Juris-Informatics and PROlog-based LEGal reasoning system: PROLEG

Ken Satoh
National Institute of Informatics and Sokendai

1. Self Introduction
2. Current Trend of AI application to law domain
3. Our main activity:Juris-informatics
   (a) COLIEE
   (b) PROLEG: (PROlog based LEGal reasoning support system)
4. Conclusion

My Background

- I have been working on logical foundations of AI under incomplete information called “non-monotonic reasoning”.
- To seek the application of my work, I entered the law school in University of Tokyo and learned law in 2006-2009.
- During learning at the law school, I found that the theory of judgement in civil litigation called “the ultimate fact theory” taught in the law school is closely related with my work.
- I developed a programming language called PROLEG (PROlog based LEGal reasoning support system) based on the above idea aimed at assisting law students and lawyers.
- As a by-product of the research, I passed the bar exam in Japan in 2017.
- I also started workshops on “juris-informatics”(JURISIN) from 2007.

Current Trend of AI application to law domain

Currently, there are several commercial products related with Law domain using AI technology and there are some usage of AI technique in the court

- E-discovery Support in US
- Relevant Document Retrieval
- Checking Contracts
- Patent Troll Detection
- Predicting Financial Ombudsman Decision
- Document Classification System
- Correctional Offender Management Profiling for Alternative Sanctions (COMPAS)
- Usage of Bayesian Network in Criminal Court

This survey was done with the help of Kevin Ashley.

E-discovery support in US

In US, there is an important process of fact finding called e-discovery. Rule 26(f): At the parties: planning meeting, issues expected to be discussed include:

- Any issues relating to disclosure or discovery of *electronically stored information*, including the form or forms in which it should be produced
- Any issues relating to preserving discoverable information

Example of E-discovery(U.S. v. Philip Morris et al.)

“Using Visual Analytics in E-Discovery and E-Disclosure Cases” by Jason Baron
VASS 2012: Second UKVAC International Visual Analytics Summer School

- Civil lawsuit brought by Clinton Administration against tobacco companies in 1999.
- Claimed that companies have conspired since 1953 to defraud the American public as to the true health effects of smoking
- During litigation, 1,726 Requests to Produce from tobacco companies for tobacco-related records (including email) from 30 federal agencies

Example of E-discovery(U.S. v. Philip Morris et al.)

- 32 million Clinton-era email records held by National Archives are investigated.
- 25 people made information retrieval task for 6 month to give 80000 relevant emails to Philip Morris et al.

Example of Retrieval Command:
(((master settlement agreement OR msa) AND NOT (medical savings account OR metropolitan standard area)) OR s. 1415 OR (ets AND NOT educational testing service) OR (liggett AND NOT sharon a. liggett) OR atco OR lorillard OR (pmi AND NOT presidential management intern) OR pm usa OR rjr OR (b&w AND NOT photo*) OR phillip morris OR batco OR ftc test method OR star scientific OR vector group OR joe camel OR (marlboro AND NOT upper marlboro)) AND NOT (tobacco* OR cigarette* OR smoking OR tar OR nicotine OR smokeless OR synar amendment OR philip morris OR r.j. reynolds OR (”brown and williamson”) OR (”brown & williamson”) OR bat industries OR liggett group)

E-discovery Support Tool

- UBIC has been providing support for e-discovery since 2009.
- Moreover, the international competition of E-discovery (TREC Legal Track, http://trec-legal.umiacs.umd.edu/) by machine learning shows that AI provides better retrieval command than human.

Relevant Document Retrieval

- ROSS Intelligence developed document retrieval system using IBM Watson program to replace legal assistants by computer program.
- ROSS made a contract with Baker & Hostetler, which is the large law firm in US in 2016.
- ROSS program saves the Baker and Hostetler firm a lot of time and money that they would have spent on human paralegals and legal researchers.

Checking Contracts

- LawGeex provides a system which reviews articles in contract documents and points out problematic articles.
- `https://blog.lawgeex.com/ai-more-accurate-than-lawyers/#more-4058` says that they compete with lawyers about Non-Disclosure Agreements (NDAs) review and the system give 94% accuracy of problematic article detection wheres 20 lawyers' average of correctness is 86% on average.

