ite,pco,¬epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,¬cpb,¬pbt
...	...
Case order	Case 1 > Case 2 = Case 3 > Case 4 = Case 5 = Case 6 = Case 7 = Case 8 = Case 9



```
presumptively_valid(argument([rad,ite,pco],[ifg])).
presumptively_valid(argument([crc],[ils])).
presumptively_valid(argument([crc,ils],[l3fti,fin])).

rebutting_attack(argument([rad,ite,pco],[ifg]),[npo]).

conclusive(argument([crc,ihe],[m3fti,fin])).
conclusive(argument([ifg,fpp],[crc])).
conclusive(argument([crc,ils,l3fti,fin,cpb],[pbt])).
conclusive(argument([crc,ils,cdt,fin,cpb],[pbt])).
```

Figure 2. The Chinese copyright infringement model: argument structure (left); in Prolog (right)

rule-based steps and an exception-based attack. The structure is valid in the case model we built. Following the definitions of the case model formalism, the arguments in Figure 2 (right) can be extracted in the model. The Prolog program confirms the validity of these arguments. These Prolog queries are evaluated as true, which means the results of the Prolog program correspond to our analysis of the Chinese copyright infringement model.

3. Conclusion

The results of this paper show that an implementation of the case model formalism can be used to support the modeling of a legal domain with a complex argument structure involving combined support and attack[2]. In this way, we have shown a computational connection between cases, rules and arguments, applied to the civil law system of the Netherlands and to criminal law in the Chinese legal system. AI and legal reasoning technology needs to combine rule-based reasoning, case-based reasoning and argumentation together, paving the way for argumentation technology that bridges cases and rules, as it is common in the law.

References

[1] B. Verheij. Correct Grounded Reasoning with Presumptive Arguments. In L. Michael and A. Kakas, editors, *15th European Conference on Logics in Artificial Intelligence, JELIA 2016. Larnaca, Cyprus, November 9–11, 2016. Proceedings (LNAI 10021)*. Springer, Berlin, 2016.

[2] B. Verheij. Formalizing Arguments, Rules and Cases. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL 2017)*, 2017.

[3] The National People's Congress of the People's Republic of China. Criminal Law of the People's Republic of China. http://www.npc.gov.cn/englishnpc/Law/2007-12/13/content_1384075.htm, 2011.

[4] The Supreme People's Court of The People's Republic of China, The Supreme People's Procuratorate of the People's Republic of China, and The Ministry of Public Security of the People's Republic of China. Judicial interpretation on the application of law in handling criminal cases involving infringement of intellectual property rights (in Chinese). http://www.court.gov.cn/fabu-xiangqing-2903.html, 2011.

[2] The full program code and the Chinese copyright infringement model are available at https://github.com/Zhe333/Appendix.git