A Use Case on GDPR of Modular-PROLEG for Private International Law

Takahiro Sawasaki[1], Ken Satoh[1] and Aurore Clément Troussel[2]

[1]National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
[2]Cyberjustice Laboratory, Faculty of Law, Université de Montréal, C.P. 6128, Downtown Branch Montreal (Quebec), Canada, H3C 3J7

Abstract
When a legal dispute involves the laws of two or more countries, private international law resolves the conflict of laws by identifying which country has jurisdiction over the case and which national law applies. The automation of reasoning in the choices of jurisdiction and law has been done by Modular-PROLEG for PIL. The paper presents a new use case of Modular-PROLEG for PIL in which one must choose EU's General Data Protection Regulation (GDPR) or Japan's law on the data protection according to Japan's private international law.

Keywords
legal reasoning, private international law, choice of jurisdiction, choice of law, GDPR, PROLEG

1. Introduction

When a legal dispute involves the laws of two or more countries, private international law resolves the conflict of laws by identifying which country has jurisdiction over the case and which national law applies. To automate the complex process which private international law (PIL) performs, Satoh et al. [1] implemented a meta-interpreter Modular-PROLEG for PIL which automatically reasons about the choice of jurisdiction and law in the context of a dispute over legitimate parent-child relationship.

In this paper, we use Modular-PROLEG for PIL in a new context : a transfer of personal data between three countries. The question is which law between the EU's General Data Protection Regulation (GDPR) and the Japanese data protection law should apply to the transfer according to Japan's PIL. This case illustrates how Modular-PROLEG for PIL can be used in practice.

Related works include the one of Dung and Sartor [2] which provide an analysis of private international law and propose a formal model based on a framework called modular argumentation, which is an instance of abstract argumentation defined as a pair $(AR, attacks)$ of a set AR of arguments and a binary relation $attacks$ on AR [3]. Also,

Joint Proceedings of ISWC2022 Workshops: the International Workshop on Artificial Intelligence Technologies for Legal Documents (AI4LEGAL) and the International Workshop on Knowledge Graph Summarization (KGSum) (2022)

tsawasaki@nii.ac.jp (T. Sawasaki); ksatoh@nii.ac.jp (K. Satoh); aurore.troussel@umontreal.ca (A. C. Troussel)

Calegari et al. [4] implement a reasoning about jurisdiction using a framework called Arg-tuProlog, which is a lightweight modular argumentation inspired by [2]. Markovich [5] uses the input/output logic introduced in [6] to specify rules in PIL. The advantage of Modular-PROLEG for PIL over these works is that it identify both jurisdiction and law in international legal dispute. Another related work is Governatori et al. [7], which presents a comprehensive computational framework on the PIL's reasoning patterns modeled by [2, 8, 9]. In addition to reason about the choice of law and jurisdiction, this framework allows to reflect about interpretive arguments. However, unlike Modular-PROLEG for PIL, this framework is not a specific implementation of the reasoning about the choice of jurisdiction and law. Modular-PROLEG for PIL also differs from PROLEG developed in [10] mainly in that PROLEG cannot reason about the choice of law.

The paper proceeds as follows. Section 2 briefly introduces Modular-PROLEG for PIL implemented in [1]. Section 3 presents a hypothetical case in which several laws including GDPR are involved and identifies the applicable laws in order to accept or reject the claims made in the case. Section 4 illustrates how we implemented the reasoning found in Case of Data Transfer by Modular-PROLEG for PIL, confirming that our implementation returns the answers corresponding to the acceptance or rejection of the claims made in Section 3.

2. Modular-PROLEG for PIL

In this section, we refer to [1] to briefly introduce the syntax and meta-interpreter of Modular-PROLEG for PIL.[1] We skip the semantics here since it is less relevant to seeing a use case of Modular-PROLEG for PIL. See [1, p. 8] for the details.

