

Arg-tuProlog: A Modular Logic Argumentation Tool for PIL

Roberta CALEGARI ^{a,1}, Giuseppe CONTISSA ^a, Giuseppe PISANO ^a,
Galileo SARTOR ^c and Giovanni SARTOR ^{a,b}

^a *CIRSFID—Alma AI, University of Bologna, Italy*

^b *European University Institute, Florence, Italy*

^c *University of Torino, Torino, Italy*

Abstract. Private international law (PIL) addresses overlaps and conflicts between legal systems by distributing cases between the authorities of such systems (jurisdiction) and establishing what rules these authorities have to apply to each case (choice of law). A modular argumentation tool, Arg-tuProlog, is here presented that enables reasoning with rules and interpretations of multiple legal systems.

Keywords. modular argumentation, private international law, Arg-tuProlog

1. Introduction

In our increasingly pervasive and interconnected world, the application and enforcement of the law makes it necessary to take into account the interplay of multiple normative systems, especially when dealing with international contracts and other commercial and social interactions involving different countries. Moreover, normative systems may also interact or conflict on different levels: this is true of both national legal systems and of various transnational or international laws and conventions. All these sources of law need to be considered to properly reason about the law. The research in this paper focuses on the field of private international law (PIL) – a growing and important domain of the law – which deals with the coexistence of multiple normative systems, having distinct and often contradictory rules, and the legal interaction of persons connected to different legal systems, trying to establish priorities between them. Conflicts about competences and rules are addressed by identifying which authority is responsible for making a decision in each given case (jurisdiction), and which set of norms should be applied (applicable law). A logical analysis of PIL has highlighted how this body of law can be suitably modelled by modular argumentation [1] so as to provide a formal model of the interaction among multiple legal systems. For this reason, we are here showing the Arg-tuProlog[2] modular extension to support modular reasoning according to the concept of modularity introduced in [1]—not yet captured and implemented in any known ready-to-use technology.

¹Roberta Calegari and Giovanni Sartor have been supported by the H2020 ERC Project “CompuLaw” (G.A. 833647). Giuseppe Contissa, Giuseppe Pisano and Galileo Sartor have been supported by the European Union’s Justice programme under Grant Agreement No. 800839 for the project “InterLex: Advisory and Training System for Internet-related private International Law”.

Arg-tuProlog is a lightweight modular argumentation tool that fruitfully combines modular logic programming and legal reasoning. It makes it possible to represent, reason, and carry out an argument on conditional norms featuring obligations, prohibitions, and (strong or weak) permissions – including under any burden-of-persuasion constraints that may apply – fully supporting the modular argumentation model, i.e., allowing for theory fragmentation, thus enabling the coexistence of different modules. Arg-tuProlog allows the design of knowledge organised in distinct and separate modules that can “call” one another. In particular, a knowledge module can be used by itself, or by referring to another module. This second approach is done by directly calling and querying the relevant module.

2. The domain of private international law: an example

In this section, we will provide an example of interaction between national and transnational normative systems. We will focus on one of the EU’s main PIL instruments, the Brussels Regulation: while it provides common EU rules on jurisdiction and the recognition and enforcement of judgments, there are some cases where it points to national legislation for the relevant answer. We have built an example that focuses on the conflict between national laws, namely the Italian and Bulgarian. The legal texts and their representations are extracted from the work done in the context of the Interlex project.²

Example 1 (General jurisdiction rule) *In this example we consider article 3.1 of the Italian Law No. 218 of 31 May 1995 (Reform of the Italian System of Private International Law) and article 4 of the Bulgarian Law DB, bp. 42 of 17.05.2005 r. (Private International Law Code).*

Article 3.1 (Scope of jurisdiction) Italian courts shall have jurisdiction if the defendant is domiciled or resides in Italy or has a representative in this country who is enabled to appear in court pursuant to Article 77 of the Code of Civil Procedure, as well as in the other cases provided for by law. [...]

Thus Italian courts shall have jurisdiction if the defendant is domiciled or resides in Italy.

Article 4.1 (General Jurisdiction) The Bulgarian courts and other authorities shall have international jurisdiction where: the defendant has a habitual residence, statutory seat or principal place of business in the Republic of Bulgaria; [...]

Thus Bulgarian courts shall have jurisdiction if the defendant has an habitual residence, statutory seat, or the principal place of business in Bulgaria.

Let us consider, as a first scenario, the case of Marius, an Italian citizen with his primary residence in the city of Rome. Marius is summoned to appear in front of a judge to answer a complaint brought against him. Based on this information we can determine that the Italian court of Rome should be assigned jurisdiction in this complaint.

In a second scenario, Marius is also the owner of a business in Bulgaria. In this case, the Bulgarian PIL law – called by the Brussels Regulation – would assign jurisdiction to a Bulgarian court. Since both rules are valid, the jurisdiction in Marius’s case belongs

²The European project Interlex is aimed at developing a consultative and training system for internet-related PIL, making it available as an online platform.

to both the Italian and Bulgarian court. If no priority was set, then a conflict of laws would arise, with two equally valid indications of jurisdiction.