Checking Contract: Kira System(continued)

`https://info.kirasystems.com/news/`
`kira-systems-receives-50-million-`
investment-insight-venture-partners—

- Kira recived 50 million US dollars investment from venture capital in 2018.

- Kira system classifies each item in a contract according to their kind and purpose.
- Kira system also provides a suggestion of improvement of each item.

Patent Troll Detection

- LexisNexis, which is the largest legal database company in the world, bought a company called LexMachina in 2017.
- LexMachina analyses patent litigations whether they are invoked by patent troll company (to enforce patent rights against accused infringers far beyond the patent's actual value).
- LexMachina program gives 85 percent accuracy to find patent troll and it may be useful for a private company to prepare for fighting with such patent trolls.

Predicting Financial Ombudsman Decision

- `http://www.bbc.co.uk/news/technology-41829534` says that Case-Cruch company made AI program to predict whether the Financial Ombudsman would allow a claim.
- The humans and the AI were given the basic facts of hundreds of PPI (payment protection insurance) mis-selling cases and asked to predict whether the Financial Ombudsman would allow a claim.
- They submitted 775 predictions and the computer won hands down, with Case Cruncher getting an accuracy rate of 86.6%, compared with 66.3% for the lawyers.

Document Classification System: RAVN

`https://www.wired.co.uk/article/ ravn-artificial-intelligence`

- The Serious Fraud Office investingates corruption at Rolls-Royce.
- There are over 30 million documents seeked as evidences.
- However, they need to classify documents into "privileged"/"non-privileged".
- RAVN system classifies 600,000 documents daily where as in the previous work by the barristers, they only processed 3,000 documents a day.
- RAVN system cut out 80 per cent of the work.

Correctional Offender Management Profiling for Alternative Sanctions (COMPAS)

A Popular Algorithm Is No Better at Predicting Crimes Than Random People `https://www.theatlantic.com/technology/archive/2018`

- The COMPAS system predicts defendants' possibility of comitting crime again and is used for sentencing decision.
- The first system was created in 1998 in USE and has been used.
- Recently, COMPAS is no better at predicting an individual's risk of recidivism than random volunteers recruited from the internet. Julia Dressel and Hany Farid, "The accuracy, fairness, and limits of predicting recidivism" `Science Advances, Vol. 4, no. 1, DOI: 10.` (2018)

Usage of Bayesian Network in Criminal Court

Henri Prakken, “Argument schemes for discussing Bayesian modellings of complex criminal cases” Proc. of JURIX 2017, pp. 69 - 78 (2017).

- Use of likelihood ratios in court to report on the evidential value of single forensic pieces of evidence is quite common in the Netherlands.
- The paper analyses that in one case using Bayesian Network, even a seemingly expert on Bayesian Network (climate physicist) made a mistake of evidential reasoning.

Introduction: JURIS-INFORMATICS

Ultimate purpose of the research:
Applying Informatics/AI techniques to legal domain to establish new research area I would like to call *juris-informatics*.

- Currently, AI and Law community seems to narrow their interests.
- So, we would like to extend the scope of research between informatics and law by putting a new name for this research.
- We would like to simulate the success of bio-informatics by giving impact to law domain using IT technology.

Background:

- Rapid society change (especially related with IT development)
- Processing large legal data
- (In Japan), slow litigation process (especially in intelligent property litigation)

Our Main Activities of Juris-Informatics

- I developed a language called PROLEG (PROlog based LEGal reasoning support system) to simulate judge's judgements in civil litigation.
- I have been a steering committee member of workshops on juris-informatics (JURISIN). We have held 10 JURISIN workshops in 2007-2016. We will have JURISIN 2017 in November 2017 as well.
- We set up a competition on this domain:
  Competition on Legal Information Extraction/Entailment
  (previous one:COLIEE-17) in London
  `http://webdocs.cs.ualberta.ca/~miyoung2/COLIEE2017`
  with Randy Goebel and Mi-Young Kim from University of Alberta.

COLIEE tasks: statute law

Retrieval Task of Relevant Articles in Statute Law

Given a question Q and the entire Civil Code Articles, we have to retrieve a relevant set of ”S1, S2, ..., Sn” in Civil Code Articles which is useful for the entailment task.

