



HAL
open science

Compliance Check of Norms for Algorithmic Law

Ken Satoh

► **To cite this version:**

| Ken Satoh. Compliance Check of Norms for Algorithmic Law. 2023. hal-04227440

HAL Id: hal-04227440

<https://hal.science/hal-04227440v1>

Preprint submitted on 3 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Compliance Check of Norms for Algorithmic Law

Ken SATOH

(National Institute of Informatics)
2022 FFJ / Air Liquide Fellow

October 2023

Compliance Check of Norms for Algorithmic Law

Ken SATOH

Abstract

Along with advances of artificial intelligence into society, a situation has emerged in which artificial intelligence is closely related to the real world and legal and ethical problems caused by artificial intelligence naturally occur. Problems with AI decision making systems used by governments would be very critical since it directly relates with human rights. We call this kind of AI systems by which governments replace human officials “algorithmic law”. In this research, we apply logical inference for conflict detection and resolution for compliance check of legal and ethical norms to algorithmic law whose specification is represented explicitly (which we call white box algorithmic law) and we apply case-based generalization to generalize a violated case in algorithmic law whose specification is not represented such as deep learning-based algorithmic law (which we call black box algorithmic law).

Keywords

Algorithmic Law, Compliance Check, Norm, White box Approach, Abductive Logic Programming, Black box Approach, Case-based Generalization

Acknowledgement

This research was prepared in part while Ken Satoh visited the Fondation France-Japon (FFJ) de l'EHESS as a 2022 FFJ/Air Liquide Fellow thanks to 2022 FFJ/Air Liquide Fellowship. He thanks the FFJ and the Air Liquide for their support and hospitality during his stay. This work was supported by JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4 and JSPS Grant-in-Aid for Challenging Research (Pioneering) Grant Number 19H05470, Japan.

Contents

Abstract	4
1. Introduction	4
2. White Box Approach	6
2.1 Overview of the Approach	6
2.2 A Semantics of Abductive Framework.....	7
2.3 Compliance Check Method using Abductive Framework.....	8
2.3.1 Subprocedure <i>derive</i>	9
2.3.2 Subprocedure <i>literal_con</i>	10
2.3.3 Subprocedure <i>rule_con</i>	11
2.3.4 Subprocedure <i>deleted_con</i>	11
2.4 Examples	12
2.4.1 GPRD.....	12
2.4.2 Driver License Qualification	14
2.4.3 Marriage Qualification	14
2.4.4 Loan Qualification	16
3. Black Box Approach	17
3.1 Overview of the approach	17
3.2 How to generalize a violated case	18
3.3 Example of compliance check for decision making for giving child rearing allowance	20
4. Conclusion	23
References	24

Compliance Check of Norms for Algorithmic Law

Ken Satoh

National Institute of Informatics

Abstract

Along with advances of artificial intelligence into society, a situation has emerged in which artificial intelligence is closely related to the real world and legal and ethical problems caused by artificial intelligence naturally occur. Problems with AI decision making systems used by governments would be very critical since it directly relates with human rights. We call this kind of AI systems by which governments replace human officials “algorithmic law”. In this research, we apply logical inference for conflict detection and resolution for compliance check of legal and ethical norms to algorithmic law whose specification is represented explicitly (which we call white box algorithmic law) and we apply case-based generalization to generalize a violated case in algorithmic law whose specification is not represented such as deep learning-based algorithmic law (which we call black box algorithmic law).

Acknowledgement

This research was prepared in part while Ken Satoh visited the Fondation France-Japon (FFJ) de l’ EHESS as a 2022 FFJ/Air Liquide Fellow thanks to 2022 FFJ/Air Liquide Fellowship. He thanks the FFJ and the Air Liquide for their support and hospitality during his stay. This work was supported by JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4 and JSPS Grant-in-Aid for Challenging Research (Pioneering) Grant Number 19H05470, Japan.

Keywords

Algorithmic Law, Compliance Check, Norm, White box Approach, Abductive Logic Programming, Black box Approach, Case-based Generalization

1 Introduction

Along with advances of artificial intelligence into society[10], a situation has emerged in which artificial intelligence is closely related to the real world and legal problems caused by artificial

intelligence naturally occur. In particular, if AI plays a role in decision making, it may cause a great harm to humans. Such problems have been actually occurring.

- Facebook had a strategy of advertisements in consideration of race and gender[8]which deprives minorities of the opportunity to see these ads in order to improve the lives of those who buy these products.
- The US recidivism prediction system, COMPAS, has been used to calculate the offense prediction rate of offenders for bail decisions since 1989. Recently, there is a doubt that racial consideration would be made in this recidivism prediction[11].
- Student clearinghouse called Admission Post Bac (APB), in France allocates students to universities, preparatory schools, and technical high schools. There is, however, no disclosure about decision making process. A trial was made to open the strategy to the public by a trial, and as a result of the analysis, a serious suspicion in decision making was revealed in another trial[12].

A serious problem occurs if governments use such AI systems to support decision making by governments or more seriously to replace human officials which make such decisions. We call this kind of AI systems by which governments replace human officials “algorithmic law” since these decisions made by the AI system will be legally effective to enforce people. Along with frequent uses of such AI systems, algorithmic law will be a great matter for civil rights. Therefore, civil control of algorithmic law should be prepared urgently. One way to solve these problems is to let governments to monitor such AI systems by themselves but the check for these decision making might be loosely done. So more effective way is to expose such AI systems to the public, especially for AI decision making systems used by governments. In fact, in France, a lawsuit has been invoked to require government agencies to publish a source program of the algorithmic law, and as a result, various software source was released[13].

However, even if a source program is published, there is a serious technical problem for checking compliance of the law; Currently, analysis of software code is done manually. According to increasing number of such software, it would be very difficult to find a problem by humans. For this reason, in this project, we develop methods of compliance check for algorithmic law. Our original goal was to develop automatic compliance check of “white box algorithmic law” which has a clear specification of the algorithm. To do this, we extend our proof procedure of abductive logic programming[7] to infer which hypothetical situations lead to violation. However, due to advance of deep learning technology, there would be a possibility of using neural-based algorithmic law for decision making so we decide to consider this type of algorithmic law (we call it here “black box algorithmic law”) as well. One possible approach for “black box algorithmic law” is that we induce logical rules from the algorithm in order to make it “white box algorithmic law” and then use the above compliance check method for a derived “white box algorithm law”. However, it would be costly to get all the logical rules and actually only we have to accomplish is to find a part of algorithm violating with legal/ethical

rules. Therefore, instead of making “black box algorithm law” into “white box algorithm law”, every time we find a violated case, we generalize violated cases to avoid further violation using case-based learning.

There have been various proposals for compliance check[9, 5, 2, 4, 3] for “white box algorithmic law”. [9] takes an approach in that they check compliance during execution dynamically whereas we take a different approach for static compliance check from specification. [5] checks whether a set of execution traces violate compliance rules or not. So, the check is not overall but only for a set of the concrete cases. On the other hand, our proposal checks all the situations and detect all the possible violated situations. [2] checks whether an incomplete execution trace in business process violates compliance rules which are represented as integrity constraints by completing incomplete execution using abduction. Our approach also uses abduction but we use abduction for the different purpose to represent a violated case. In [4], we propose an ODRL (the Open Digital Rights Language) regulatory compliance profile that can be used to model both regulatory requirements in terms of deontic concepts (permissions, prohibitions, obligations and dispensations), and business policies in the forms of sets of permissions required to execute the policy. The ODRL policies are subsequently translated into ASP rules, which not only cater for automatic compliance checking, but also for non-compliance detection and explanation for a concrete execution trace. So it has the same problem as[5]. [3] uses natural language processing to suggest problematic clauses in privacy policies which would not be compliant GDPR (General Data Protection Regulation). However, the work is based on deep learning based NLP so there would be no guarantee for finding all the problematic clauses.

About compliance check for black box algorithmic laws, on the other hand, although there are researchers discussing discrimination or biases in deep neural network, compliance check for black box algorithmic laws in general has not been addressed, so our approach seem to be the initial proposal for such black box algorithms.

2 White Box Approach

2.1 Overview of the Approach

The system architecture of our compliance check for “white box algorithmic law” can be seen at Fig. 1. We translate a specification of the AI system implementing algorithmic law into a logical formula. We assume that there will be a rigorous specification of the AI system. Once we have a rigorous specification, it would be easy to translate it into a logical formula. We also formalize a situation of violation as an integrity constraint in logic programming. If there is a violation of constraints, we could trace logical reasoning step and identify which part of the specification is a source of violating the constraints. Once we find a conflict, we manually revise the specification of the AI system. We iterate this process until no violations find then we use this compliant algorithm for normal processing of decision making.

We also assume that there is an ontological knowledge which relates basic operation in the

algorithmic law with predicates in integrity constraints. Then, we will check compliance of algorithmic law as follows:

- Firstly, we translate integrity constraints into a rule with the head of the rule as `violation(...)`.
- We also add all the facts as abducibles so that we do not specify any variables except variables which must be different.
- We then call a query evaluation procedure of abductive logic programming with a goal of `violation(...)`.
- If there is a set of assumptions returned with the success of the goal `violation(...)`, then the set of assumptions shows an uncompliant case.

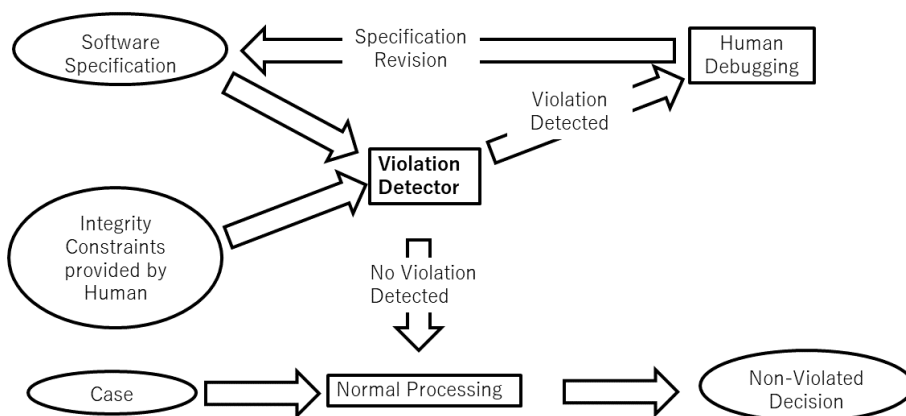


Figure 1: The Architecture for White Box Algorithmic Law

2.2 A Semantics of Abductive Framework

We mainly follow the definition of abductive framework in [7], but we modify it slightly for notational conveniences. Firstly, we define a rule and an integrity constraint.

Definition 1 Let H be an atom, and $L_1, \dots, L_m (m \geq 0)$ be literals each of which is an atom or a negated atom of the form $\sim B$. A rule is of the form:

$$H \Leftarrow L_1, L_2, \dots, L_m.$$

We call H the *head* of the rule and L_1, \dots, L_m the *body* of the rule. Let R be a rule. $head(R)$, $body(R)$ and $pos(R)$ denote the head of R , the set of literals in the body of R and the set of positive literals in $body(R)$ respectively.

