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

Heng ZHENG, Minghui XIONG, Bart VERHEIJ

Prolog Program Code

The following predicates will be used as basic predicates [1, 2, 3, 4] in the program of case model implementation:

contradict(X,Y): this predicate checks if X is not(Y), or Y is not(X).

contradict_element_list(X, List): this predicate checks if X is contradicted with all elements in List.

contradict_list_list(List1, List2): this predicate checks if all elements in List1 are contradicted with all elements in List2.

length_list(List, Num): Num is the number of the elements in List.

negation_conclusion(Conclusion_list, Neg_Conclusion_list): this predicate will turn the element in conclusion list of an argument into its negation form.

remove_sublists(List1, List2): List2 is a list without any sublist and contains all elements in List1.

cases_without_preference(model_num(N),Cases_list): Cases_list is the set of all cases in model N without preferred relation.

member_list(List1,List2): this predicate checks if all elements in List1 are in List2.

```
%case model
case_model(model_num(N)) :-
          case(model_num(N),case_num(_),_),
          case_order(model_num(N),_),!.

case(model_num(2),case_num(201),[not(rad),not(ite),not(pco),not(epr),not(avp),not(
    psa),not(ifg)]).
case(model_num(2),case_num(202),[rad,ite,pco,npo,not(ifg)]).
case(model_num(2),case_num(203),[rad,ite,pco,not(npo),not(epr),not(avp),not(psa),ifg
    ,not(fpp)]).
case(model_num(2),case_num(204),[rad,ite,pco,not(epr),not(avp),not(psa),ifg,fpp,ihe,
    not(ils),crc,m3fti,not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(205),[rad,ite,pco,not(epr),not(avp),not(psa),ifg,fpp,not(
    ihe),ils,crc,not(m3fti),not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(206),[rad,ite,pco,not(epr),not(avp),not(psa),ifg,fpp,not(
    ihe),ils,crc,not(m3fti),l3fti,not(cdt),fin,cpb,pbt]).
case(model_num(2),case_num(207),[rad,ite,pco,not(epr),not(avp),not(psa),ifg,fpp,not(
    ihe),ils,crc,not(m3fti),not(l3fti),cdt,fin,cpb,pbt]).
case(model_num(2),case_num(208),[rad,ite,pco,not(epr),not(avp),not(psa),ifg,fpp,not(
    ihe),ils,crc,not(m3fti),l3fti,not(cdt),fin,not(cpb),not(pbt)]).
case(model_num(2),case_num(209),[rad,ite,pco,not(epr),not(avp),not(psa),ifg,fpp,not(
    ihe),ils,crc,not(m3fti),not(l3fti),cdt,fin,not(cpb),not(pbt)]).
case(model_num(2),case_num(210),[epr,not(pco),not(avp),not(psa),ifg,not(fpp)]).
case(model_num(2),case_num(211),[epr,not(pco),not(avp),not(psa),ifg,fpp,ihe,not(ils)
    ,crc,m3fti,not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(212),[epr,not(pco),not(avp),not(psa),ifg,fpp,not(ihe),ils
    ,crc,not(m3fti),not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(213),[epr,not(pco),not(avp),not(psa),ifg,fpp,not(ihe),ils
    ,crc,not(m3fti),l3fti,not(cdt),fin,cpb,pbt]).
case(model_num(2),case_num(214),[epr,not(pco),not(avp),not(psa),ifg,fpp,not(ihe),ils
    ,crc,not(m3fti),not(l3fti),cdt,fin,cpb,pbt]).
case(model_num(2),case_num(215),[epr,not(pco),not(avp),not(psa),ifg,fpp,not(ihe),ils
    ,crc,not(m3fti),l3fti,not(cdt),fin,not(cpb),not(pbt)]).
```

```
case(model_num(2),case_num(216),[epr,not(pco),not(avp),not(psa),ifg,fpp,not(ihe),ils
      ,crc,not(m3fti),not(l3fti),cdt,fin,not(cpb),not(pbt)]).
case(model_num(2),case_num(217),[rad,avp,not(pco),not(epr),not(psa),ifg,not(fpp)]).