Entailment Task between Articles and a Question in Statute La

Given a question Q and relevant articles S1, S2, ..., Sn, we have to determine if the relevant articles entail/does not ”Q”.

COLIEE tasks: Statute Law(continued)

2011 Q 3(1): The counterparty can rescind a contract by an agent without authority of agency of the principal if the counterparty does not know the ratification of the contract by the principal even after the ratification is made to the agent without authority.

Article 113(Unauthorized Agency)
1. Any contract concluded by a person who holds himself/herself out as an agent of others without authority of agency shall be void vis-a-vis the principal unless ratified by the principal.
2. Any ratification or refusal to ratify may not be asserted vis-a-vis the counterparty unless it is made to such counterparty; provided, however, that, this shall not apply to the cases where the counterparty has come to know such fact.

new COLIEE tasks: case law from 2018

Retrieval Task of Relevant Articles in Law

Given a question Q and the entire Civil Code Articles, we have to retrieve a relevant set of ”S1, S2, ..., Sn” in Civil Code Articles which is useful for the entailment task.

Entailment Task between Articles and a Question in Statute La

Given a question Q and relevant articles S1, S2, ..., Sn, we have to determine if the relevant articles entail/does not ”Q”.

Our Other Activities of Juris-Informatics

- **Formalization of Causality based on Abduction** (with Satoshi Tojo, JAIST, Japan)
  "Disjunction of Causes and Disjunctive Cause: a Solution to the Paradox of 'Conditio Sine Qua Non' using Minimal Abduction", Proc. of JURIX 2006.
- **Detection and Resolution of Conflicts in Legal Systems using Inductive Logic Programming** (with Marina De Vos, Julian Padget, Tingting Li, University of Bath, UK)
  Conflict between the current legal system and new rules
  Conflict between legal systems in various countries
- **Learning Legal Reasoning Structure from Precedent Cases**
  (with Duangtida Athakravi, Alessandra Russo, Krysia Broda, Imperial College, UK)

PROLEG: (PROlog based LEGal reasoning support system)

1. Reasoning Steps in Civil Code Litigation
2. The Japanese Presupposed Ultimate Fact Theory
3. PROLEG
4. Generality of PROLEG

Reasoning Steps in Civil Code Litigation

1. Fact Finding Phase
2. Subsumption Phase
3. Judgement Phase

Fact Finding Phase

- A judge decides the truth value of facts in the considered case. Facts should contribute to judgement using evidence.
- Evidence itself is also a fact so we need to decide the truth value of evidence.
- We need evidential reasoning as well as reasoning from basic facts.
- Note that the truth value might be undefined (non-liquet).

For example, suppose a judge would like to decide whether there was an agreement about purchase of goods or not. If a plaintiff gave some basic facts such as email exchanges which can derive the fact of agreement, a judge will take into account of this base fact to decide whether there was an agreement.

Subsumption Phase

- A judge makes a correspondence between specific facts and legal concepts.
- The truth value of specific facts is reflected to the truth value of concrete legal concepts.

For example, suppose a judge decides the truth value of agreement for the purchase of goods is true. Then, this agreement corresponds with “establishment of contract” in a legal term so the truth value of establishment of contract is true.

Judgement Phase

- According to the truth value, a judge decides which legal rights or obligation exists.
- However, the truth value is three valued so some judgement could become “undefined” in a deductive logic and in this case, a judge cannot make a deductively correct decision.
- In litigation, a judge cannot allow to give a judgement of “I do not know” and is forced to make a decision.

To solve this undefined situation, **the Japanese Presupposed Ultimate Fact Theory** is invented.

JUF theory

- **The Japanese Presupposed Ultimate Fact Theory** (called "*Yoken-jijitsu-ron*" in Japanese, **JUF theory**, for short, in this presentation), has been mainly developed by judges in the Japanese Legal Training Institute.
- This is an interpretation of civil law from the judge's view in order to handle the uncertainty in the court because of a lack of enough evidence.
- This theory has not been attracted by university scholars in law departments in Japan for years, but thanks to the Japanese Judicial System Reform, this theory started to be taught in law schools and interest in this theory has grown.

Introduction: JUF Theory(continued)

- When I learned this theory, it surprised me very much that judges made such a rigorous framework without knowing mathematical formalizations.
- We propose a representation of legal knowledge which is more friendly of JUF theory to lawyers than a logic program and simultaneously executable by a meta-interpreter written in PROLOG.