Definition 2 Let $L_1, \dots, L_m (m \geq 0)$ be literals. An integrity constraint is of the form:

$$\perp \Leftarrow L_1, L_2, \dots, L_m.$$

For a given program (with integrity constraints), we define a stable model (in other words, answer set) as follows.

Definition 3 Let T be a logic program and Π_T be a set of ground rules obtained by replacing all variables in each rule in T by every element of its Herbrand universe¹. Let M be a set of ground atoms from Π_T and Π_T^M be the following (possibly infinite) program.

$$\Pi_T^M = \{H \Leftarrow B_1, \dots, B_k \mid H \Leftarrow B_1, \dots, B_k, \sim A_1, \dots, \sim A_m \in \Pi_T \\ \text{and } A_i \notin M \text{ for each } i = 1, \dots, m.\}$$

Let $\min(\Pi_T^M)$ be the least model of Π_T^M . A stable model for a logic program T is M iff $M = \min(\Pi_T^M)$ and $\perp \notin M$.

This definition gives a stable model of T which satisfies all integrity constraints. We say that T is *consistent* if there exists a stable model for T .

Now, we define an *abductive framework*.

Definition 4 An abductive framework is a pair $\langle T, A \rangle$ where A is a set of predicate symbols, called *abducible predicates* and T is a set of rules each of whose head is not in A .

We call a set of all ground atoms for predicates in A *abducibles*.

Now, we define a semantics of an abductive framework.

Definition 5 Let $\langle T, A \rangle$ be an abductive framework and Θ be a set of abducibles. A generalized stable model $M(\Theta)$ is a stable model of $T \cup \{H \Leftarrow \mid H \in \Theta\}$.

We say that $\langle T, A \rangle$ is *consistent* if there exists a generalized stable model $M(\Theta)$ for some Θ .

2.3 Compliance Check Method using Abductive Framework

Before showing our query evaluation method, we need the following definitions. Let l be a literal. Then, \tilde{l} denotes the complement of l .

Definition 6 Let T be a logic program. A set of resolvents w.r.t. a literal l and T , $\text{resolve}(l, T)$ is the following set of rules:

$$\text{resolve}(l, T) = \\ \{(\perp \Leftarrow L_1, \dots, L_k)\theta \mid l \text{ is negative and} \\ H \Leftarrow L_1, \dots, L_k \in T \text{ and } \tilde{l} = H\theta \text{ by the most general unifier, } \theta\} \cup \\ \{(H \Leftarrow L_1, \dots, L_{i-1}, L_{i+1}, \dots, L_k)\theta \mid \\ H \Leftarrow L_1, \dots, L_k \in T \text{ and } l = L_i\theta \text{ by the most general unifier, } \theta\}$$

¹We assume that we virtually have infinite number of constants in T

The first set of resolvents are for negation as failure and the second set of resolvents corresponds with “forward” evaluation of the rule introduced in [6].

Definition 7 *Let T be a logic program. A set of deleted rules w.r.t. a literal l and T , $del(l, T)$, is the following set of rules:*

$$del(l, T) = \{(H \leftarrow L_1, \dots, L_k)\theta \mid H \leftarrow L_1, \dots, L_k \in T \text{ and } \tilde{l} = L_i\theta \text{ by the most general unifier, } \theta\}$$

Our query evaluation procedure consists of 4 subprocedures, $derive(p, \Delta)$, $literal_con(l, \Delta)$, $rule_con(R, \Delta)$ and $deleted_con(R, \Delta)$ where p is a non-abducible atom and Δ is a set of literals already assumed and l is a literal and R is a rule.

$derive(p, \Delta)$ returns a ground substitution for the variables in p and a set of literals. This set of literals is a union of Δ and literals newly assumed during execution of the subprocedure.

The subprocedures have a **select** operation and a **fail** operation. The **select** operation expresses a nondeterministic choice among alternatives. The **fail** operation expresses immediate termination of an execution with failure. Therefore, a subprocedure succeeds when its inner calls of subprocedures do not encounter **fail**. We say *a subprocedure succeeds with (θ and) Δ* when the subprocedure successfully returns (θ and) Δ .

The informal specification of the 4 subprocedures is as follows.

1. $derive(p, \Delta)$ searches a rule R of p in a program T whose body can be made true with a substitution θ under a set of assumptions Δ . To show that every literal in the body can be made true, we call $derive$ for non-abducible positive literals in the body. Then, we check the consistency of other literals in the body with T and Δ .
2. $literal_con(l, \Delta)$ checks the consistency of a literal l with T and Δ . To show the consistency for assuming l , we add l to Δ ; then, we check the consistency of resolvents and deleted rules w.r.t. l and T .
3. $rule_con(R, \Delta)$ checks the consistency of a rule R with T and Δ . We can prove the consistency by showing that either a literal in $body(R)$ can be falsified or $body(R)$ can be made true and $head(R)$ consistent.

This procedure can also be used to check integrity for rule addition.

4. $deleted_con(R, \Delta)$ checks if a deletion of R does not cause any contradictions with T and Δ . To show the consistency of the implicit deletion of R , it is sufficient to prove that the head of R can be made either true or false.

2.3.1 Subprocedure $derive$

$derive(p, \Delta)$

p : a non-abducible atom; Δ : a set of literals

```

begin
  if  $p$  is ground and  $p \in \Delta$  then return  $(\varepsilon, \Delta)$ 
  elseif  $p$  is ground and  $\sim p \in \Delta$  then fail
  else
    begin
      select  $R \in T$  s.t.  $head(R)\theta = p\theta$ 
      if such a rule is not found then fail
       $\Delta_0 := \Delta$  and  $\theta_0 := \theta$  and  $i := 0$ 
      for every  $l \in pos(R\theta_i)$  do
        if  $derive(l, \Delta_i)$  succeeds with  $(\sigma_i, \Delta_{i+1})$ 
          then  $\theta_{i+1} := \theta_i\sigma_i$  and  $i := i + 1$  and continue
         $\delta := \theta_i$ 
        for every  $l \in abd(R\delta) \cup neg(R\delta)$  do
          if  $literal\_con(l, \Delta_i)$  succeeds with  $\Delta_{i+1}$ 
            then  $i := i + 1$  and continue
          if  $literal\_con(p\delta, \Delta_i)$  succeeds with  $\Delta'$ 
            then return  $(\delta, \Delta')$ 
      end
    end
  end ( $derive$ )

```

2.3.2 Subprocedure $literal_con$

$literal_con(l, \Delta)$

l : a literal; Δ : a set of literals

```

begin
  if  $l \in \Delta$  then return  $\Delta$ 
  elseif  $l = \perp$  or  $\tilde{l} \in \Delta$  then fail
  else
    begin
       $\Delta_0 := \{l\} \cup \Delta$ ,  $i := 0$ 
      for every  $R \in resolve(l, T)$  do
        if  $rule\_con(R, \Delta_i)$  succeeds with  $\Delta_{i+1}$ 
          then  $i := i + 1$  and continue
      for every  $R \in del(l, T)$  do
        if  $deleted\_con(R, \Delta_i)$  succeeds with  $\Delta_{i+1}$ 
          then  $i := i + 1$  and continue
      end
      return  $\Delta_i$ 
    end
  end ( $literal\_con$ )

```

2.3.3 Subprocedure *rule_con*

rule_con(R, Δ) R : rule; Δ : set of literals

```

begin
   $\Delta_0 := \Delta, i := 0$ 
  select (a) or (b)
  (a) select  $l \in \text{body}(R)$ 
    if  $l$  is positive or abducible and
      literal_con( $\tilde{l}, \Delta_i$ ) succeeds with  $\Delta_{i+1}$ 
      then  $i := i + 1$  and continue
    elseif  $l$  is negative and
      derive( $\tilde{l}, \Delta$ ) succeeds with  $(\varepsilon, \Delta_{i+1})$ 
      then  $i := i + 1$  and continue
  (b)  $\Delta_i^0 := \Delta_i, j := 0$ 
    for every  $p \in \text{pos}(R)$  do
      if derive( $p, \Delta_i^j$ ) succeeds with  $(\varepsilon, \Delta_i^{j+1})$ 
        then  $j := j + 1$  and continue
    for every  $l \in \text{abd}(R) \cup \text{neg}(R)$  do
      if literal_con( $l, \Delta_i^j$ ) succeeds with  $\Delta_i^{j+1}$ 
        then  $j := j + 1$  and continue
    if literal_con(head( $R$ ),  $\Delta_i^j$ ) succeeds with  $\Delta_{i+1}$ 
      then  $i := i + 1$  and continue
  return  $\Delta_i$ 
end (rule_con)

```

2.3.4 Subprocedure *deleted_con*

deleted_con(R, Δ) R : rule; Δ : set of literals

```

begin
   $\Delta_0 := \Delta, i := 0$ 
  select (a) or (b)
  (a) if derive(head( $R$ ),  $\Delta_i$ ) succeeds with  $(\varepsilon, \Delta_{i+1})$ 
    then  $i := i + 1$  and continue
  (b) if literal_con( $\sim$ head( $R$ ),  $\Delta_i$ ) succeeds with  $\Delta_{i+1}$ 
    then  $i := i + 1$  and continue
  return  $\Delta_i$ 
end (deleted_con)

```

Theorem 1 (*Correctness for Query Evaluation*)

Suppose *derive*($p, \{\}$) succeeds with (θ, Δ) . Then, there exists a generalized stable model $M(\Theta)$ for T such that $\text{abd}(\Delta) \subseteq \Theta$ and $M(\Theta) \models p\theta$.

2.4 Examples

We give the following four examples for compliance check.

1. Checking GDPR article 6-(a).
2. Checking discrimination of driver's license qualification according to eye sight.
3. Checking discrimination of marriage qualification according to gender.
4. Checking discrimination of loan acceptance with gender and annual salary.

2.4.1 GDPR

In GDPR article 6-(a) says that “Processing shall be lawful only if and to the extent that at least one of the following applies: (a) the data subject has given consent to the processing of his or her personal data for one or more specific purposes.” We represent the violation of 6-(a) as follows:

```
violation(C) <=  
  transfer(ControllerOrProcessor, personal_data(DataSubject, Data), ThirdParty),  
  not(get_consent(DataSubject, ControllerOrProcessor)).
```

In this example, there are three methods to get consents (`interaction1`, `interaction2`, `interaction3`) and we assume that we know in advance that only `interaction2` and `interaction3` satisfies getting true consent.

```
/* Ontological Knowledge about Consent*/  
get_consent(CorP, DS) <= interaction1(CorP, DS).  
get_consent(CorP, DS) <= interaction2(CorP, DS).
```

We also have the following definitions of each interaction.

```
transfer(CorP, personal_data(DS, DT), TP) <=  
  controller_or_processor(CorP),  
  personal_data(DS, DT),  
  third_party(TP),  
  interaction(CorP, DS),  
  transfer_act(CorP, personal_data(DS, DT), TP).
```

```
interaction(CorP, DS) <= interaction1(CorP, DS).  
interaction(CorP, DS) <= interaction2(CorP, DS).  
interaction(CorP, DS) <= interaction3(CorP, DS).
```

```

/* The last interaction is in an opt-in manner to get a consent */
interaction1(CorP,DS) <=
    ask_click_permission_button(CorP,DS).
interaction2(CorP,DS) <=
    ask_send_an_email_consent_to_c_or_p(CorP,DS).
interaction3(CorP,DS) <=
    ask_to_reject_if_not_consent(CorP,DS).