case(model_num(2),case_num(218),[rad,avp,not(pco),not(epr),not(psa),ifg,fpp,ihe,not(
      ils),crc,m3fti,not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(219),[rad,avp,not(pco),not(epr),not(psa),ifg,fpp,not(ihe)
      ,ils,crc,not(m3fti),not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(220),[rad,avp,not(pco),not(epr),not(psa),ifg,fpp,not(ihe)
      ,ils,crc,not(m3fti),l3fti,not(cdt),fin,cpb,pbt]).
case(model_num(2),case_num(221),[rad,avp,not(pco),not(epr),not(psa),ifg,fpp,not(ihe)
      ,ils,crc,not(m3fti),not(l3fti),cdt,fin,cpb,pbt]).
case(model_num(2),case_num(222),[rad,avp,not(pco),not(epr),not(psa),ifg,fpp,not(ihe)
      ,ils,crc,not(m3fti),l3fti,not(cdt),fin,not(cpb),not(pbt)]).
case(model_num(2),case_num(223),[rad,avp,not(pco),not(epr),not(psa),ifg,fpp,not(ihe)
      ,ils,crc,not(m3fti),not(l3fti),cdt,fin,not(cpb),not(pbt)]).
case(model_num(2),case_num(224),[psa,not(pco),not(epr),not(avp),ifg,not(fpp)]).
case(model_num(2),case_num(225),[psa,not(pco),not(epr),not(avp),ifg,fpp,ihe,not(ils)
      ,crc,m3fti,not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(226),[psa,not(pco),not(epr),not(avp),ifg,fpp,not(ihe),ils
      ,crc,not(m3fti),not(l3fti),not(cdt),fin]).
case(model_num(2),case_num(227),[psa,not(pco),not(epr),not(avp),ifg,fpp,not(ihe),ils
      ,crc,not(m3fti),l3fti,not(cdt),fin,cpb,pbt]).
case(model_num(2),case_num(228),[psa,not(pco),not(epr),not(avp),ifg,fpp,not(ihe),ils
      ,crc,not(m3fti),not(l3fti),cdt,fin,cpb,pbt]).
case(model_num(2),case_num(229),[psa,not(pco),not(epr),not(avp),ifg,fpp,not(ihe),ils
      ,crc,not(m3fti),l3fti,not(cdt),fin,not(cpb),not(pbt)]).
case(model_num(2),case_num(230),[psa,not(pco),not(epr),not(avp),ifg,fpp,not(ihe),ils
      ,crc,not(m3fti),not(l3fti),cdt,fin,not(cpb),not(pbt)]).


case(model_num(1),case_num(101),[not(dmg)]).
case(model_num(1),case_num(102),[not(dut),dmg,not(unl),not(vrt),not(vst),not(vun)]).
case(model_num(1),case_num(103),[not(dut),dmg,unl,not(imp),not(ift),not(ila),not(ico
      )]).
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(106),[dut,dmg,unl,imp,cau,vrt,not(vst),not(vun),not(ift),
      ila,not(ico),not(jus),prp]).
case(model_num(1),case_num(107),[dut,dmg,unl,imp,cau,vrt,not(vst),not(vun),not(ift),
      not(ila),ico,not(jus),prp]).
case(model_num(1),case_num(108),[dut,dmg,unl,imp,cau,not(vrt),vst,not(vun),ift,not(
      ila),not(ico),not(jus),prp]).
case(model_num(1),case_num(109),[dut,dmg,unl,imp,cau,not(vrt),vst,not(vun),not(ift),
      ila,not(ico),not(jus),prp]).
case(model_num(1),case_num(110),[dut,dmg,unl,imp,cau,not(vrt),vst,not(vun),not(ift),
      not(ila),ico,not(jus),prp]).
case(model_num(1),case_num(111),[dut,dmg,unl,imp,cau,not(vrt),not(vst),vun,ift,not(
      ila),not(ico),not(jus),prp]).
case(model_num(1),case_num(112),[dut,dmg,unl,imp,cau,not(vrt),not(vst),vun,not(ift),
      ila,not(ico),not(jus),prp]).
case(model_num(1),case_num(113),[dut,dmg,unl,imp,cau,not(vrt),not(vst),vun,not(ift),
      not(ila),ico,not(jus),prp]).
case(model_num(1),case_num(114),[not(dut),dmg,not(unl),vrt,not(vst),jus]).
case(model_num(1),case_num(115),[not(dut),dmg,not(unl),not(vrt),vst,jus]).
case(model_num(1),case_num(116),[not(dut),dmg,unl,imp,cau,vst,not(prp)]).

%The set of cases

cases_set(model_num(2),[case_num(201),case_num(202),case_num(203),case_num(204),