Note about the JUF theory

- The JUF theory is a theory of determination of conclusion used by judges under incomplete information **after both parties finish their activity in the proceedings**, in other words it is used for **judgement phase**.
- The JUF theory divides conditions of legal rules into two categories; for one of which a plaintiff has a burden of persuasion and the other of which a defendant has a burden.
- **The JUF theory is not a theory of argumentation**, but both parties should obey the JUF theory to win the case, and the activities of both parties resembles an argumentation.
- Our (maybe too exaggerated) claim: **the activities at court are not argumentation, but persuasion activities toward judges (at least in Japan).**

Working Example

- A plaintiff made a lease contract for his house between him and the defendant.
- The defendant let his girl friend use a room for a piano lesson.
- The plaintiff claimed the contract was ended by his cancellation of the contract because of the defendant's sublease without permission.
- In turn, the defendant attacked the plaintiff's claim that his subleasing a room to her does not cause any destructin of confidence with the plaintiff because the time of use per day was very short.
- In turn, the plaintiff attacked the defendant's counter-argument that the defendant abused the confidence with the plaintiff because there were neighbors' complaints about noise from piano lessons during subleasing.

Working Example(continued)

To handle the above case, we take into account the Japanese Civil Code Article 612,

- Phrase 1 states that a lessee may not assign the lessee's rights or sublease a leased thing without obtaining the approval of the lessor, and
- Phrase 2 states that if the lessee allows any third party to make use of or take profits from a leased thing in violation of the provisions of the preceding paragraph, the lessor may cancel the contract.

However, according to the Supreme Court case rule, Phrase 2 is not applicable in exceptional situations where the sublease does not harm the confidence between a lessee and a lessor, and therefore the lessor cannot cancel the contract unless they prove the lessee's destructin of confidence.

Reasoning by the JUF Theory for Working Example

According to the JUF theory,

- To get a desired conclusion, based on Phrase 2, the plaintiff must firstly show that:
  - The contract with the defendant was established.
  - The contract ended because of the sublease to the defendant's sister.
- The desired conclusion for the plaintiff is obtained without proving other conditions (such as non-permission) if there were no counter-arguments,

Reasoning by the JUF Theory for Working Example(continued)

- However, according to the JUF theory, if one of the following counter-arguments is proved by the defendant, the desired conclusion is not obtained.
  - The plaintiff gave the defendant permission by Phrase 1.
  - The defendant did not abuse the confidence of the plaintiff according to the above Supreme Court case rule.
- However, according to the JUF theory, even if there are some facts related to a non-abuse of confidence, the plaintiff can challenge defendant's counter-argument if the plaintiff gave other facts related to the destructin of confidence.

PROLEG
(PROlog based LEGal reasoning support system)

- We implement the JUF theory by using logic programming technology.
- We call the system PROLEG (PROlog based LEGal reasoning support system).
- PROLEG rulebase consists of the following expression.
  - A **rule** of the form of Horn clauses (without negation as failure): $H \Leftarrow B_1, ..., B_n$.
  - An **exception** is an expression of the form $exception(H, E)$ where H, E are atoms each of which is the head of rule.
    This expresses a **rebuttal** of H by E.
- PROLEG factbase consists of the records of activities and judge's confidence over basic facts.
- The above expressions fit lawyer's understanding of civil code nicely.

Reflection of Lawyers' Reasoning: Rulebase

- We could formalize about by logic programming with "Negation as Failure", but the semantics may not be understandable for lawyers.
- Instead of direct representation of JUF theory in logic programming, we use a meta-term to re represent the JUF theory.
- We introduce "`exception`" predicate which takes two arguments, the former of which is the head of default rule and the latter of which is an exception of the rule.