```

We also need a hypothetical case description which is implemented as abducibles for fact predicates.

```

/* abduce all the fact without instantiation */
controller_or_processor(CorP) <= abd(controller_or_processor(CorP)).
personal_data(DS,DT) <= abd(personal_data(DS,DT)).
third_party(TP) <= abd(third_party(TP)).
ask_click_permission_button(CorP,DS) <=
    abd(ask_click_permission_button(CorP,DS)).
ask_send_an_email_consent_to_c_or_p(CorP,DS) <=
    abd(ask_send_an_email_consent_to_c_or_p(CorP,DS)).
ask_to_reject_if_not_consent(CorP,DS) <=
    abd(ask_to_reject_if_not_consent(CorP,DS)).
transfer_act(CorP,personal_data(DS,DT),TP) <=
    abd(transfer_act(CorP,personal_data(DS,DT),TP)).

```

By asking violation(C), we get the following output:

```

violation(_806)
transfer(_824,personal_data(_832,_834),_828)
transfer_act(_824,personal_data(_832,_834),_828)
transfer_act*( _824,personal_data(_832,_834),_828)
interaction(_824,_832)
interaction3(_824,_832)
ask_to_reject_if_not_consent(_824,_832)
ask_to_reject_if_not_consent*( _824,_832)
third_party(_828)
third_party*( _828)
personal_data(_832,_834)
personal_data*( _832,_834)
controller_or_processor(_824)
controller_or_processor*( _824)

```

This means that interaction3 violates GDPR 6-(a).

2.4.2 Driver License Qualification

This example shows discrimination detection in the context of driver's license qualification. In this case, we need to compare two individuals (we use two constants 'a' and 'b' to represent two individuals in the program below) with different values of property (eyesight) to get different results and the system automatically detect such discrimination. Then, humans will check whether this distinction due to the property is reasonable or not and if it is not reasonable, it is regarded as discrimination. In this example, the distinguishing property is eyesight.

```
violation(C) <=
    different(Property,P1,P2),
    decision(C,P1,R1),
    decision(C,P2,R2),
    call(\+(R1=R2)).
decision(driver_license,X,accept) <=
    property(eyesight,X,normal).
decision(driver_license,X,reject) <=
    property(eyesight,X,narrow).
different(eyesight,P1,P2) <=
    property(eyesight,P1,normal),
    property(eyesight,P2,narrow).

property(eyesight,a,normal) <= true.
property(eyesight,b,narrow) <= true.
```

Output is as follows. Abductive logic programming system detects that there is a violation of equivalence according to eyesight where 'a' is rejected for driver's license but 'b' is rejected. Then humans will decide whether this distinction is reasonable or not.

```
violation(driver_license)
different(eyesight,a,b)
decision(driver_license,b,reject)
property(eyesight,b,narrow)
decision(driver_license,a,accept)
property(eyesight,a,normal)
```

2.4.3 Marriage Qualification

This example shows discrimination detection in the context of marriage qualification for individuals. We assume that there are three genders; male, female and other where male and female genders are allowed to get married but not other gender.

In this case, we need to compare three individuals (we use three constants 'a', 'b' and 'c' to represent three individuals in the program below) with different values of property (gender) to

get different results and the system automatically detects such discrimination. Then, humans will check whether this distinction due to the property is reasonable or not and if it is not reasonable, it is regarded as discrimination. In this example, the distinguishing property is gender.

```
violation(C) <=
    different(Property,P1,P2),
    decision(C,P1,R1),
    decision(C,P2,R2),
    call(\+(R1=R2)).
decision(marriage,X,accept) <=
    property(gender,X,male).
decision(marriage,X,accept) <=
    property(gender,X,female).
decision(marriage,X,reject) <=
    property(gender,X,other).
different(gender,P1,P2) <=
    property(gender,P1,male),
    property(gender,P2,female).
different(gender,P1,P2) <=
    property(gender,P1,male),
    property(gender,P2,other).
different(gender,P1,P2) <=
    property(gender,P1,female),
    property(gender,P2,other).
```

```
property(gender,a,male) <= true.
property(gender,b,female) <= true.
property(gender,c,other) <= true.
```

Output is as follows. Abductive logic programming system detects that there is a violation of equivalence according to gender where 'a' and 'b' are accepted for marriage but 'c' is rejected. Then humans will decide whether this distinction is reasonable or not.

```
different(gender,b,c)
different(gender,a,b)
violation(marriage)
different(gender,a,c)
decision(marriage,a,accept)
property(gender,a,male)
decision(marriage,c,reject)
property(gender,c,other)
```



```
decision(marriage,b,accept)
property(gender,b,female)
```

2.4.4 Loan Qualification

This example shows discrimination detection in the context of loan qualification. We assume that there are two genders; male and female and we assume that people could get loan but there are different conditions for income for each male and female applicant; if annual salary for male is more than 3 million yen, he could get a loan but if annual salary for female is more than 5 million yen, she could get a loan.

In this case, we need to compare two individuals (we use three constants 'a' and 'b' to represent two individuals in the program below) with different values of property (gender) to get different results.

```
violation(C) <=
  different(Property,P1,P2),
  decision(C,P1,R1),
  decision(C,P2,R2),
  call(\+(R1=R2)).
decision(loan,X,accept) <=
  property(gender,X,male),
  property(income,X,I),
  greater_than_or_equal(I,3).
decision(loan,X,reject) <=
  property(gender,X,male),
  property(income,X,I),
  less_than(I,3).
decision(loan,X,accept) <=
  property(gender,X,female),
  property(income,X,I),
  greater_than_or_equal(I,5).
decision(loan,X,reject) <=
  property(gender,X,female),
  property(income,X,I),
  less_than(I,5).

different(gender,P1,P2) <=
  property(gender,P1,male),
  property(gender,P2,female).

0 <= property(gender,P1,male), property(gender,P2,female),
```

```

    property(income,P1,I), not(property(income,P2,I)).
0 <= property(gender,P1,male), property(gender,P2,female),
    property(income,P2,I), not(property(income,P1,I)).

```

```

property(gender,a,male) <= true.
property(gender,b,female) <= true.

```

```

property(income,X,I) <= abd(property(income,X,I)).
greater_than_or_equal(I,3) <= abd(greater_than_or_equal(I,3)).
greater_than_or_equal(I,5) <= abd(greater_than_or_equal(I,5)).
less_than(I,3) <= abd(less_than(I,3)).
less_than(I,5) <= abd(less_than(I,5)).

```

Output is as follows. The system automatically detects when discrimination occurs that is, if the annual income for 'a' and 'b' are the same if the income (represented as `_1246`) is between 3 and 5 million yen, 'a' is accepted but 'b' is not. Then, humans will check whether this distinction due to the property is reasonable or not and if it is not reasonable, it is regarded as discrimination.

```

violation(loan)
decision(loan,b,reject)
less_than(_1246,5)
less_than*(_1246,5)
decision(loan,a,accept)
greater_than_or_equal(_1246,3)
greater_than_or_equal*(_1246,3)
different(gender,a,b)
property(income,a,_1246)
property*(income,a,_1246)
property(income,b,_1246)
property*(income,b,_1246)
property(gender,b,female)
property(gender,a,male)

```

3 Black Box Approach

3.1 Overview of the approach

The system architecture of our compliance check for “black box algorithmic law” can be seen at Fig. 2. In an administrative procedure, people usually fill in some forms of application which usually consists of fixed number of questionnaires and according the value of such questionnaires

(we call them *factors*). Suppose that we have n -factors for a decision and for each i -factors the government asks some a value among $V_i^1, \dots, V_i^{i_k}$. Then, using the answer from the applicant, the government virtually has a decision table based on a value of n -factors whether the decision is “accept” or “reject”. This means that we could represent the decision (for acceptance or rejection) in a logical formula (disjunctions of conjunctions of (factor,value) pairs). Moreover, we assume that a taxonomic relations in the values of factors so that once we find a violated case, we generalize it into general rules of violation to avoid further violation. Then, if an new input satisfies conditions of such violating rules, the system will warn this violation and humans make a decision for the new input instead of using the black box algorithm.

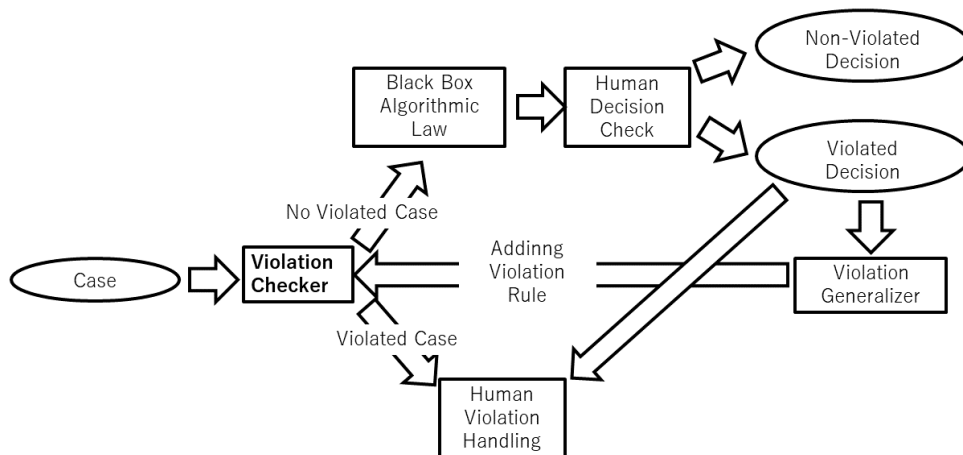


Figure 2: The Architecture for Black Box Algorithmic Law

3.2 How to generalize a violated case

Suppose that values of i -factors are $V_i^1, \dots, V_i^{i_k}$. In order to generalize a violated case, we assume that there is a tree structured relations between factors and super-factors. We assume that there is a tree where

- We denote the root node of the tree as \top .
- leaf nodes are $V_i^1, \dots, V_i^{i_k}$.
- there is a parent (super-factor) for each node n denoted as $parent(n)$.

We define the lowest common ancestor (LCA) (also called least common ancestor) of two nodes v and w in a tree as the lowest (i.e. deepest) node that has both v and w as descendants denoted as $lca(v, w)$. For example, in the tree of Fig. 3,

- $lca(V_1^1, V_1^1)$ is V_1^1 .
- $lca(V_1^1, V_1^2)$ is b .
- $lca(V_1^1, V_1^3)$ is a .
- $lca(V_1^1, V_1^4)$ is \top .

A case is represented as a tuple of m factors, $\langle V_1, V_2, \dots, V_m \rangle$. Let C be a case. We denote each value of i -th factor V_i as $C[i]$. Then, given a case $\langle V_1^1, V_2^1, \dots, V_m^1 \rangle$ (called the *bottom case* denoted as C_b), we can construct a lattice whose nodes are a set of cases and which expresses generalization relation of cases as follows.

Definition 8 Let C be a case. We define a set of cases $C \uparrow_{C_b}^l$ for $l(1 \leq l \leq m)$ as follows: $C' \in C \uparrow_{C_b}^l$ if C' satisfies the following condition:

- $lca(C_b[l], C'[l]) = \text{parent}(lca(C_b[l], C[l]))$
- $lca(C_b[j], C'[j]) = lca(C_b[j], C[j])$ for $j \neq l(1 \leq j \leq m)$.