2.1. Syntax

We refer to a Horn clause in either of the following forms as a rule:

- $H\#C \Leftarrow B_1\#C_1, \cdots, B_n\#C_n$
- $envoi(H\#C, RC) \Leftarrow B_1\#C_1, \cdots, B_n\#C_n$

where H, B_i are atomic formulas and C, C_i, RC are variables. We call $H\#C$ and $envoi(H\#C, RC)$ as the head of each rule, and $B_i\#C_i$s as requisites of each rule. When $C = C_i$ for all $i \leqslant n$, we often write the above rules as $(H \Leftarrow B_1, \cdots, B_n)\#C$ and $(envoi(H, RC) \Leftarrow B_1, \cdots, B_n)\#C$. The intended readings of $H\#C$ and $envoi(P, RC)\#C$ are "H holds under the law of country C" and "the law of country C applies the law of country RC to P," respectively. We also refer to expressions of the following forms as exception and fact:

- $exception(H\#C, E\#C)$

[1] There are two differences on Modular-PROLEG for PIL. First, in [1] the set $\mathscr{R}$ may contain exceptions, but may not in our paper. This change is only for readability. Second, where a country has no explicit specification of choice of law regarding a case, the meta-interpreter in our paper chooses the law of the country as the applicable law.

- $fact(H\#C)$

where $exception(H\#C, E\#C)$ is often written as $exception(H, E)\#C$. We read them as "E is an exception of H under the law of country C" and "H is a fact under the law of country C," respectively.

A program $\mathscr{P}$ of Modular-PROLEG for PIL is a tuple $(\mathscr{R}, \mathscr{E}, \mathscr{F})$ of finite sets $\mathscr{R}$, $\mathscr{E}$, $\mathscr{F}$ consisting of rules, exceptions and facts, respectively.

2.2. Meta-Interpreter of Modular-PROLEG for PIL

Here is the meta-interpreter of Modular-PROLEG for PIL (Algorithm 1, 2).[2]

Algorithm 1 Modular-PROLEG Meta-Interpreter for PIL

```
pilprove($S$) goal set $S$
if $S == \emptyset$ then return($\emptyset$);
for all atom $A\#C \in S$ do
    if $A\#C$ is an atom s.t. $fact(F\#C) \in \mathscr{F}$ and $F\#C$ is unifiable with $A\#C$ by mgu $\theta$ then
        $S := (S - \{A\#C\})\theta$;
        return(pilprove($S$))
    else   % $A\#C$ should be a head in some rules
        $RC$ := choice_of_law($A\#C, \{\}$);
        select a rule $(H \Leftarrow B_1, \ldots, B_n)\#RC \in \mathscr{R}$ where its head $H\#C$ is unifiable with $A\#C$ by mgu $\theta$;
        if such a rule does not exist then return(false);
        $S := ((S - \{A\#C\}) \cup \{B_1\#RC, \ldots, B_n\#RC\})\theta$;
        if pilprove($S$) == $S\delta$ by mgu $\delta$ then
            for all $exception(G, E)\#C \in \mathscr{E}$ s.t. $(A\#C)\theta\delta$ is unifiable with $G\#C$ by mgu $\eta$ do
                if pilprove($\{(E\#C)\theta\delta\eta\}$) then return(false);
            return($S\delta$)
        else if pilprove($S$) == false then return(false)
```

Algorithm 2 Choice of Law

```
choice_of_law($A\#C, ReferringHistory$) $A\#C$: a goal
if $C \in ReferringHistory$ then return($C$)
else if $A == envoi(\_, \_)$ then return($C$)
else if $solve(pil, \{envoi(A, RC)\#C\}) == envoi(A\#C, RC)\theta$ by mgu $\theta$ then
    **return**(choice_of_law($(A\#RC)\theta, \{RC\theta\} \cup ReferringHistory$))
else **return**($C$)   % choose a law of home country if there are no specifications of choice of law
```

[2] Unfortuately, the current version of Modulr-PROLEG for PIL cannot generate the graphical schema of the legal reasoning from a given program. To see how these algorithms work, see [1].