Reflection of Lawyers' Reasoning: Rulebase(continued)

```
cancellation_due_to_sublease <=
    effective_lease_contract,
    effective_sublease_contract,
    using_leased_thing,
    manifestation_cancellation.

effective_lease_contract <=
    agreement_of_lease_contract,
    handover_to_lessee.
effective_sublease_contract <=
    agreement_of_sublease_contract,
    handover_to_sublessee.
```

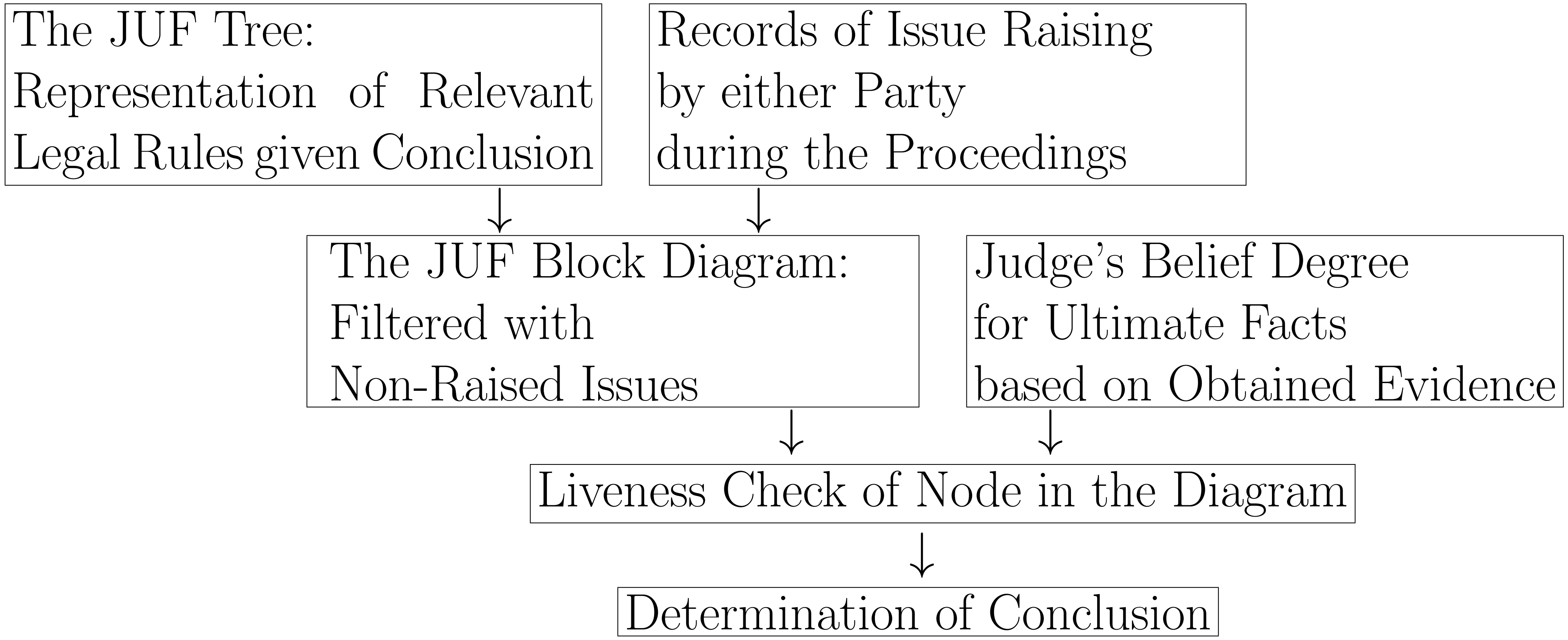
Reflection of Lawyers' Reasoning: Rulebase(continued)

```
exception(cancellation_due_to_sublease,
                    get_approval_of_sublease).
exception(cancellation_due_to_sublease,
                    nonabuse_of_confidence).
get_approval_of_sublease <=
    approval_of_sublease,approval_before_cancellation.
nonabuse_of_confidence<=
    fact_of_nonabuse_of_confidence.
```

Reflection of Lawyers' Reasoning: Factbase

For the base predicate, we put facts in the currentcase and using these facts, we will apply rules to derive conclusion.

Overview of PROLEG Architecture


The JUF Tree: Representation of Relevant Legal Rules given Conclusion
Records of Issue Raising by either Party during the Proceedings
The JUF Block Diagram: Filtered with Non-Raised Issues
Judge's Belief Degree for Ultimate Facts based on Obtained Evidence
Liveness Check of Node in the Diagram
Determination of Conclusion

Demonstration

- Sentential representation of reasoning process in PROLEG
- Diagrammatic representation of reasoning process in PROLEG

Generality of PROLEG

- We can show that a representation power of PROLEG and that of PROLOG are the same.
- We argue that PROLEG can be used not only for civil litigation, but other areas of reasoning with "general rules and exceptions"

Theoretical Generality of PROLEG

We make the following assumption to make PROLEG comparable with LP.