Now, we construct the lattice as follows:

1. The bottom node corresponds with a set of the bottom case C_b .
2. Suppose that a node in the lattice be a set of cases $N = \{\langle C_1, \dots, C_n \rangle\}$ then we define l -th direct upper nodes from the node as follows: $\bigcup_{i=1}^n C_n \uparrow_{C_b}^l$ and denoted as $N \uparrow_{C_b}^l$.

Consider trees of Figs. 3 and 4 and $C_b = \{\langle V_1^1, V_2^1 \rangle\}$. The direct upper nodes of $\{C_b\} \uparrow_{C_b}^1$ and $\{C_b\} \uparrow_{C_b}^2$ are $\{C1\}$ and $\{C2, C3\}$ respectively where $C1 = \langle V_1^2, V_2^1 \rangle$, $C2 = \langle V_1^1, V_2^2 \rangle$ and $C3 = \langle V_1^1, V_2^3 \rangle$ since, for example,

- $lca(C_b[2], C2[2]) = lca(V_2^1, V_2^2) = x$,
 $lca(C_b[2], C3[2]) = lca(V_2^1, V_2^3) = x$
and $\text{parent}(lca(C_b[2], C_b[2])) = \text{parent}(lca(V_2^1, V_2^1)) = \text{parent}(V_2^1) = x$

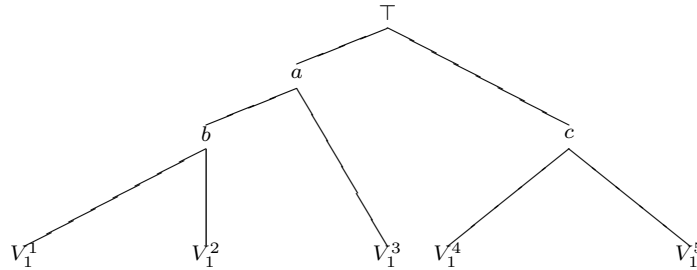


Figure 3: Tree Example1

- $lca(C_b[1], C2[1]) = lca(V_1^1, V_1^1) = a$
 $lca(C_b[1], C3[1]) = lca(V_1^1, V_1^1) = a$
and $lca(C_b[1], C_b[1]) = lca(V_1^1, V_1^1) = a$

We show a tree constructed by the above definition in Fig. 5.

The node of this tree corresponds with a tuple of super factors which represents a generalization of the violated bottom case. We show such the corresponding lattice in Fig. 6 by computing lca for each element with C_b in the tuple of node in Fig. 5. Chosen a node N in Fig. 5, if we show all the cases in all the node N below (including N) are violated, then we obtain a tuple of super factors which shows a set of violated cases.

To compute the most general nodes from the bottom case C_b , we need the following definition.

Definition 9 Let C be a case. We define a set of cases $C \downarrow_{C_b}^l$ for $l(1 \leq l \leq m)$ as follows: $C' \in C \downarrow_{C_b}^l$ if C' satisfies the following condition:

- $parent(lca(C_b[l], C'[l])) = lca(C_b[l], C[l])$
- $lca(C_b[j], C[j]) = lca(C_b[j], C[j])$ for $j \neq l(1 \leq j \leq m)$.

Suppose that a node in the lattice be a set of cases $N = \{C_1, \dots, C_n\}$ then we define l -th direct child nodes from the node as follows: $\bigcup_{i=1}^n C_i \downarrow_{C_b}^l$ and denoted as $N \downarrow_{C_b}^l$.

Now we show an algorithm of generalizing the bottom case in terms of violation in Fig. 7. We call $OneStepGeneralization(C_b,)$ which returns maximal nodes of generalization.

3.3 Example of compliance check for decision making for giving child rearing allowance

For example, in Japan, there are two supports for a family with children and one is “child allowance” which supports all the families with a child under 15 years old and the other is “child rearing allowance” a single parent family with a child under 18 years old.

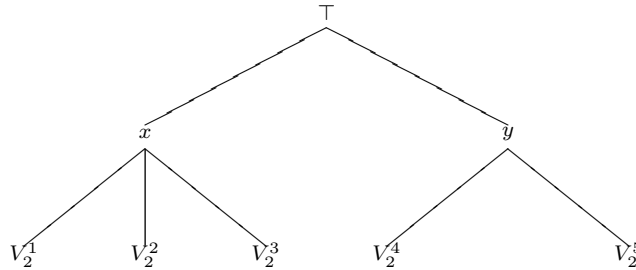


Figure 4: Tree Example2

The government decides acceptance of application of childrearing allowance from a parent of a child according to the following attribute. Please note that we omit other unrelated factors for the sake of simplicity.

- A parent receives other allowance such as disability pension, survivors pension or old age pension.
- A parent does not receive any pension.
- A child is a legitimate child or an illegitimate child.
- A illegitimate child is a recognized illegitimate child or an unrecognized illegitimate child.

In the original child rearing law says that if a parent does not receive any pension and a child is an unrecognized illegitimate child meaning that a child does not have a father legally but the father recognized the child as his own child.

There were two litigations related with the child rearing support. The first one was a litigation between Ms. A and the Japanese government ([14]) as follows: Ms. A is a visually impaired person categorized as Grade 1-Type 1 in the Schedule of the National Pension Law, and receives disability welfare pension under the said law. Ms. A has reared A, her son who was born to her and her common-law husband on May 12, 1955, by herself since her separation from her husband. On February 23, 1970, Ms. A applied to the Japanese Government for the recognition of the qualification to receive child rearing allowance under the Child Rearing Allowance Law. The government made a decision to reject the application as of March 23, 1970 on the grounds that Ms. A received disability welfare pension and therefore she was not qualified to receive child rearing allowance under Article 4(3)(iii) of the Child Rearing Allowance Law before amendment by Law No. 93 of 1973. The local court admitted her claim saying that the decision is against the Japanese constitution, but the higher courts rejected her claim. Nonetheless, after the local court decision, the government amended Article 4(3)(iii) of the Child Rearing Allowance Law to let people to receive both of supports.

The second one was a litigation between Ms. B and the Japanese government([15]) as follows: Ms. B conceived a child outside marriage, gave birth to it, took custody of it, and was paid child rearing allowance as a mother who has taken a child as provided by the Enforcement Order Article 1-2, subpara.3. However, the Japanese government made a decision to disqualify the child as eligible for child rearing allowance on October 27, 1993, on the ground that the child was recognised by the father on May 12, of the same year. It is because there is an exception note about Article 1-2, subpara. 3 saying that “except for a child recognised by the father”. In this litigation, the supreme court judged that this paragraph against the constitution and this exception has been deleted.

We would like to generalize these decision using our algorithm. We generalize this violation using the following factor-subfactor relations.

```

subfactor(receiving_pension_non_related,
  [no_pension,receiving_pension]).
subfactor(receiving_pension,
  [disability_pension,old_age_pension,survivors_pension]).
subfactor(child_anystatus,
  [legitimate_child,illegitimate_child]).
subfactor(illegitimate_child,
  [unrecognized_illegitimate_child,recognized_illegitimate_child]).

```

We show a tree for each factors of pension (Fig. 8) and child status (Fig. 9).

Suppose that an application from a single parent with a recognized illegitimate child below 18 who receives disability pension: `[disability_pension,recognized_illegitimate_child]` is rejected. Then according to the above judgements, the decision is against the constitution and we generalize this decision to avoid the further violation.

Then, starting from this case, we generalize to find another violated cases in a breadth first manner using the following lattice according to factor-subfactor relation for pension and child status. We abbreviate basic factors as follows: `disability_pension` as `disp`, `old_age_pension` as `oap`, `survivors_pension` as `surp`, `no_pension` as `nop`, `recognized_illegitimate_child` as `rec`, `unrecognized_illegitimate_child` as `unrec`, and `legitimate_child` as `leg`.

```

violate([disability_pension,recognized_illegitimate_child])
'Is 'decision([old_age_pension,recognized_illegitimate_child],reject)' violated?'
|: y.
'Is 'decision([survivors_pension,recognized_illegitimate_child],reject)' violated?'
|: y.
'Is 'decision([no_pension,recognized_illegitimate_child],reject)' violated?'
|: y.
'Is 'decision([no_pension,unrecognized_illegitimate_child],accept)' violated?'
|: n.
'Is 'decision([old_age_pension,unrecognized_illegitimate_child],reject)' violated?'
|: y.
'Is 'decision([survivors_pension,unrecognized_illegitimate_child],reject)' violated?'
|: y.
'Is 'decision([old_age_pension,legitimate_child],reject)' violated?'
|: n.
'Generalized Result'
generalized([receiving_pension_non_related,recognized_illegitimate_child])
generalized([receiving_pension,illegitimate_child])

```

In Fig. 18, we do not need to ask user about $\{\langle \text{oap}, \text{leg} \rangle, \langle \text{surp}, \text{leg} \rangle\}$ since we know that one of the lower node does not violate the constitution.

Then, we have two generalization from $\{\langle \text{disp}, \text{rec} \rangle\}$; $\{\langle \text{nop}, \text{rec} \rangle\}$ (which corresponds with $\{\langle \text{receiving_pension_non_related}, \text{rec} \rangle\}$ in terms of a super factor) and $\{\langle \text{oap}, \text{unrec} \rangle, \langle \text{surp}, \text{unrec} \rangle\}$ (which corresponds with $\{\langle \text{receiving_pension}, \text{illegitimate_child} \rangle\}$).

4 Conclusion

Our contribution in this paper are as follows:

- For the compliance check of the white box algorithmic laws, we propose a method of detecting violated situations of legal rules by using abductive logic programming proof procedure.
- For the compliance check of the black box algorithmic laws, we propose a method of generalizing of violated case by case-based learning technique.

As a future work, we need to solve the following limitations of the proposed methods.

- For the compliance check of the white box algorithmic laws, we make two assumptions:
 - We have rigorous software specification of algorithmic laws
 - We can explicitly present integrity constraints for violation.

However, if we only have a programming source of the algorithm, it would be difficult to get software specification. We need to develop a method of inverting the programming source into logical specification. If a compliance rule is very abstract such as “secure system”, we need to break down such abstract compliance rules into more concrete integrity constraints (such as “the password to enter the system should consist of at least one special symbols and should be more than seven characters”). We need to develop a method how to break down abstract compliance rules into more concrete integrity constraints. A kind of hybrid approach of combining various AI methods should be considered[1].

- For the compliance check of the black box algorithmic laws, we only consider yes/no questions for generalization of concepts. However to enhance applicability we need to extend our generalization for more complex conditions in a first-order language.
- After this research was finished in FFJ, a rapid emergence of generative AI systems has been made a great impact to the society. These generative AI systems could be used for advice in decision making but the systems are all black-box systems. Moreover these advice are not only yes/no questions but written in natural language sentences so it would become more difficult to assess the advice on whether it is against human rights. One possibility would be to develop another generative AI system to assess the advice.