3. A Case on Transfer of Personal Data among Three Countries

A use case on legitimate parent-child relationship in Japan has already been presented in [1, p. 2–3]. In this section, we present a new use case of Modular-PROLEG for PIL in which one must invoke the laws of two or more countries including GDPR.

Let us consider the following hypothetical case on transfers of personal data between three countries, hereinafter "Case of Data Transfer":

> Case of Data Transfer A Japanese company who is established in Japan has two branch offices. One is in Country 1 inside the EU and the other is in Country 2 outside the EU. To compare and monitor the productivity of its employees in its different offices, the company constantly performs transfers of data of the employees between Japan, Country 1 and Country 2. The employees and the company agreed on a contract ($cj1$) according to which the employees might work in collaboration with Country 1's office for specific tasks. In addition, the employees and the company signed another contract ($c12$) according to which the employees who collaborated with Country 1's office might work with Country 2's office. The two contracts briefly mention that employees' data such as name and job position might be transferred to other offices in order to allow personnel transfer. None of these contracts explained to the employees that their data will be transferred to Country 1 and Country 2's offices to compare and monitor their productivity with other employees. The company has not obtained any explicit agreement on these data transfers from the employees. Furthermore, the company has not provided any appropriate safeguard on the data transfers and Country 2 does not benefit from an adequate level of data protection according to EU law (i.e., Country 2 does not have an adequacy decision). Some of the employees working at the main office in Japan have questioned the situation, and finally sued the company for damages because the company transferred their data in breach of the contracts ($cj1$) and ($c12$). The employees and the company agreed that the law that governs the formulation and effect of the contract ($cj1$) is the Japanese law, and that the law of Country 1 governs those of the contract ($c12$).

In recent years, cases involving international data transfers, such as Schrems II[3], received much attention, thus it is interesting to implement Modular-PROLEG for PIL in Case of Data Transfer.

The question is whether the employees' claims for damages would be accepted. More precisely, we wonder if the following litigations can succeed:

($lj1$) The litigation of the employees of the Japan's office against the Japanese company for damages because the company transferred their data from Japan's office to Country 1's office in breach of the contract ($cj1$);

[3] https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX:62018CJ0311

($l12$) The litigation of the employees of the Japan's office against the Japanese company for damages because the company transferred their data from Country 1's office to Country 2's office in breach of the contract ($c12$).

In what follows, we assume that Japan has jurisdiction over the employees' litigations and that each of ($lj1$) and ($l12$)'s claims for damages are accepted if and only if the contested data transfer is illegal under the applicable law.

The following legal reasoning shows that ($lj1$)'s claims for damages would not be accepted: Japan has jurisdiction over the litigations involving the employees of Japan's office. Thus, the PIL of Japan, i.e., Act on General Rules for Application of Laws (AGRAL),[4] determines which country's law governs the acceptance of ($lj1$)'s claims. Because ($lj1$) is grounded on the fact that the company breached the contract ($cj1$), the governing law of the acceptance of ($lj1$)'s claims is the law that governs the formulation and effect of the contract ($cj1$). To identify the law that governs the contract ($cj1$), we refer to Article 7 of AGRAL:

> Article 7, AGRAL The formation and effect of a juridical act are governed by the law of the place chosen by the parties at the time of the act.

Under Japanese law, the contract ($cj1$) is a juridical act. In addition, in Case of Data transfer, the employees and the company choose the Japanese law to govern the formation and effect of the contract ($cj1$). Thus, according to Article 7 of AGRAL, the Japanese law is the applicable law that determines whether ($lj1$)'s claims are accepted.