      case_num(205),case_num(206),case_num(207),case_num(208),case_num(209),case_num
      (210),case_num(211),case_num(212),case_num(213),case_num(214),case_num(215),
      case_num(216),case_num(217),case_num(218),case_num(219),case_num(220),case_num
      (221),case_num(222),case_num(223),case_num(224),case_num(225),case_num(226),
      case_num(227),case_num(228),case_num(229),case_num(230)]).

cases_set(model_num(1),[case_num(101),case_num(102),case_num(103),case_num(104),
      case_num(105),case_num(106),case_num(107),case_num(108),case_num(109),case_num
      (110),case_num(111),case_num(112),case_num(113),case_num(114),case_num(115),
      case_num(116)]).

%The list of cases with preferred relation

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(107),case_num(108),case_num(109),case_num
      (110),case_num(111),case_num(112),case_num(113)],[case_num(114),case_num(115),
      case_num(116)]]).

case_order(model_num(2),[case_num(201),[case_num(202),case_num(203),case_num(210),
      case_num(217),case_num(224)],[case_num(204),case_num(205),case_num(206),
      case_num(207),case_num(208),case_num(209),case_num(211),case_num(212),case_num
      (213),case_num(214),case_num(215),case_num(216),case_num(218),case_num(219),
      case_num(220),case_num(221),case_num(222),case_num(223),case_num(225),case_num
      (226),case_num(227),case_num(228),case_num(229),case_num(230)]]).

% Case order validity.

duplicate_case(Cases_list,[Case]) :- member(Case,Cases_list).
duplicate_case(Cases_list,[Case|Cases_set]) :-
        member(Case,Cases_list),
        duplicate_case(Cases_list,Cases_set).

ordering_valid(model_num(N)) :-
        cases_without_preference(model_num(N),Cases_list),
        cases_set(model_num(N),Cases_set),
        member_list(Cases_list,Cases_set),
        length_list(Cases_list,Length1),
        length_list(Cases_set,Length2),
        Length1 = Length2,
```

```
        duplicate_case(Cases_list,Cases_set),!.

%Cases: consistent, incompatible, different

consistent_case(_,_,[]) :- !.
consistent_case(model_num(N),case(model_num(N),M,Case_M),[H|T]) :-
        case(model_num(N),M,Case_M),
        case(model_num(N),H,Case_H),
        not(not(Case_M) = Case_H),
        consistent_case(model_num(N),case(model_num(N),M,Case_M),T).

consistent_list(_,[]) :- !.
consistent_list(model_num(N),[H|T]) :-
      cases_without_preference(model_num(N),Cases_list),
      consistent_case(model_num(N),case(model_num(N),H,_),Cases_list),
      consistent_list(model_num(N),T).

case_model_consistent(model_num(N)) :-
        case_model(model_num(N)),
        case_order(model_num(N),Case_order),
        remove_sublists(Case_order,Cases_list),
        consistent_list(model_num(N),Cases_list),!.

incompatible_case(_,_,[]) :- !.
incompatible_case(model_num(N),M,[H|T]) :-
      M = H,
      incompatible_case(model_num(N),M,T);
      case(model_num(N),M,Case_M),
      case(model_num(N),H,Case_H),
      contradict_list_list(Case_M,Case_H),
      incompatible_case(model_num(N),M,T).

incompatible_list(_,[]) :- !.
incompatible_list(model_num(N),[H|T]) :-
      cases_without_preference(model_num(N),Cases_list),
      incompatible_case(model_num(N),H,Cases_list),
      incompatible_list(model_num(N),T).

case_model_incompatible(model_num(N)) :-