References

- [1] Akata, Z. et al., “A Research Agenda for Hybrid Intelligence:Augmenting Human Intellect by Collaborative, Adaptive, Responsible and Explainable Artificial Intelligence”. *Computer* 53 (8), pp. 18-28 (2020). <https://doi.org/10.1109/MC.2020.2996587>
- [2] Chesani, F., Mello, P., De Mesellis, R., Di Francescomarino, C., Ghidini, C., Mnotali, M., Tessaris, S., “Compliance in Business Processes with Incomplete Information and Time Constraints: a General Framework based on Abductive Reasoning”, *Fundamenta Informaticae*, Vo. 161 pp. 75-111 (2018).
- [3] Contissa, G., Docter, K., Lagioia, F., Lippi, M., Micklitz, H.-W., Przemyslaw, P., Sartor, G., Torroni, P., “Claudette Meets GDPR: Automating the Evaluation of Privacy Policies Using Artificial Intelligence”, <https://ssrn.com/abstract=3208596> (2018).
- [4] De Vos, M., Kirrane, S., Padget, J., Satoh, K., “ODRL Policy Modelling and Compliance Checking”, *Proc. of RuleML+RR 2019*, LNCS 11784, pp. 36-51 (2019). https://doi.org/10.1007/978-3-030-31095-0_3
- [5] Governatori, G., Hashmi, M., Lam, H.-P., Villata, S., Palmirani, M., “Semantic Business Process Regulatory Compliance Checking using LegalRuleML”, . *Proc. of EKAW 2016*, LNCS 10024, pp. 746-761 (2016). https://doi.org/10.1007/978-3-319-49004-5_48
- [6] Sadri, F., Kowalski, R., “A Theorem-Proving Approach to Database Integrity, Foundations of Deductive Database and Logic Programming”, (J. Minker, Ed.), Morgan Kaufmann Publishers, pp. 313-362 (1988).
- [7] Satoh, K., “A Query Evaluation Method for Abductive Logic Programming”, the Proceedings of the Joint International Conference and Symposium on Logic Programming (JIC-SLP92), pp. 671-685 (1992).
- [8] Speicher, T., Ali, M., Venkatadri, G., Ribeiro, F.N., Arvanitakis, G., Benevenuto, F., Gum-madi, K.P., Loiseau, P., Mislove, M., “Potential for Discrimination in Online Targeted Advertising”, *Proceedings of Machine Learning Research* 81 (2018)
- [9] van Riemsdijk, M. B., Dennis, L. A., Fisher, M., Hindriks, K. V., “Agent Reasoning for Norm Compliance: a Semantic Approach”, *Proc. of AAMAS 2013*, pp. 499-506 (2013).
- [10] Verheij, B., “Artificial intelligence as law”, Presidential address to the seventeenth international conference on artificial intelligence and law. *Artificial Intelligence and Law* 28 (2), pp. 181-206 (2020). <https://doi.org/10.1007/s10506-020-09266-0>
- [11] <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>

- [12] Tribunal administratif Bordeaux (Bordeaux Administrative Court), req. No. 1504236, Jun. 16, 2016.
- [13] <https://www.impots.gouv.fr/portail/ouverture-des-donnees-publiques-de-la-dgfip>
- [14] “Judgment concerning whether or not Article 4(3)(iii) of the Child Rearing Allowance Law (before amendment by Law No. 93 of 1973) is in violation of Article 25 of the Constitution”, Case Number: 1976 (Gyo-Tsu) 30, 1982 July 7, Minshu Vol. 36, No. 7, at 1235. https://www.courts.go.jp/app/hanrei_en/detail?id=68
- [15] “Judgment upon the case concerning the legality of the part ”(except for a child recognised by the father)” of Article 1-2, para.3 of the Enforcement Order of the Law on the Child Rearing Allowance (before the amendment by the cabinet order No.224 of 1998)”, Case Number: 1996 (Gyo-Tsu) 42, 2002 January 1, Minshu Vol.56, No.1, at 246. https://www.courts.go.jp/app/hanrei_jp/detail2?id=62476 (in Japanese).

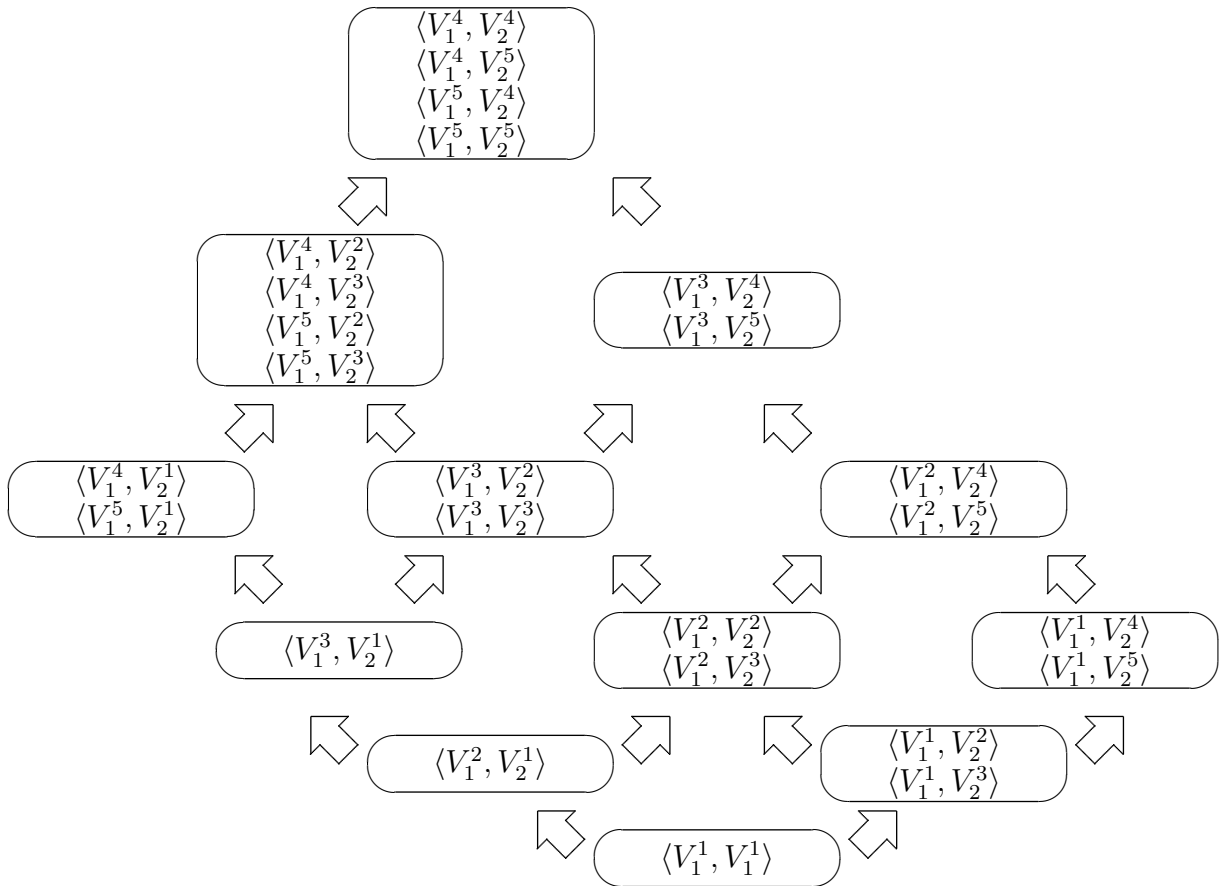


Figure 5: Lattice starting from $\{\langle V_1^1, V_2^1 \rangle\}$

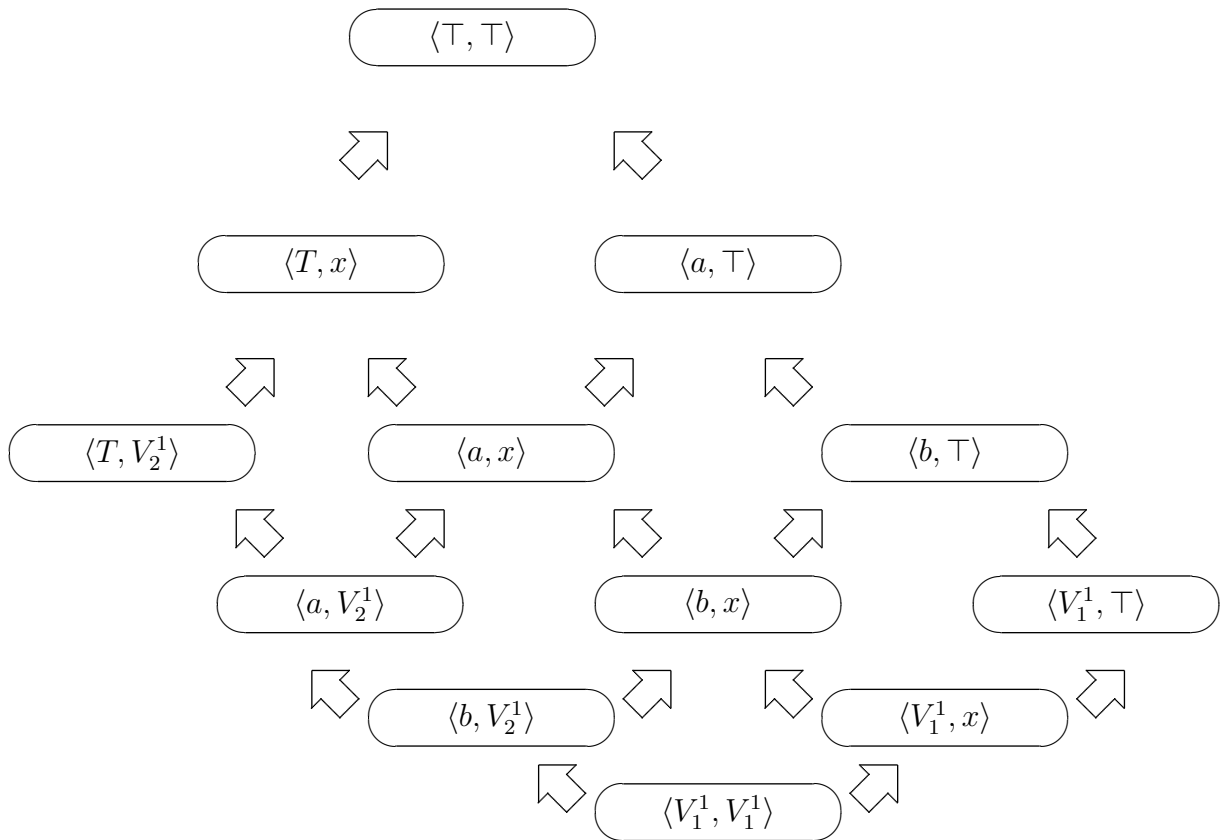


Figure 6: Factor Lattice starting from $\{\langle V_1^1, V_2^1 \rangle\}$

```

OneStepGeneralization(ListOfNodes, AlreadyGeneralizedNodes)
ListOfNodes: Set of set of cases
AlreadyGeneralizedNodes: Set of set of cases
begin
  if ListOfNodes =  $\emptyset$  return(maximal_element(AlreadyGeneralizedNodes))
  GeneralizedListOfNodes =  $\emptyset$ 
  for every  $N \in \textit{ListOfNodes}$  do
    begin
      OneUpperNodes :=  $\bigcup_{l=1}^m N \uparrow_{C_b}^l$ 
      OneGeneralizableNodes :=
         $\{N' \in \textit{OneUpperNodes} \mid \bigcup_{l=1}^m N' \downarrow_{C_b}^l \subseteq \textit{AlreadyGeneralizedNodes}\}$ 
      for every  $N'' \in \textit{OneGeneralizableNodes}$  do
        begin
          for every  $C \in N''$  do
            begin
              ask a user whether  $C$  is violated.
            end
          if every  $C \in N''$  is violated then
            GeneralizedListOfNodes := GeneralizedListOfNodes  $\cup$   $\{N''\}$ 
          end
        AlreadyGeneralizedNodes := AlreadyGeneralizedNodes  $\cup$  GeneralizedListOfNodes
      return OneStepGeneralization(GeneralizedListOfNodes, AlreadyGeneralizedNodes)
    end
  end