- In the original definition, loop is not allowed, but here we allow a loop.
- In the original definition, we distinguish from facts and rules, but here we do not distinguish it.

PROLEG semantics a la ASP

The definition of PROLEG program:

- A rule is a Horn clause of the form $H \Leftarrow B_1, ..., B_n$.
- An exception is an expression of the form $exception(H, E)$ where H, E are atoms.
- A program P is a pair $\langle \mathcal{H}, \mathcal{E} \rangle$ where $\mathcal{H}$ is a set of rules and $\mathcal{E}$ is a set of exceptions.

PROLEG semantics a la ASP(continued)

Let M be a set of atoms. We define *a set of applicable rules w.r.t. M*, $\mathcal{H}^M$, as follows:

$$\{R \in \mathcal{H} | \neg \exists exception(head(R), E) \in \mathcal{E} \text{ s.t. } E \in M\}.$$

This means that if some exception is found for a conclusion H of rule R, we do not consider such rule R for derivation.

The semantics of P (called an *extension* of P) is given as a set of atoms M s.t. $M = min(\mathcal{H}^M)$
where $min(T)$ is the minimum model of a set of Horn clauses T.

Examples of PROLEG semantics

Example 1 *Let* $P_1 = \{\mathcal{H}_1, \mathcal{E}_1\}$ *and* $\mathcal{H}_1$ *be the following set of rules:*

```
P <=.
Q <=.
R <=.
```

and $\mathcal{E}_1$ *be the following set of exceptions:*

```
exception(P,Q).
exception(Q,R).
```

Let $M_1 = \{\texttt{P,R}\}$. *Then, since* $\mathcal{H}^{M_1} = \{\texttt{P <=. R<=.}\}$ *and* $M_1 = min(\mathcal{H}^{M_1})$, M_1 *is an extension.*
Let $M_1' = \{\texttt{Q}\}$. *Then, since* $\mathcal{H}^{M_!'} = \{\texttt{Q <=. R<=.}\}$ *and* $M_1' \neq min(\mathcal{H}^{M_1'})$, M_1' *is not an extension.*

Examples of PROLEG semantics

Example 2 *Let* $P_2 = \{\mathcal{H}_2, \mathcal{E}_2\}$ *and* $\mathcal{H}_2$ *be the following set of rules:*

```
P <=.
Q <=.
```

and $\mathcal{E}_2$ *be the following set of exceptions:*

```
exception(P,Q).
exception(Q,P).
```

Let $M_2 = \{\texttt{P}\}$. *Then, since* $\mathcal{H}^{M_2} = \{\texttt{P <=.}\}$ *and* $M_2 = min(\mathcal{H}^{M_2})$, M_2 *is an extension.*
Let $M_3 = \{\texttt{Q}\}$. *Then, since* $\mathcal{H}^{M_3} = \{\texttt{Q <=.}\}$ *and* $M_3 = min(\mathcal{H}^{M_3})$, M_3 *is also an extension.*

Examples of PROLEG semantics(continued)

Example 3 *Let* $P_3 = \{\mathcal{H}_3, \mathcal{E}_3\}$ *and* $\mathcal{H}_3$ *be the following set of rules:*

```
P <=.
```

and $\mathcal{E}_3$ *be the following set of exceptions:*

```
exception(P,P).
```

Let $M_4 = \{\texttt{P}\}$*. Then, since* $\mathcal{H}^{M_4} = \{\}$ *and* $M_4 \neq min(\mathcal{H}_4^M)$*,* M_4 *is not an extension.*
Let $M_5 = \{\}$*. Then, since* $\mathcal{H}^{M_5} = \{\texttt{P <=.}\}$ *and* $M_5 \neq min(\mathcal{H}_5^M)$*,* M_5 *is neither an extension.*
Since there is no other possibility of extension of P_2*, there is no extension for* P_2*.*

Translation from PROLEG to PROLOG(ASP) (JURISIN 2010)