        case_order(model_num(N),Case_order),
        remove_sublists(Case_order,Cases_list),
        incompatible_list(model_num(N),Cases_list),!.

different_case(_,_,[]) :- !.
different_case(model_num(N),M,[H|T]) :-
      M = H,
      different_case(model_num(N),M,T);
      case(model_num(N),M,Case_M),
      case(model_num(N),H,Case_H),
      not(Case_M = Case_H),
      different_case(model_num(N),M,T).

different_list(_,[]) :- !.
different_list(model_num(N),[H|T]) :-
      cases_without_preference(model_num(N),Cases_list),
      different_case(model_num(N),H,Cases_list),
      different_list(model_num(N),T).

case_model_different(model_num(N)) :-
        case_model(model_num(N)),
        case_order(model_num(N),Case_order),
        remove_sublists(Case_order,Cases_list),
        different_list(model_num(N),Cases_list),!.

%Case model validity

case_model_valid(model_num(N)) :-
      case_model_consistent(model_num(N)),
      case_model_incompatible(model_num(N)),
      case_model_different(model_num(N)),
      ordering_valid(model_num(N)),!.

%Coherent Arguments

coherent_casemade(_,_,_) :- fail.
coherent_casemade(Premise,Conclusion,[Case_num|Other_cases]) :-
      case(_,Case_num,Case),
      append(Premise,Conclusion,Casemade),
      member_list(Casemade,Case),!;
      coherent_casemade(Premise,Conclusion,Other_cases).

coherent(argument(model_num(N),argu_num(_),Premise,Conclusion)) :-
        cases_without_preference(model_num(N),Cases_list),
        coherent_casemade(Premise,Conclusion,Cases_list).

%Conclusive Arguments

case_with_casemade(model_num(N),Premise,Conclusion,Case_num) :-
    case(model_num(N),Case_num,Case),
    append(Premise,Conclusion,Casemade),
    member_list(Premise,Case),
    member_list(Casemade,Case),!.

conclusive_case_check(model_num(N),Premise,Conclusion,Case_num) :-
    case_with_casemade(model_num(N),Premise,Conclusion,Case_num),!;
    case(model_num(N),Case_num,Case),
    not(member_list(Premise,Case)).

conclusive_cases_list_check(model_num(N),Premise,Conclusion,[Case_num]) :-
```

```
        conclusive_case_check(model_num(N),Premise,Conclusion,Case_num),!.
conclusive_cases_list_check(model_num(N),Premise,Conclusion,[Case_num|Other_cases])
    :-
        conclusive_case_check(model_num(N),Premise,Conclusion,Case_num),
        conclusive_cases_list_check(model_num(N),Premise,Conclusion,Other_cases).

conclusive(argument(model_num(N),argu_num(_),Premise,Conclusion)) :-
          coherent(argument(model_num(N),argu_num(_),Premise,Conclusion)),
          cases_without_preference(model_num(N),Cases_list),
          conclusive_cases_list_check(model_num(N),Premise,Conclusion,Cases_list),!.

%Presumptively valid Arguments

best_case_casemade_basic(model_num(_),Premise,Conclusion,Case_num,Best_case_casemade
    ) :-
    case(model_num(_),Case_num,Case),
    append(Premise,Conclusion,Casemade),
    member_list(Casemade,Case),
    Best_case_casemade = Case_num.

best_case_casemade(model_num(_),_,_,_) :- !.
best_case_casemade(model_num(_),Premise,Conclusion,[Case_num|Other_cases],
     Best_case_casemade) :-
       best_case_casemade_basic(model_num(_),Premise,Conclusion,Case_num,
            Best_case_casemade),!;
       best_case_casemade(model_num(_),Premise,Conclusion,Other_cases,
            Best_case_casemade).