```

Figure 7: Generalizing Violated Case

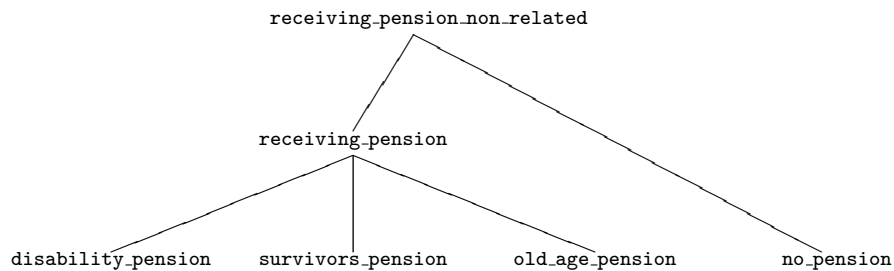


Figure 8: Factor-Subfactor Relation (pension)

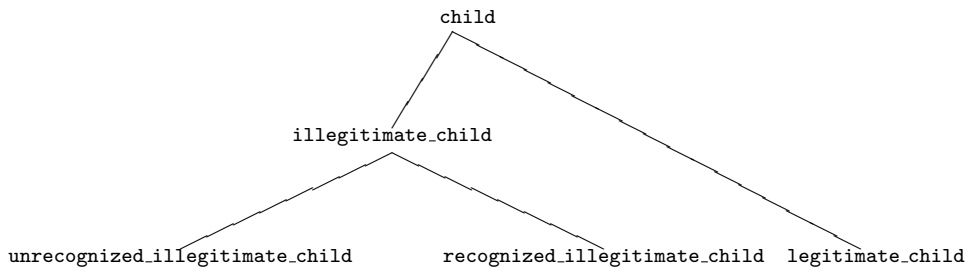


Figure 9: Factor-Subfactor Relation (child status)

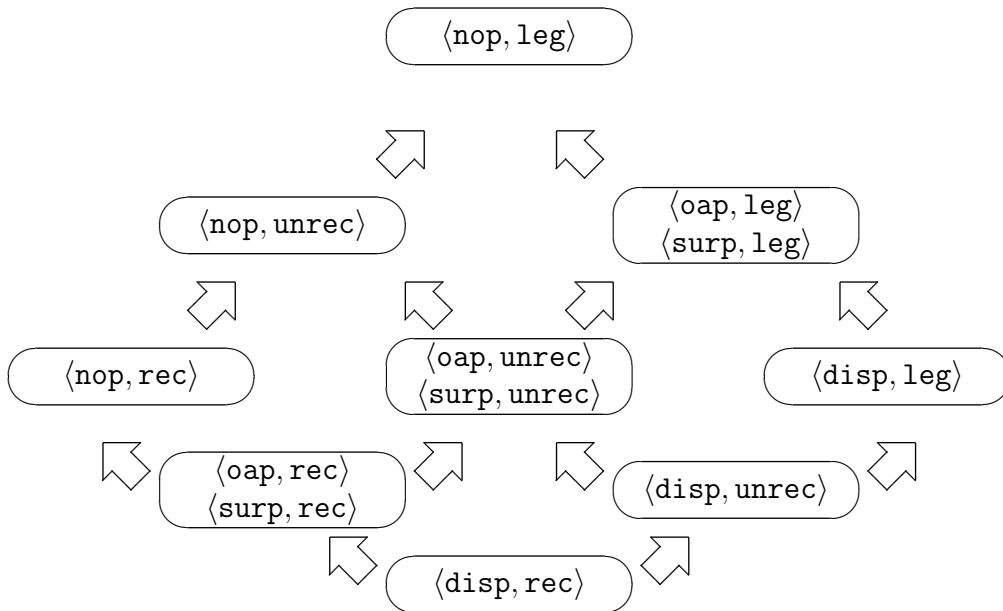


Figure 10: Lattice starting from $\{\langle \text{disp}, \text{rec} \rangle\}$

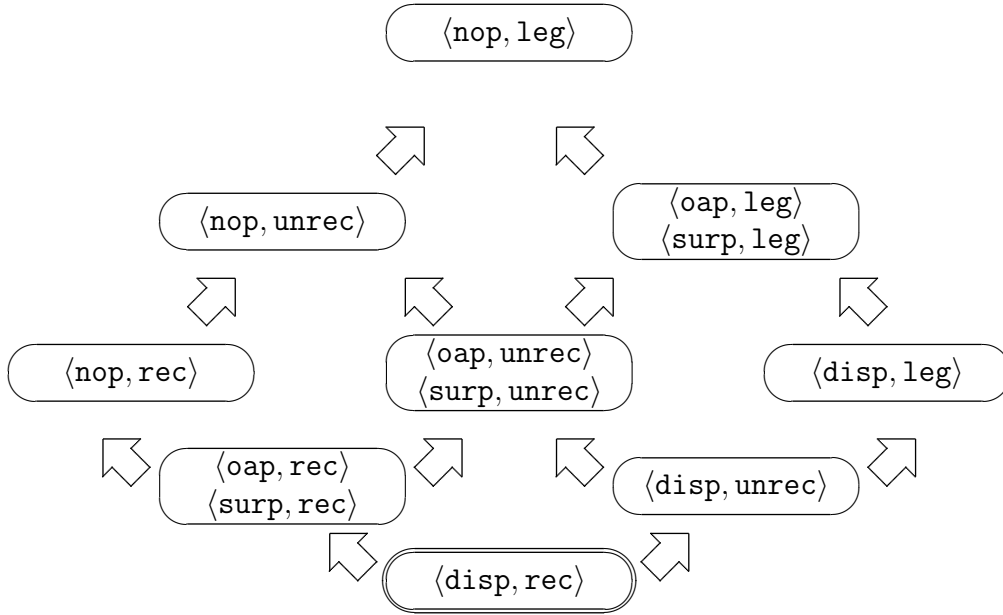


Figure 11: Execution of Generalization (1)

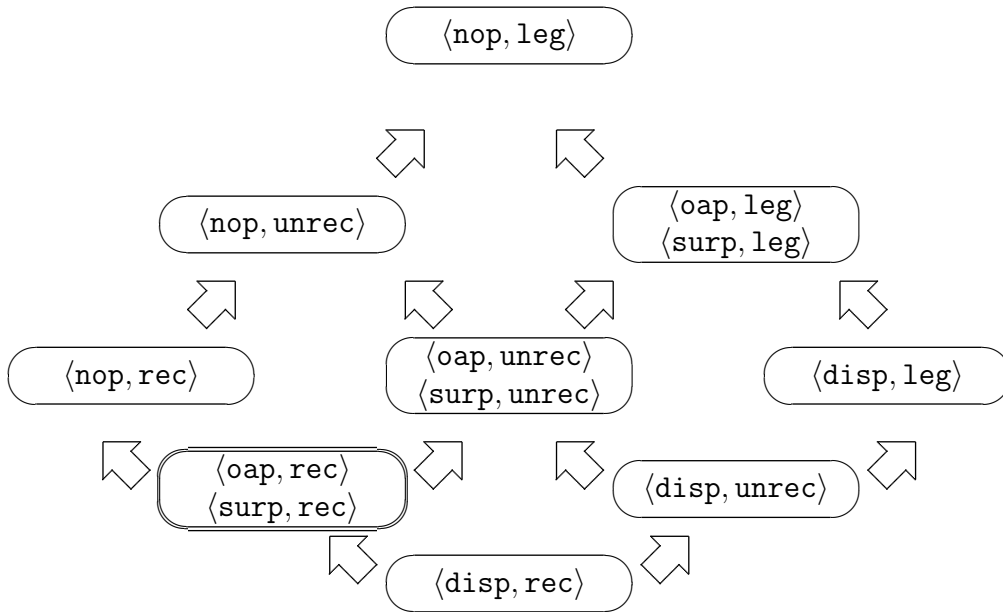


Figure 12: Execution of Generalization (2)

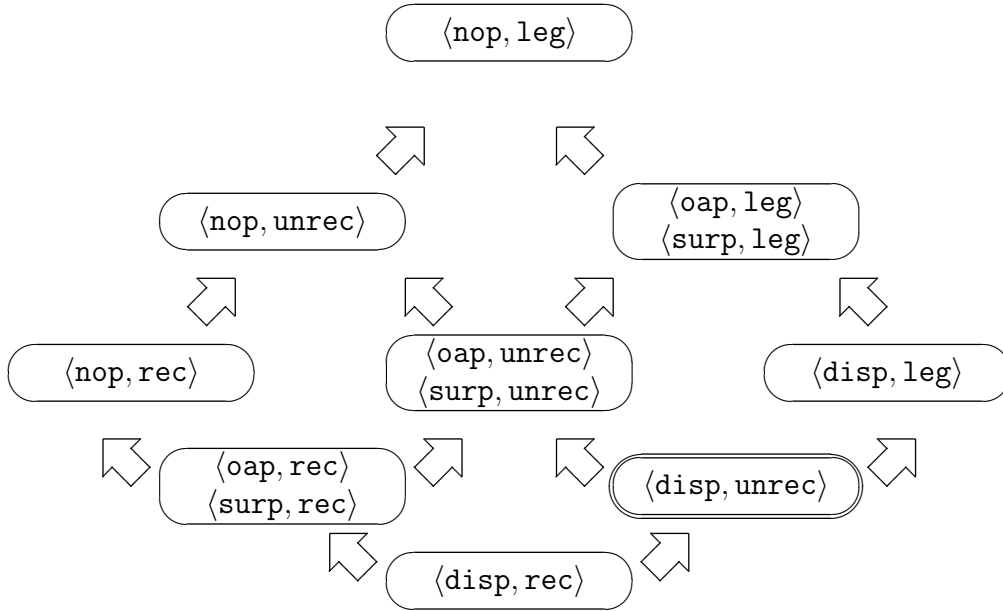


Figure 13: Execution of Generalization (3)