Now that the applicable law is identified, we can test whether ($lj1$)'s claims are accepted under the Japanese law. The legal question is whether the transfer of the employees' data from Japan's office to Country 1's office is lawful. To answer this question, the most relevant article is Article 24 of the Japanese law called Act on the Protection of Personal Information (APPI).[5]

> Article 24, APPI A personal information handling business operator, except in those cases set forth in each item of the preceding Article, paragraph (1), shall, in case of providing personal data to a third party [...] in a foreign country [...], in advance obtain a principal's consent to the effect that he or she approves the provision to a third party in a foreign country. [...]

The exceptional clause "except in those cases set forth in each item of the preceding Article" does not apply to Case of Data Transfer. At a first glance, as the company did not obtain an explicit agreement on the disputed data transfers from the employees, the data transfer from the Japan's office to the Country 1's office could be considered as illegal. However, as the receiver of personal data is a branch office of the company, it has the same legal personality, and thus it is not considered as a third party under APPI.[6]

[4] http://www.japaneselawtranslation.go.jp/law/detail/?printID=&id=3783&re=02&vm=02 (Mar. 22, 2022)

[5] http://www.japaneselawtranslation.go.jp/law/detail/?printID=&id=2781&vm=02&re=02 (Mar. 22, 2022)

[6] See p. 45 and p. 5 of the two guidelines of APPI which are found at https://www.ppc.go.jp/files/pdf/210101_guidlines01.pdf and https://www.ppc.go.jp/files/pdf/210101_guidlines02.pdf (Mar. 22, 2022), respectively (Japanese only).

Therefore, it is not illegal to transfer the employees' data from the Japan's office to the Country 1's office without the employees' agreement, from which it follows that ($lj1$)'s claims are not accepted.

Conversely, the following legal reasoning shows that ($l12$)'s claims would be accepted: Using the same legal argument as in ($lj1$), we consider that the law that governs the formation and effect of the contract ($c12$) is the applicable law. The employees and the company have chosen Country 1's law to govern the contract (c12), thus it is Country 1's law that determines the acceptance of ($l12$)'s claims. As Country's 1 is in the EU, the GDPR applies to the case.

To know whether ($l12$)'s claims are accepted, we need to know whether it is illegal to transfer the employees' data from Country 1's office to Country 2's office. The important articles are Article 44 and Article 49(1) of GDPR.[7]

> Article 44, GDPR Any transfer of personal data which are undergoing processing or are intended for processing after transfer to a third country or to an international organisation shall take place only if, subject to the other provisions of this Regulation, the conditions laid down in this Chapter are complied with by the controller [...].

In a nutshell, Article 44 rules that any transfer of personal data to a third country is prohibited by default. However, there are exceptional rules allowing such a transfer (Articles 45, 46, 49(1) etc.). Regarding Case of Data Transfer, Article 49(1) is particularly important.

> Article 49(1), GDPR In the absence of an adequacy decision [...] or of appropriate safeguards [...], a transfer or a set of transfers of personal data to a third country or an international organization shall take place only on one of the following conditions:
>
> (a) The data subject has explicitly consented to the proposed transfer, after having been informed of the possible risks of such transfers for the data subject due to the absence of an adequacy decision and appropriate safeguards.
>
> (b) The transfer is necessary for the performance of a contract between the data subject and the controller or the implementation of pre-contractual measures taken at the data subject's request.
>
> ⋮

Case of Data Transfer does not satisfy the conditions for the applications of exceptional rules: Country 2 does not benefit from an adequacy decision with the EU, and the company did not put in place appropriate safeguards to transfer data. In addition, the employees did not explicitly consent to the transfer of data from Country 1 to Country 2 and the transfers of data are not strictly necessary for the performance of the contract (c12) as the goal of the transfers is to compare and monitor the productivity of

[7] https://gdpr-text.com/read/article-44/#para_gdpr-a-44 (Mar. 22, 2022)

employees. Therefore, it is illegal to transfer the employees' data from Country 1's office to Country 2's office, from which it follows that $(l12)$' claims are accepted.

4. Implementation of Reasoning in the Case

In the previous section, we considered a hypothetical case named as Case of Data Transfer in which $(l12)$'s claims are accepted but $(lj1)$'s claims are not. In this section, we illustrate how we implemented the reasoning found in Case of Data Transfer by Modular-PROLEG for PIL.