The example is now reified in the Arg-tuProlog framework to show the technology's effectiveness and potential. The Brussels Regulation, the Italian national law, and the Bulgarian national law have been mapped onto the Arg-tuProlog framework exploiting three distinct modules: one for the Brussels Regulation, one for the Italian national law, and one for the Bulgarian national law. In the following, we list an extract from the Brussels Regulation codification (*BrusselsRegulation.pl* module) that makes it possible to establish jurisdiction according to the content of the articles.

```
hasJurisdiction(Article, Country, Court, ClaimId):-
  personRole(PID, ClaimId, defendant), memberState(MemberLaw),
  call_module([MemberLaw, ClaimId],
  hasJurisdiction(Article, Country, Court, ClaimId)).
hasJurisdiction(Article, Country, Court, ClaimId):-
  claimObject(ClaimId, rightsInRem), memberState(MemberLaw),
  call_module([MemberLaw, ClaimId],
  hasJurisdiction(Article, Country, Court, ClaimId)).
```

The Italian law module – *italy.pl* – is a simple theory that includes the Prolog translation of the articles from the Italian PIL law as described above. The articles may be represented in the Arg-tuProlog system as follows:

```
hasJurisdiction(art3_1, italy, Court, ClaimId) :-
  personRole(PID, ClaimId, defendant), personDomicile(PID, italy, Court).
hasJurisdiction(art3_1, italy, Court, ClaimId):
  personRole(PID, ClaimId, defendant),
  personAgent(AgentId, PID), personDomicile(AgentId, italy, Court).
hasJurisdiction(art51, italy, Court, ClaimId):-
  claimObject(ClaimId, rightsInRem), immovProperty(ClaimId, italy, Court).
```

The Bulgarian national law is represented by the *bulgaria.pl* module which contains the Prolog translation from the Bulgarian PIL law as described in 2:

```
hasJurisdiction(art4_1, bulgaria, Court, ClaimId):-
  personRole(PID, ClaimId, defendant),
  personDomicile(PID, bulgaria, Court).
hasJurisdiction(art4_1, bulgaria, Court, ClaimId):-
  personRole(PID, ClaimId, defendant),
  personPlaceOfBusiness(PID, bulgaria, Court).
hasJurisdiction(art12, bulgaria, Court, ClaimId):-claimObject(ClaimId,
  rightsInRem), immovProperty(ClaimId, bulgaria, Court).
```

Let us consider the case discussed in Example 1. The facts and details of the case are stored in a separate module (*claim1.pl*), listed in the following.

```
personRole(marius, claim1, defendant).
personDomicile(marius, italy, rome).
personPlaceOfBusiness(marius, italy, rome).
memberState(bulgaria).
memberState(italy).
```

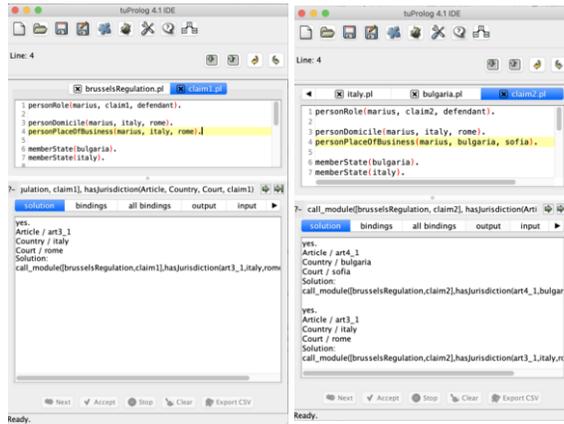


Figure 1. Arg2p interface: result of claim1 (left) and claim2 (right).

To evaluate the case, we can select the jurisdiction simply by calling the following goal over the top module *brusselsRegulation.pl*:

```
call_module([brusselsRegulation, claim1],
            hasJurisdiction(Article, Country, Court, claim1)).
```

Figure 1 (left) shows the result, which is that under article 3.1 of the Italian law, the court to which the case is assigned is in Rome, Italy. The result is perfectly consistent, since the defendant is domiciled in Italy (and also his place of business). Let us now consider the same case, with the only difference that the place of the defendant's business is in Sofia, Bulgaria (*claim2.pl*).

```
personRole(marius, claim2, defendant).
personDomicile(marius, italy, rome).
personPlaceOfBusiness(marius, bulgaria, sofia).
memberState(bulgaria).
memberState(italy).
```

As shown in Figure 1 (right), the answer in this case is twofold. Article 4.1 of the Bulgarian law and Article 3.1 of the Italian law should apply at the same time, assigning jurisdiction to the Sofia (Bulgarian) court in one case and the Rome (Italian) court in the other. The system makes it possible to detect and point out this inconsistency, indicating that two different articles, with different answers in the matter of jurisdiction, should apply simultaneously.

References

- [1] P.M. Dung and G. Sartor, The modular logic of private international law, *Artificial Intelligence and Law* **19**(2–3) (2011), 233–261. doi:10.1007/s10506-011-9112-5.
- [2] G. Pisano, R. Calegari, A. Omicini and G. Sartor, Arg-tuProlog: a tuProlog-based argumentation framework, in: *CILC 2020 – Italian Conference on Computational Logic. Proceedings of the 35th Italian Conference on Computational Logic*, CEUR Workshop Proceedings, CEUR-WS, Rende, CS, Italy, 2020.