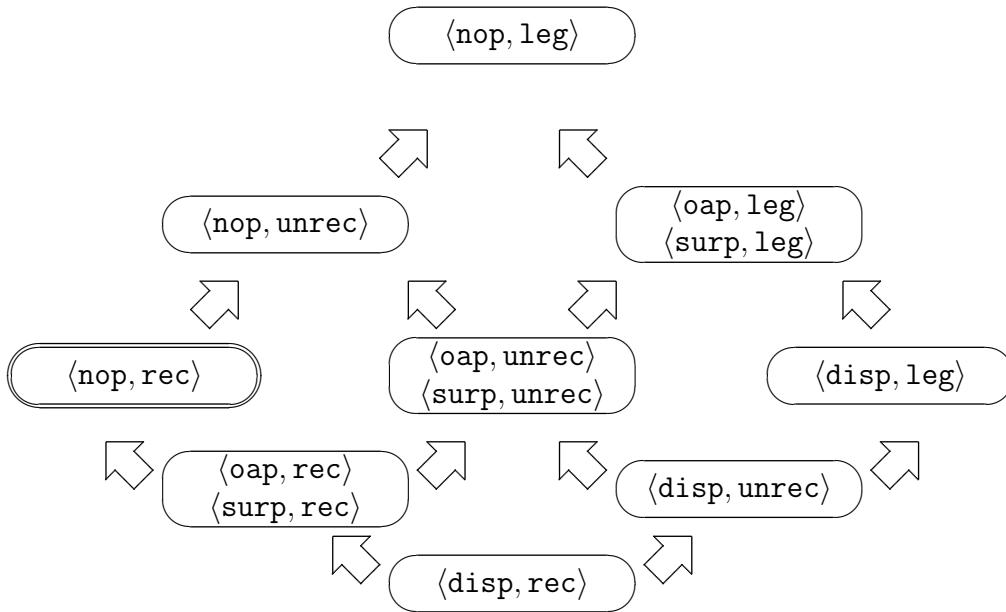


Figure 14: Execution of Generalization (4)

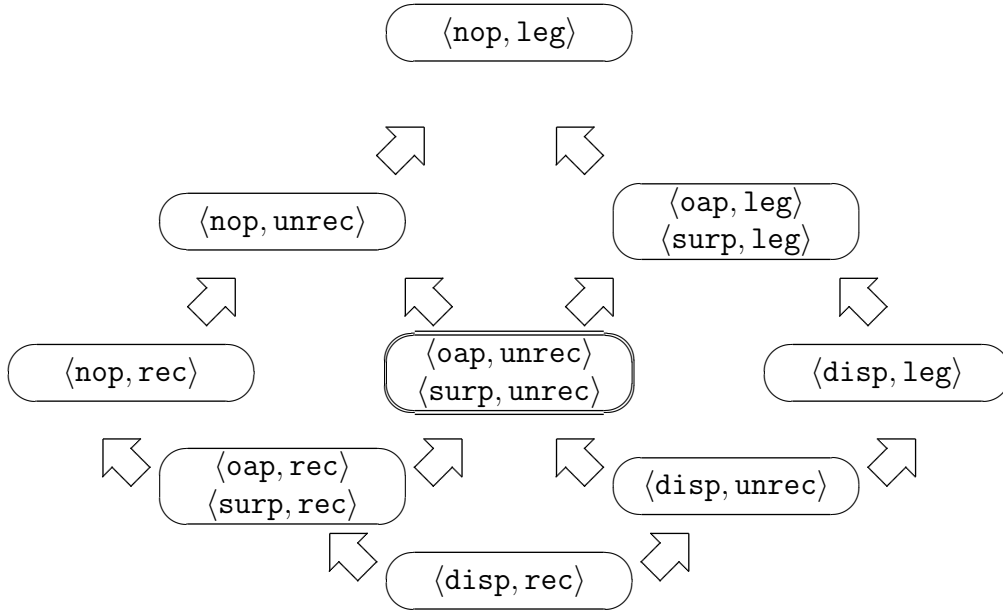


Figure 15: Execution of Generalization (5)

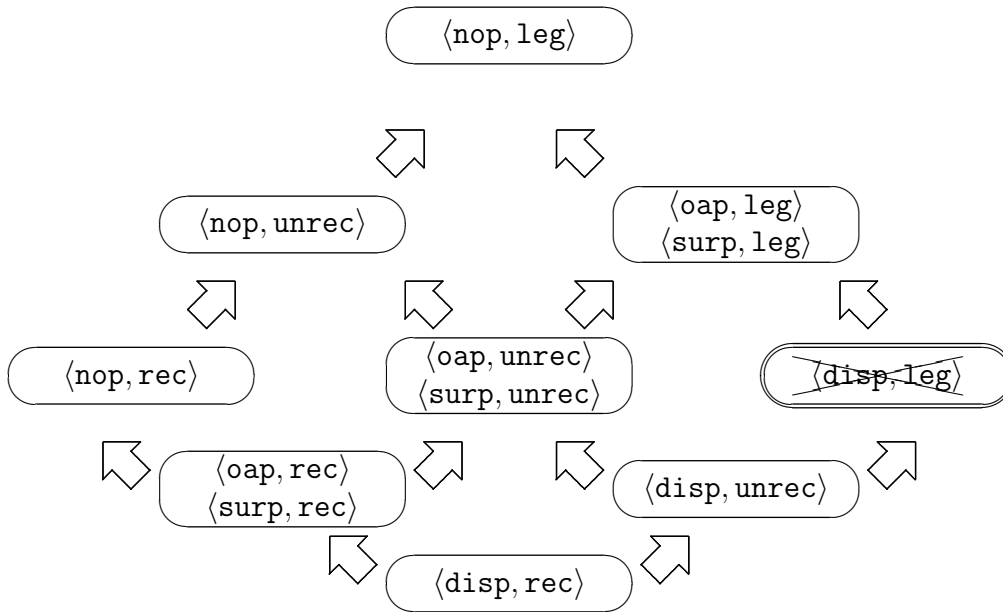


Figure 16: Execution of Generalization (6)

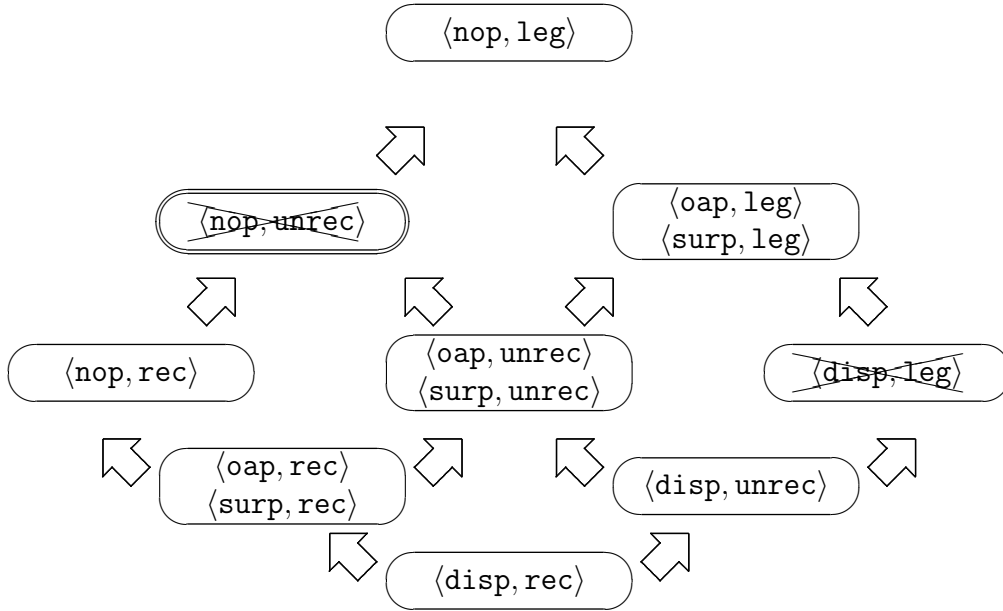


Figure 17: Execution of Generalization (7)

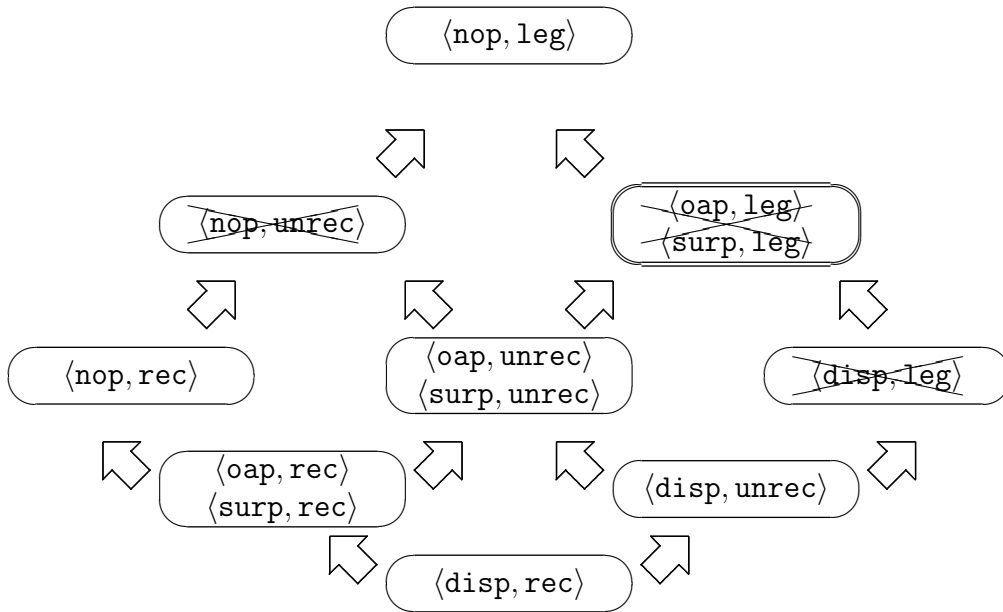


Figure 18: Execution of Generalization (8)

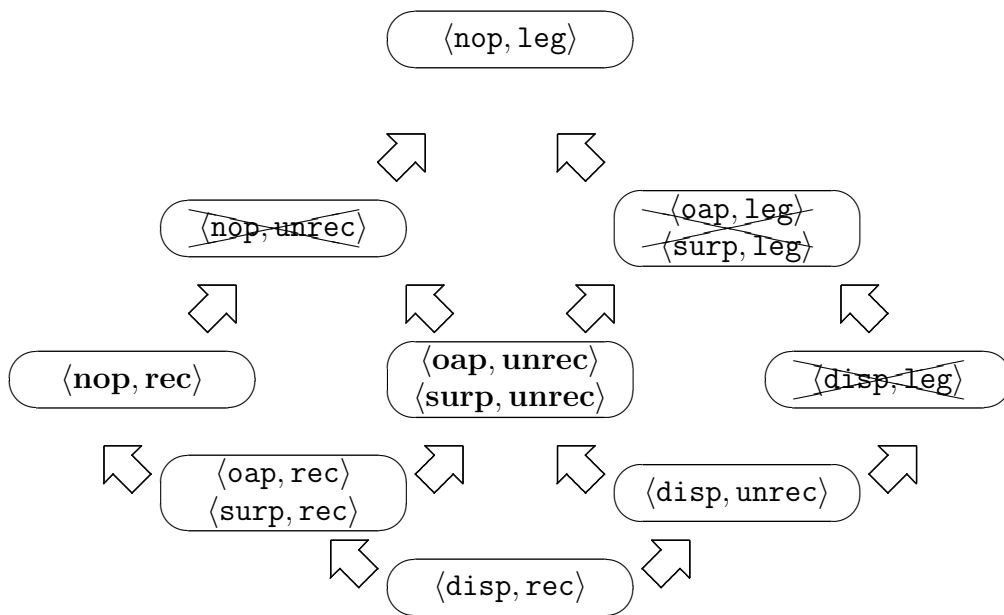


Figure 19: Result of Generalization

Previous FFJ Discussion Papers

DP 23-03 (October 2023)