We used Modular-PROLEG for PIL to implement national laws with non-envoi rules, private international laws with envoi rules, and exceptional laws with exceptions, together with appropriate paraphrases if necessary.

It is important to explain how negation-as-failure (NAF) is handled in Modular-PROLEG for PIL. Modular-PROLEG for PIL does not have a built-in predicate like not/1 which functions as NAF, but legal texts usually have many negative expressions like "in the absence of an adequacy decision". Thus, we need a method to formalize these expressions literally. Our method consist in adding the following three lines into every program:

```
1: (negation(P) <= call(true))#C.
2: exception(negation(P),P)#C.
3: (envoi(negation(P),RC) <= envoi(P,RC))#C.
```

where (i) P must be a ground term in testing *negation*(P)s and (ii) the line 2 must be the only exception of *negation*(P) in the program. Technically P could be any ground term, but practically it is supposed to be an atomic formula. The condition (i) is necessary because programs without (i) can easily go into a loop. For example, a program consisting only of a rule $(p \Leftarrow \mathit{negation}(\_))\#\_$ and the lines 1–3 for negation/1 goes into a loop given an input p. The condition (ii) is necessary to make negation/1 function as NAF.

The set of the lines 1–3 guarantees that negation/1 functions as NAF. The line 1 says that the negation of an atomic formula succeeds by default, which means that if the negation of an atomic formula fails, then its (only) exception must succeed. The line 2 says that an atomic formula is the exception of its negation, which means that if an atomic formula succeeds then its negation must fail. Accordingly, they jointly say that an atomic formula succeeds if and only if its negation fails. The line 3 is needed to make sure that the selected law for *negation*(P) is the same as the selected law for P.

It should also be noted that the above method correctly captures the notion of the burden of proof. Consider a trial in which a defendant tries to claim that H is the case according to some article "if B is not the case then H is the case". In jurisprudence, the proof of the negation of something is very difficult to provide and is called probatio diabolica. For this reason, the defendant does not bear the burden of proof of the negation of B in the trial. Instead, the plaintiff bears the burden of proof of B. When we implement this article as $H \Leftarrow \mathit{negation}(B)$, the above method correctly captures the fact that the plaintiff bears the burden of proof of B because then *negation*(B) succeeds by default unless B succeeds.

Below, Article 7 of AGRAL, Article 24 of APPI and Articles 44, 49(1) of GDPR are implemented with Modular-PROLEG for PIL.

Article 7 of AGRAL has the form of a rule in Modular-PROLEG for PIL and can be implemented straightforwardly as follows:

```
1: (formulationAndEffect(Pla,Def,Contract,RC) <=
     plaintiff(Pla),
     defendant(Def),
     jurisAct(Contract,Pla,Def),
     agral(ArticleNum,[Contract,RC]))#ja.
2: (agral(7,[Contract,RC]) <= choseAtThatTime(parties(Contract),RC))#ja.
```

The line 1 describes the prerequisites of Article 7 and the related articles, and the line 2 describes the substantial condition of Article 7.

Article 24 of APPI does not have the form of a rule in Modular-PROLEG for PIL, thus it must be paraphrased to implement. We paraphrase it as a rule that any transfer of personal data to a third party in a third country of Japan will be illegal by default, except when the data subject gave his consent and in the specific situations listed in Article 23(1). We thus implemented it as follows:

```
(illegal(transfer(S,G,Data)) <= appi(ArticleNum,transfer(S,G,Data)))#ja.
(appi(24,transfer(S,G,Data)) <=
   operator(X,transfer(S,G,Data),appi),
   recipient(Y,transfer(S,G,Data),appi),
   inThirdCountry(Y,ja),
   thirdParty(Y,X,appi),
   transfer(S,G,Data)).
exception(appi(24,transfer(S,G,Data)),ex_appi(24,transfer(S,G,Data)))#ja.
(ex_appi(24,transfer(S,G,Data)) <=
   subj(Subj,Data),
   consent(Subj,transfer(S,G,Data)))#ja.
(ex_appi(24,transfer(S,G,Data)) <=
   ex_appi(23,1,transfer(S,G,Data)))#ja.
```