$P = \langle \mathcal{H}, \mathcal{E} \rangle$ where $\mathcal{H}$ is as follows:

$$C \Leftarrow B_{11}, ..., B_{1n_1}.$$
$$C \Leftarrow B_{21}, ..., B_{2n_2}.$$
$$\vdots$$
$$C \Leftarrow B_{k1}, ..., B_{kn_k}.$$

and $\mathcal{E}$ is as follows:

$$exception(C, E_1).$$
$$\vdots$$
$$exception(C, E_m).$$
$$\Downarrow$$

$C{:}-B_{11}, ..., B_{1n_1},$ `not` $E_1, ...,$ `not` $E_m.$
$C{:}-B_{21}, ..., B_{2n_2},$ `not` $E_1, ...,$ `not` $E_m.$
$\vdots$
$C{:}-B_{k1}, ..., B_{kn_k},$ `not` $E_1, ...,$ `not` $E_m.$

Note that rules with the same head has the same negative literals.

Translation from PROLEG to PROLOG(ASP)

Consider the working example where $P_1 = \{\mathcal{H}_1, \mathcal{E}_1\}$ and $\mathcal{H}_1$ be the following set of rules.

```
P<=.
Q<=.
```

and $\mathcal{E}_1$ be the following set of exceptions:

```
exception(P,Q).
exception(Q,R).
```

A model of this PROLEG program is $\{P, R\}$.

$tr_e(P_1)$ is as follows:

```
  P:- not Q.
  Q:- not R.
  R.
```

A model of this PROLOG program is $\{P, R\}$.

Translation from PROLEG to PROLOG(ASP)

Theorem 1 *Let P be a PROLEG program and $tr_e(P)$ be a PROLOG program obtained by the above translation. Then, there is an extension of P, M if and only if there is an answer set M of $tr_e(P)$.*

Translation from PROLOG(ASP) to PROLEG

Let P be the following PROLOG program:

C: $-B_{11}, ..., B_{1n_1}$, `not` $E_{11}, ...,$ `not` E_{1m_1}.
C: $-B_{21}, ..., B_{2n_2}$, `not` $E_{21}, ...,$ `not` E_{2m_1}.
$\vdots$
C: $-B_{k1}, ..., B_{kn_k}$, `not` $E_{k1}, ...,$ `not` E_{km_k}.

Then, a translation of a PROLOG program P, $tr_o(P)$ is defined as the following PROLEG program $\langle \mathcal{H}_o, \mathcal{E}_o \rangle$ where $\mathcal{H}_o$ is as follows:

$C \Leftarrow C_1. \quad C_1 \Leftarrow B_{11}, ..., B_{1n_1}.$
$C \Leftarrow C_2. \quad C_2 \Leftarrow B_{21}, ..., B_{2n_2}.$
$\vdots$
$C \Leftarrow C_k. \quad C_k \Leftarrow B_{k1}, ..., B_{kn_k}.$

and $\mathcal{E}_e$ is as follows:

$exception(C_1, E_{11}). \quad \cdots \quad exception(C_1, E_{1m_1}).$
$exception(C_2, E_{21}). \quad \cdots \quad exception(C_2, E_{2m_2}).$
$\vdots$
$exception(C_k, E_{k1}). \quad \cdots \quad exception(C_k, E_{km_k}).$

Translation from PROLOG(ASP) to PROLEG

Consider the following PROLOG program $tr_e(P_1)$:

```
P: − not Q.
Q: − not R.
R.
```

Then, $tr_o(tr_e(P_1)) = \langle \mathcal{H}_{oe}, \mathcal{E}_{oe} \rangle$ becomes (by introducing some new predicates) where $\mathcal{H}_{oe}$ is as follows:

```
P <= c1.
c1 <=.
Q <= c2.
c2 <=.
R <=.
```

and $\mathcal{E}_{oe}$ becomes the following set of exceptions:

```
exception(c1,Q).
exception(c2,R).
```

The model of this PROLEG program is $\{P, c2, R\}$.