“The Role of Art for the Future of Society”, Koichiro ETO (National Institute of Advanced Industrial Science and Technology), 2021 FFJ/Air Liquide Fellow

DP 23-03 (July 2023)

“ROBOTS IN THE WILD: An Ethnography of Robot-Human Interactions outside the Laboratory”, Fabio GYGI (EHESS, SOAS), 2022 FFJ/Air Liquide Fellow

DP 23-02 (June 2023)

“Political Economy shaped by Financialization: Impacts on Monetary Policy and Foreign Economic Policy of Japan”, Saori KATADA (University of Southern California), 2022 FFJ/Banque de France Fellow

DP 23-01 (March 2023)

“Public spaces of mobility in Paris, Tokyo, and Buenos Aires”, Andrés Borthagaray (Furban / City on the Move), 2022 FFJ/Michelin Foundation Fellow

DP 22-07 (October 2022)

“A Study on “Jurisdiction” and Consistency in Urban Policy: Crossed perspectives Japan-France”, Mari Uchiumi (Komazawa University), 2022 FFJ Visiting Researcher

DP 22-06 (September 2022)

“Monetary, fiscal and demographic interactions in Japan: impact and a comparative assessment”, Pierre L. Siklos (Wilfrid Laurier University and Balsillie School of International Affairs), 2021 FFJ/Banque de France Fellow

DP 22-04 (July 2022)

“Diffusion of Connectivity Technology and development of new business models: How Connectivity Technologies redefine the role of human within organization? Empirical learnings from Japanese mobility service providers”, Haruki Sawamura (École Polytechnique), 2019 FFJ/Valeo Fellow

DP 22-03 (July 2022)

“Bibliometrics analysis of mHealth and AI to discover the concept of AI in the field of mHealth”, Kota Kodama (Ritsumeikan University), 2020 FFJ/Air Liquide Fellow

DP 22-02 (May 2022)

“Cycling for all? A feminist analysis of the Tokyo Bicycle Utilisation Promotion strategy”, Marion Lagadic (University of Oxford), 2021 FFJ/Michelin Foundation Fellow

DP 22-01 (Mars 2022)

“Potentials and Challenges of The Connected Autonomous Shared Electric Vehicle (CASE) from Urban Geography Perspective in Southeast Asia Mega-Urban Regions”, Kulacha Sirikhan (The University of Tokyo), 2021 FFJ/Valeo Fellow

DP 21-05 (December 2021)

“East Asian and European Firms: Comrades or Competitors”, Willem Thorbecke (Research Institute of Economy, Trade and Industry), 2020 FFJ/Banque de France Fellow

DP 21-04 (December 2021)

“The Mobility of Paris Residents and Retailers: Their Viewpoints on the Effects of the City’s Pedestrianization”, Meriç Kirmizi (Ondokuz Mayıs University), 2020 FFJ/Michelin Foundation Fellow

DP 21-03 (August 2021)

“City logistics for sustainable and liveable cities”, Eiichi Taniguchi (Kyoto University), 2020 FFJ/Michelin Foundation Fellow

DP 21-02 (May 2021)

“The impact of Paris 2024 on the construction of the Grand Paris Express: a hidden extra cost of the Olympic”, Alexandre Faure (Fondation France-Japon de l’EHESS, EHESS)

DP 21-01 (March 2021)

“Local Government and Innovation for Sustainable

mobility”, Soichiro Minami (Policy Research Institute for Land, Infrastructure, Transport and Tourism), 2018 FFJ/Valeo Fellow

DP 20-05 (Septembre 2020)

“International Spillover Effects of Unconventional Monetary Policies of Major Central Banks”, Tomoo Inoue (Seikei University), Tatsuyoshi Okimoto (Australian National University), 2019 FFJ/Banque de France Fellow

DP 20-04 (July 2020)

“Characteristics of Bike taxis in African rural society: A case study of Dschang, West Cameroon”, Makiko Sakai (Tokyo University of Foreign Studies), FFJ Visiting Researcher

DP 20-03 (June 2020)

“Decomposing Preference for Redistribution Beyond the Trans-Atlantic Perspective”, Ryo Kambayashi (Hitotsubashi University), Sébastien Lechevalier (EHESS, CCJ-FFJ), Thanasak Jenmana (PSE)

DP 20-02 (April 2020)

“A Comparative Analysis of Autonomous Vehicle Policies among Korea, Japan, and France”, Jeehoon Ki (KISTEP), 2018 FFJ/Renault Fellow

DP 20-01 (March 2020)

“Comparing the Development and Commercialization of Care Robots in the European Union and Japan”, James Wright (Sheffield University), 2019 FFJ/Michelin Fellow

DP 19-05 (November 2019)

“Credit Guarantees and Zombie Firms: A Case Study of Small and Medium-Sized Enterprises in Japan”, Scott Wilbur (Yale University), 2018 FFJ/Banque de France Fellow

DP 19-04 (October 2019)

“Impact of R&D Activities on Pricing Behaviors with Product Turnover”, Yasushi Hara (Hitotsubashi University), 2018 FFJ/Michelin Fellow, Akiyuki Tonogi (Toyo University) & Konomi Tonogi (Rissho University)

DP 19-03 (September 2019)

“From Agribusiness to Food Democracy. Comparative Study on Agricultural Policy and Organic Farming in France and in Japan”, Shoichiro Takezawa (National Museum of Ethnology, Japan); 2018 FFJ/Air Liquide Fellow

DP 19-02 (June 2019)

“Legitimation of Income Inequality in Japan: A Comparison with South Korea and the United States”, Shin Arita (The University of Tokyo), Kikuko Nagayoshi (Tohoku University), Hirofumi Taki (Hosei University), Hiroshi Kanbayashi (Tohoku Gakuin University), Hirohisa Takenoshita (Keio University) and Takashi Yoshida (Shizuoka University); Prizewinners of the 2019 FFJ/SASE Best Paper Award.

DP 19-01 (May 2019)

“A Study on New Mobility Services and Sustainable Urban Development”, Ai Nishimura, 2017 FFJ/Renault Fellow

DP 18-06 (December 2018)

“A Study of New Technologies of Personal Mobility and Robot Suit for the Elderly and Persons with Disabilities”, Toshiaki Tanaka (University of Tokyo, Hokkaido University of Science), 2017 FFJ/Valeo Fellow

DP 18-05 (November 2018)

“Developments in Well-Being at Work in Japan: A Survey and a Comparison with France”, Louise Baudrand (EHESS), César Castellvi (EHESS), Nao Kinoshita (EHESS), Adrienne Sala (Sciences Po Lyon) & Sébastien Lechevalier (EHESS, Fondation France-Japon de l'EHESS)

DP 18-04 (November 2018)

“Understanding AI Driven Innovation by Linked Database of Scientific Articles and Patents”, Kazuyuki Motohashi (University of Tokyo, NISTEP and RIETI, 2017 CEAFJP/Michelin Fellow)

DP 18-03 (November 2018)

“The Yen Exchange Rate and the Hollowing-out of the Japanese Industry”, Ansgar Belke (University of

Duisburg-Essen) & Ulrich Volz (SOAS University of London, 2017 CEAFJP/Banque de France Fellow)

DP 18-02 (October 2018)

“Cross-cultural (France and Japan) and Multidisciplinary Discussion on Artificial Intelligence and Robotics: Tendencies and Research Prospects”, Naoko Abe (CEAFJP Research Fellow)

DP 18-01 (July 2018)

“Impact of Shareholder-Value Pursuit on Labor Policies at Japanese Joint-Stock Companies: Case of Nikkei Index 400”, Kostiantyn Ovsiannikov (University of Tsukuba, Prizewinner of the 2018 FFJ/SASE Best Paper Award)

DP 17-05 (November 2017)

“Female Board of Directors and Organisational Diversity in Japan”, Yukie Saito (CEAFJP Associate Researcher, University of Geneva, Institut de Recherches Sociologiques)

DP 17-04 (August 2017)

“*Keiretsu* Divergence in the Japanese Automotive Industry: Why Have Some, but Not All, Gone?”, Akira Takeishi (Graduate School of Economics, Kyoto University; CEAFJP Visiting Researcher) et Yoshihisa Noro (Mitsubishi Research Institute, Inc.)

DP 17-03 (June 2017)

“Globalization and Support for Unemployment Spending in Asia: Do Asian Citizens Want to Embed Liberalism?”, Sijeong Lim (University of Amsterdam) et Brian Burgoon (University of Amsterdam) ; Prizewinners of the SASE/FFJ Best Paper Award.

DP 17-02 (April 2017)

“Does ‘Driving Range’ Really Matter? The Hidden Cost of Internal Combustion Engine Vehicles and the Potentially New Value Proposition of Electric Vehicles: Two Cases from Countryside and Urban Side of Japan”, Hidetada Higashi (2016 CEAFJP/Valeo Fellow)

DP 17-01 (March 2017)

“How Can We Understand the Differences between France and Japan in the Growth of Shared Mobility

Services? The Paradox of Trust and its Social Construction”, Naoko Abe (2016 CEAFJP/Renault Fellow)

DP 16-03 (September 2016)

“Parameter Bias in an Estimated DSGE Model: Does Nonlinearity Matter?”, Yasuo Hirose (Keio University) and Takeki Sunakawa (University of Tokyo)

DP 16-02 (April 2016)

“Financialization and Industrial Policies in Japan and Korea: Evolving Complementarities and Loss of Institutional Capabilities”, Sébastien Lechevalier (EHESS), Pauline Debanes (EHESS), and Wonkyu Shin (Kyung Hee University)

DP 16-01 (April 2016)

“How Do Credit Hours Assure the Quality of Higher Education? Time-Based vs. Competency-Based Debate”, Ayaka Noda (National Institution for Academic Degrees and Quality Enhancement of Higher Education (NIAD-QE)

DP 15-04 (December 2015)

“Government Policy and the Evolution of Japan’s Photovoltaic Industry, 1961-2014”, Maki Umemura (Cardiff University, 2015 CEAFJP/Michelin Fellow)

DP 15-03 (December 2015)

“Japan’s Financial Crisis and Lost Decades”, Naohisa Hirakata (Bank of Japan), Nao Sudo (Bank of Japan), Ikuo Takei (Bank of Japan), Kozo Ueda (Waseda University, 2015 CEAFJP/Banque de France Fellow)

DP 15-02 (May 2015)

“Can Increased Public Expenditure Efficiency Contribute to the Consolidation of Public Finances in Japan?”, Brieuc Monfort (CEAFJP Associate Researcher)

DP 15-01 (May 2015)

“Policy Regime Change Against Chronic Deflation?
Policy Option under a Long-Term Liquidity Trap”,
Ippei Fujiwara (RIETI, Keio University, Australian
National University), Yoshiyuki Nakazono (Yokohama
City University), Kozo Ueda (Waseda University,
2014 CEAFJP/Banque de France Fellow)