comparison_premise_sublist(model_num(N),Premise,[Case_num]) :-
       case(model_num(N),Case_num,Case),
       not(member_list(Premise,Case)),!.
comparison_premise_sublist(model_num(N),Premise,[Case_num|Other_cases]) :-
       is_list([Case_num|Other_cases]),
       case(model_num(N),Case_num,Case),
       not(member_list(Premise,Case)),
       comparison_premise_sublist(model_num(N),Premise,Other_cases).

comparison_premise_case(model_num(N),Premise,E) :-
       case(model_num(N),E,Case),
       not(member_list(Premise,Case)),!.

comparison_premise_basic(model_num(N),Premise,Case_num) :-
  comparison_premise_sublist(model_num(N),Premise,Case_num),!;
  comparison_premise_case(model_num(N),Premise,Case_num).

comparison_premise(_,Best_case_casemade,_,[Element]) :-
    Best_case_casemade = Element,!;
    member(Best_case_casemade,Element),!.
comparison_premise(model_num(N),Best_case_casemade,Premise,[Element|Other_cases]) :-
    Best_case_casemade = Element,!;
    member(Best_case_casemade,Element),!;
    comparison_premise_basic(model_num(N),Premise,Element),
    comparison_premise(model_num(N),Best_case_casemade,Premise,Other_cases).

presumptively_valid(argument(model_num(N),argu_num(_),Premise,Conclusion)) :-
           coherent(argument(model_num(N),argu_num(_),Premise,Conclusion)),
           case_order(model_num(N),Case_order),
           cases_without_preference(model_num(N),Cases_list),
           best_case_casemade(model_num(N),Premise,Conclusion,Cases_list,
                Best_case_casemade),
           comparison_premise(model_num(N),Best_case_casemade,Premise,Case_order),!.

%Successful Attack

successful_attack(argument(model_num(N),argu_num(_),Premise,Conclusion),
     Defeating_circumstance) :-
           append(Premise,Defeating_circumstance,Premise_and_defeating),
           presumptively_valid(argument(model_num(N),argu_num(_),Premise,Conclusion))
                ,
           not(presumptively_valid(argument(model_num(N),argu_num(_),
                Premise_and_defeating,Conclusion))).

%Rebutting Attack

rebutting_attack(argument(model_num(N),argu_num(_),Premise,Conclusion),
     Defeating_circumstance) :-
           successful_attack(argument(model_num(N),argu_num(_),Premise,Conclusion),
                Defeating_circumstance),
           negation_conclusion(Conclusion,Neg_C),
           append(Premise,Defeating_circumstance,Premise_and_defeating),
           presumptively_valid(argument(model_num(N),argu_num(_),
                Premise_and_defeating,Neg_C)).

%Undercutting Attack

undercutting_attack(argument(model_num(N),argu_num(_),Premise,Conclusion),
     Defeating_circumstance) :-
           not(rebutting_attack(argument(model_num(N),argu_num(_),Premise,Conclusion)
                ,Defeating_circumstance)).

presumption(argument(model_num(N),argu_num(_),Premise,Conclusion)):-
           Premise = [],
           presumptively_valid(argument(model_num(N),argu_num(_),Premise,Conclusion))
                .

%Undermining Attack

undermining_attack(argument(model_num(N),argu_num(_),Premise,Conclusion),
     Defeating_circumstance) :-
```

```
        presumption(argument(model_num(N),argu_num(_),Premise,Conclusion)),
        successful_attack(argument(model_num(N),argu_num(_),Premise,Conclusion),
            Defeating_circumstance).
```