Translation from PROLOG(ASP) to PROLEG

Theorem 2 *Let P be a PROLOG program and $tr_o(P)$ be a PROLEG program obtained by the above translation. Then, there is an answer set M of P if and only if there is an extension of $tr_o(P)$ such that a set deleting newly introduced predicates from the extension equals to M.*

So this means that **representation power of PROLEG is the same as that of PROLOG.**

Application to Penal Code

- A general rule of criminal law is that if the prosecutor proves that the external elements (*actus reus*) and the internal elements (*mens rea*) of the crime then the actor of the crime is prosecuted.
- However, there are some exceptions such as insanity, self-defense.
- However, for self-defense, imminent and unlawful infringement must exist but for a situation such that the defender have an intention of attack using this chance, then imminent condition will be denied.

Application to Penal Code(continued)

We could represent these general rules and exceptions in PROLEG.

```
guilty(X,Y,murder)  <=
  intention_of_killing(X,Y),
  killing_act(X,Y),
  died(Y).
insane(X)  <=
  psychiatric_test_insanity(X).
self_defense(X,Y) <=
  imminent_infringement(Y,X),
  unlawful_infringement(Y,X),
  defense_action_appropriate(X,Y).
```

and exceptions as:

```
exception(guilty(X,Y,murder),insane(X)).
exception(guilty(X,Y,murder),self_defense(X,Y)).
exception(imminent_infringement(Y,X),
                      intention_of_attack(X,Y)).
```

Application to Code of Criminal Procedure

- In general, compulsory dispositions are prohibited.
- However, in the exceptional situation where special provisions are made, the compulsory dispositions are allowed.
- If dispositions are not compulsory (or in other words depositions are with consent), they are allowed.
- However, in the exceptional situation where the dispositions are the above necessary level, they are prohibited.

Application to Code of Criminal Procedure(continued)

We could write these general rules in PROLEG as:

```
prohibited(Police,X,Disposition)  <=
  compulsory(Disposition).
allowed(Police,X,Disposition)  <=
  with_consent(X,Disposition).
```

and exceptions as:

```
exception(prohibited(Police,X,Disposition),
  provisions(Disposition)).
exception(allowed(Police,X,Disposition),
  unnecessary(Police,X,Disposition)).
```

Application to Constitution Law

- A general rule for freedom expression is that constraining expression is prohibited.
- However, if the purpose of constraining expression is compelling and its constraining method is narrowly tailored, it is allowed.
- Also, if the constraint is content-neutral regulation, then the standard of judicial review becomes softer as an exception.

Application to Constitution Law(continued)

```
prohibitted(X,Y,Constraint)  <=
  constrain(X,Y,Constraint).
allowed(X,Y,Constraint) <=
  purpose_of_constraint_compelling(X,Y,Constraint,Purpose),
  method_of_constraint_narrowly_tailored(X,Y,Constraint,Metho
allowed(X,Y,Constraint) <=
  content_neutral_constraint(Constraint),
  purpose_of_constraint_imporant(X,Y,Constraint,Purpose),
  method_of_constraint_least_restrictive(X,Y,Constraint,Metho
```

and exception can be expressed as follows:

```
exception(prohibitted(X,Y,ConstrainedExpression),
  allowed(X,Y,ConstrainedExpression)).
```

Current Status of PROLEG Projects(continued)

- **Implementing the full JUF theory in PROLEG:** We have implemented more than 2500 rules in civil law and supreme court cases by Todai law school graduates.
- **Answering yes/no question of Japanese Bar Exams:** As a part of Japanese Bar Exams, there are multiple choice questions to choose two or three correct statements among five.
  Each choice can be regarded as yes/no questions.
  We are now starting a project to solve these yes/no questions by a combination of natural language processing and PROLEG to solve subsumption problem and provide a correct explanation why the statment is correct/incorrect.

Conclusion on PROLEG

Contributions: Proposal of PROLEG system

- which reflects lawyers' reasoning about law where exceptions are ignored unless it is explicitly stated.
- which handles legal actions at the court facts such as allegement, evidence production, plausibility and admission.

Conclusion on Juris-informatics

- I found that there are many opportunities to apply informatics/AI techniques to legal reasoning besides the above work which have not been explored because of the academic gaps between informatics/AI field and law.
- In general, there are much unwritten implicit knowledge which do not appear in textbooks in other domains and it is possible to acquire these knowledge only if we enter the field deeply.
- I believe that we should promote this kind of activity (letting researchers in informatics entering other field to get PhD or MsC) to make a great success of amalgamating informatics and other fields.