Similarly, Article 44 of GDPR can be paraphrased as a rule that any transfer of personal data to a third country out of EU is illegal by default, except when exceptional rules like Article 49(1) apply. It is implemented as follows:

```
(illegal(transfer(S,G,Data)) <= gdpr(ArticleNum,transfer(S,G,Data)))#c1.
(gdpr(44,transfer(S,G,Data)) <=
   inThirdCountry(G,eu),transfer(S,G,Data))#c1.
(exception(gdpr(44,transfer(S,G,Data)),ex_gdpr(44,transfer(S,G,Data))))#c1.
```

Finally, Article 49(1) of GDPR has the form of a rule with negations in Modular-PROLEG for PIL and can be implemented straightforwardly as follows:

```
(ex_gdpr(44,transfer(S,G,Data)) <= gdpr(49,ParagraphNum,transfer(S,G,Data)))#c1.
(gdpr(49,1,transfer(S,G,Data)) <=
   countryOf(C,G),
   negation(adequateDecision(eu,C,gdpr(45,3))),
   ctrlPrcsr(CtrlPrcsr,transfer(S,G,Data),gdpr),
   negation(asg(CtrlPrcsr,transfer(S,G,Data))),
   gdpr(49,1,ItemNum,transfer(S,G,Data)))#c1.
```

```
3: (gdpr(49,1,a,transfer(S,G,Data)) <=
     subj(Subj,Data),
     consent(Subj,transfer(S,G,Data)),
     knowRiskBeforeConsent(Subj,transfer(S,G,Data),
     consent(Subj,transfer(S,G,Data))))#c1.
4: (gdpr(49,1,b,transfer(S,G,Data)) <=
     subj(Subj,Data),
     ctrlr(CtrlPrcsr,transfer(S,G,Data),gdpr),
     contract(Contract,Subj,CtrlPrcsr),
     need(performance(Contract),transfer(S,G,Data)))#c1.
5: (gdpr(49,1,b,transfer(S,G,Data)) <=
     subj(Subj,Data),
     ctrlr(CtrlPrcsr,transfer(S,G,Data),gdpr),
     preContMeasure(Measure,Subj,CtrlPrcsr),
     need(implement(Measure),transfer(S,G,Data)))#c1.
   ...
```

where $ctrlPrcsr(X, transfer(S, G, Data), gdpr)$ means that an agent is the controller or processor of a transfer of data and $asg(CtrlPrcsr, transfer(S, G, Data))$ means that the controller or processor has provided appropriate safeguards for a transfer of data.

The paragraphs above illustrated our implementation for Case of Data Transfer. In Section 3, we wondered if the litigations ($lj1$) and ($l12$) succeeded, i.e., ($lj1$) and ($l12$)'s claims for damages were accepted. Below, we present how their claims are implemented with Modular-PROLEG for PIL:

(1) `claim(empl(o(ja)),co,inBreachOf(transfer(o(ja),o(c1),data(empl(o(ja)))),cj1))#ja`

(2) `claim(empl(o(ja)),co,inBreachOf(transfer(o(c1),o(c2),data(empl(o(ja)))),c12))#ja`

where $claim(Pla, Def, inBreachOf(Action, Contract))$ means that Pla's claims for damages against Def are accepted because $Action$ is done in breach of $Contract$. Let $\mathscr{P}_1$ be the program for Case of Data Transfer containing the above codes.[8] If we query (1) and (2) together with $\mathscr{P}_1$, the interpreter of Modular-PROLEG for PIL returns that (1) is false but (2) is true, which corresponds to our conclusion in Section 3 that ($lj1$)'s claims are not accepted but ($l12$)'s claims are accepted.