Chinese copyright infringement model

The article of Copyright Infringement in Chinese Criminal Law [5] 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.

In Art. 217, there are 4 kinds of act in copyright infringement, if someone violates other people's copyright for the purpose of making profits, then he will be sentenced to the crime of copyright infringement. The judge will sentence the person to one of 4 different kinds of punishment according to the degree of severity of his crime. According to the articles related to Art. 217 in Chinese criminal law [5], and relevant official judicial interpretations [6], if the defendant satisfies the conditions of probation, then he will be put on probation. There is a defeating circumstance: the action is not belong to "without permission of the copyright owner".

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	the act was publishing a book of which the exclusive right of publication is enjoyed by another person
avp	the act was audio or video recording produced by another person without permission of the producer
psa	the act was producing or selling a work of fine art with forged signature of another painter
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

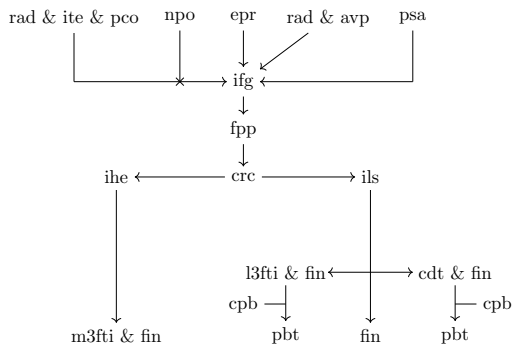
In the light of Art. 217 and the judicial interpretations related to it, a case model can be built. In our copyright infringement model, we use the elementary propositions in Table 2, shown with their formal abbreviations. The full model has 46 cases. A selection of the cases is shown in its Prolog version in Listing 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. Case 1 is built by the principle of "presumption of innocence". Case 2 is for the first defeating circumstance mentioned above. Case 3 shows the scenario that the defendant violated someone else's copyright, but he didn't do it for making profits. From Case 4 to Case 9, different punishments for the defendant's action

in Art. 217:1 are listed. In the same way, different punishments for the defendant in Art. 217:2, Art. 217:3 and Art. 217:4 are listed in the full case model. Thus Case 10 to Case 30 have similar components with Case 4 to Case 9, except the acts of infringement on copyright are different. In this model, Case 1 is maximal in the ordering, as the "presumption of innocence" is the starting principle in the process of decision making. Besides, in some situations, the defendant will not be regarded as violating someone else's copyright, for instance, Case 2 and Case 3; hence they appear in second place. The rest of cases in the preferred relation are the specific punishments of copyright infringement, they are less preferred than the cases in first and second place.

From Chinese copyright infringement, we can analyze the argument structure as in the diagram in Figure 1 (left). This argument structure shows multiple rule-based steps and exception-based attacks. The structure is valid in the case model we built, as follows.