5. Conclusion

This paper presents a new use case of Modular-PROLEG for PIL in the context of international data transfers. The task is to identify which law applies between the European data protection law (GDPR) and the Japanese data protection law (APPI) according to Japan's private international law (AGRAL). As a future work, it would be interesting to provide a more detailed implementation of GDPR, APPI and AGRAL by Modular-PROLEG for PIL. For example, in this paper we implemented the conditions that someone is a processor in GDPR and that someone is an operator in APPI as mere

[8] See https://github.com/tsawasaki/mprolegpil-usecase

facts. However, GDPR and APPI actually give us the definitional rules of processor and operator so that these conditions follow from more primitive facts. The implementation of these definitional rules would allow us to start from the more primitive facts to check that both of these condition succeed, which would more precisely reflect the real process done by jurists.

Acknowledgments

We thank Yuto Mori for making valuable comments on private international law of Japan. This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470, JSPS Topic-Setting Program to Advance Cutting-Edge Humanities JPJS00119217216 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

[1] K. Satoh, L. Giordano, M. Baldoni, Implementation of Choice of Jurisdiction and Law in Private International Law by PROLEG Meta-interpreter, in: Lecture Notes in Computer Science, Springer International Publishing, 2021, pp. 60–75. doi:10.1007/978-3-030-89391-0_4.

[2] P. M. Dung, G. Sartor, The Modular Logic of Private International Law, Artificial Intelligence and Law 19 (2011) 233–261. doi:10.1007/s10506-011-9112-5.

[3] P. M. Dung, On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-person Games, Artificial Intelligence 77 (1995) 321–357.

[4] R. Calegari, G. Contissa, G. Pisano, G. Sartor, G. Sartor, Arg-tuProlog: A Modular Logic Argumentation Tool for PIL, in: S. Villata, J. Harata, P. Kemen (Eds.), Legal Knowledge and Information Systems, volume 334 of Frontiers in Artificial Intelligence and Applications, 2020, pp. 265–268.

[5] R. Markovich, On the Formal Structure of Rules in Conflict of Laws, in: M. Araszkiewicz, V. Rodríguez-Doncel (Eds.), Legal Knowledge and Information Systems, volume 322 of Frontiers in Artificial Intelligence and Applications, 2019, pp. 199–204.

[6] D. Makinson, L. van der Torre, What is Input/Output Logic?, in: Foundations of the Formal Sciences II, Springer, 2003, pp. 163–174. doi:10.1007/978-94-017-0395-6_12.

[7] G. Governatori, F. Olivieri, A. Rotolo, A. Sattar, M. Cristani, Computing Private International Law, in: E. Schweighofer (Ed.), Legal Knowledge and Information Systems, volume 346 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2021, pp. 181–190. doi:10.3233/faia210334.

[8] A. Malerba, A. Rotolo, G. Governatori, Interpretation Across Legal Systems, in: S. V. Floris Bex (Ed.), Legal Knowledge and Information Systems, volume 294 of Frontiers in Artificial Intelligence and Applications, 2016, pp. 83–92.

[9] A. Malerba, A. Rotolo, G. Governatori, A Logic for the Interpretation of Private International Law, in: S. Rahman, M. Armgardt, H. C. N. Kvernenes (Eds.), New Developments in Legal Reasoning and Logic: From Ancient Law to Modern Legal Systems, volume 23 of Logic, Argumentation and Reasoning, Springer International Publishing, 2021, pp. 149–169. doi:10.1007/978-3-030-70084-3_7.

[10] K. Satoh, K. Asai, T. Kogawa, M. Kubota, M. Nakamura, Y. Nishigai, K. Shirakawa, C. Takano, PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology, in: New Frontiers in Artificial Intelligence, Springer Berlin Heidelberg, 2011, pp. 153–164. doi:10.1007/978-3-642-25655-4_14.