Table 2: The Chinese copyright infringement case model

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 10	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,¬fpp
Case 11	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,fpp,ihe,¬ils,crc,m3fti,¬l3fti,¬cdt,fin
Case 12	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,¬cdt,fin
Case 13	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,cpb,pbt
Case 14	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,cpb,pbt
Case 15	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,¬cpb,¬pbt
Case 16	¬rad,¬ite,¬pco,epr,¬avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,¬cpb,¬pbt
Case 17	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,¬fpp
Case 18	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,fpp,ihe,¬ils,crc,m3fti,¬l3fti,¬cdt,fin
Case 19	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,¬cdt,fin
Case 20	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,cpb,pbt
Case 21	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,cpb,pbt
Case 22	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,¬cpb,¬pbt
Case 23	rad,¬ite,¬pco,¬epr,avp,¬psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,¬cpb,¬pbt
Case 24	¬rad,¬ite,¬pco,¬epr,¬avp,psa,ifg,¬fpp
Case 25	¬rad,¬ite,¬pco,¬epr,¬avp,psa,ifg,fpp,ihe,¬ils,crc,m3fti,¬l3fti,¬cdt,fin
Case 26	¬rad,¬ite,¬pco,¬epr,¬avp,psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,¬cdt,fin
Case 27	¬rad,¬ite,¬pco,¬epr,¬avp,psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,cpb,pbt
Case 28	¬rad,¬ite,¬pco,¬epr,¬avp,psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,¬l3fti,cdt,fin,cpb,pbt
Case 29	¬rad,¬ite,¬pco,¬epr,¬avp,psa,ifg,fpp,¬ihe,ils,crc,¬m3fti,l3fti,¬cdt,fin,¬cpb,¬pbt
Case 30	¬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 10 = Case 17 = Case 24> Case 4 = Case 5 = Case 6 = Case 7 = Case 8 = Case 9 = Case 11 = Case 12 = Case 13 = Case 14 = Case 15 = Case 16 = Case 18 = Case 19 = Case 20 = Case 21 = Case 22 = Case 23 = Case 25 = Case 26 = Case 27 = Case 28 = Case 29 = Case 30




```
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 1: The Chinese copyright infringement model: argument structure (left); in Prolog (right)

Following the definitions of the case model formalism, the rule ifg $\wedge$ fpp $\Rightarrow$ crc is indeed valid in this model, in different ways. As the model shows, this argument is coherent, as all cases imply sentence ifg $\wedge$ fpp $\wedge$ crc which is the case made by the argument (ifg $\wedge$ fpp, crc). Besides, all the cases which imply the premises ifg $\wedge$ fpp, also imply the conclusion crc. So argument (ifg $\wedge$ fpp, crc) is also conclusive in the case model, and hence it is presumptively valid. And these arguments are also conclusive in the model:

- $(C, \geq) \models$ crc $\wedge$ ihe $\Rightarrow$ m3fti $\wedge$ fin
- $(C, \geq) \models$ crc $\wedge$ ils $\wedge$ l3fti $\wedge$ fin $\wedge$ cpb $\Rightarrow$ pbt
- $(C, \geq) \models$ crc $\wedge$ ils $\wedge$ cdt $\wedge$ fin $\wedge$ cpb $\Rightarrow$ pbt

Argument (rad $\wedge$ ite $\wedge$ pco, ifg) is presumptively valid. In the copyright infringement model, this argument is coherent as Case 2 implies rad $\wedge$ ite $\wedge$ pco $\wedge$ ifg, the case made by the argument. As Case 2 is also the strongest case among the cases which imply the premises of this argument, the argument (rad $\wedge$ ite $\wedge$ pco, ifg) is presumptively valid in the case model. It is not conclusive, as the defeating circumstance npo successfully attacks the argument. Since the case model makes the argument (rad $\wedge$ ite $\wedge$ pco $\wedge$ npo, $\neg$ifg) presumptively valid, this defeating circumstance is rebutting. In the same way, the following hold in the case model.

- $(C, \subseteq) \models$ crc $\rightsquigarrow$ ils
- $(C, \subseteq) \models$ crc $\wedge$ ils $\rightsquigarrow$ l3fti $\wedge$ fin
- $(C, \subseteq) \models$ rad $\wedge$ ite $\wedge$ pco $\rightsquigarrow$ ifg $\times$ npo

The Prolog program confirms the validity of the arguments above. Figure 1 (right) lists Prolog queries evaluated as true, which means the results of the Prolog program correspond to our analysis of the Chinese copyright infringement model.

References

[1] Patrick Blackburn, Johan Bos, and Kristina Striegnitz. Learn prolog now!, 2001. *Bib y en Red.*

[2] T. Fruehwirth, J. Wielemaker, L. de Koninck, and M. Triska. *SWI Prolog Reference Manual 6.2.2.* Books on Demand, 2012.

[3] Ivan Bratko. *Prolog programming for artificial intelligence.* Pearson education, 2001.

[4] Werner Hett. P-99: Ninety-nine prolog problems. https://sites.google.com/site/prologsite/prolog-problems.

[5] 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.

[6] 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.