University of Udine, Italy

XAI-LAW Towards a logic programming tool for taking and explaining legal decisions

Agostino Dovier Talissa Dreossi Andrea Formisano
Department of Mathematics,
Computer Science
and Physics

May 14, 2024

Introduction: Leibniz dream

Gottfried Wilhelm von Leibniz 1646–1716

He imagined that this machine, which he called “the great instrument of reason,” would be able to answer all questions and resolve all intellectual debate. “When there are disputes among persons,” he wrote, “we can simply say, ‘Let us calculate,’ and without further ado, see who is right.”

The (also recent) literature on legal reasoning is huge.

Disclaimer

Slovenian Wheel $\approx \pi \cdot 10^3\ BC$

Slovenian Wheel 2024 AD

We are not reinventing the wheel,
we are trying to improve it

Summary

- We present a project investigating legal (semi) automatic reasoning in the Italian criminal system
- Italian Criminal Laws are modeled in ASP; the model obtained is tested on a set of previous statements on the crimes, and, if needed, refined.
- Decisions on a new case can be suggested by the system and explained using a tool that exploits "supportedness" of stable models.
- In the same way, the decision of a judge can be input in the system and automatically explained.
- Using a system of inductive logic programming for ASP, the tool can evolve by analyzing new statements and performing model revision, by learning exceptions, and by applying rule generalization.
- To study feasibility of the approach we analyzed the crimes of theft, robbery, and personal injuries.

Italian legal code (Example)

Article 624 (theft)

Whoever takes possession of another person's movable property, **capturing it from its owner**, in order to gain profit for himself or others [...]

Article 624 bis (theft by snatching)

Whoever takes possession of another person's movable property, capturing it from the person holding it, in order to gain profit for himself or others, **snatching it out of the person's hand or from the person's body** [...]

Article 628 (robbery)

Whoever, in order to procure for himself or others an unjust profit, **by means of violence to the person or threat**, takes possession of another person's movable property, capturing it from the person who has it [...]

Italian legal code (Example)

Article 624 (theft)

Whoever takes possession of another person's movable property, **capturing it from its owner**, in order to gain profit for himself or others [...]

Article 624 bis (theft by snatching)

Whoever takes possession of another person's movable property, capturing it from the person holding it, in order to gain profit for himself or others, **snatching it out of the person's hand or from the person's body** [...]

Article 628 (robbery)

Whoever, in order to procure for himself or others an unjust profit, **by means of violence to the person or threat**, takes possession of another person's movable property, capturing it from the person who has it [...]

Italian legal code (Example)

Article 624 (theft)

Whoever takes possession of another person's movable property, **capturing it from its owner**, in order to gain profit for himself or others [...]

Article 624 bis (theft by snatching)

Whoever takes possession of another person's movable property, capturing it from the person holding it, in order to gain profit for himself or others, **snatching it out of the person's hand or from the person's body** [...]

Article 628 (robbery)

Whoever, in order to procure for himself or others an unjust profit, **by means of violence to the person or threat**, takes possession of another person's movable property, capturing it from the person who has it [...]

ASP Encodings (by hand)

- First part: encodings of articles (e.g., 624, 624 bis and 628) "Whoever takes possession of another person's movable property, capturing it from its owner[...]"

```
theft(R, V, C)  :- subtract(R, C), own(V, C),
                   theft_intention(R), take_possession(R, C),
                   agent(V), agent(R), res(C), V!=R.
```

- Second part: encodings decisions of the Court of Cassation Sentence n. 49832, 11th of December 2013, Section 2, Criminal Law, Court of Cassation

```
person_violence(R, V) :-  tight_physical_adherence(V, C),
                   substract(R, C),agent(V), agent(R),
```

ASP Encodings: room for non-determinism

No need of convincing this audience that negation and stable model semantics is useful (soimetimes needed) for knowledge representation.

If a heap is reduced by a single grain at a time, the question is: at what exact point does it cease to be considered a heap?

Vagueness

Vagueness is qualified when information is available, yet indeterminacy arises from a semantic point of view.

Vagueness encodings (our choice)

Tight/loose adherence modelled assigning levels:

```
level(1..4).


tight_physical_adherence(S, C) :-
        adherence(S, C, L), agent(S), res(C),
        L>2.

loose_physical_adherence(S, C) :-
        adherence(S, C, L), agent(S), res(C),
        L<3.

1{ adherence(S, C, L) : level(L) }1 :-
        unknown_adherence(S, C).
```

Computing different models

Example

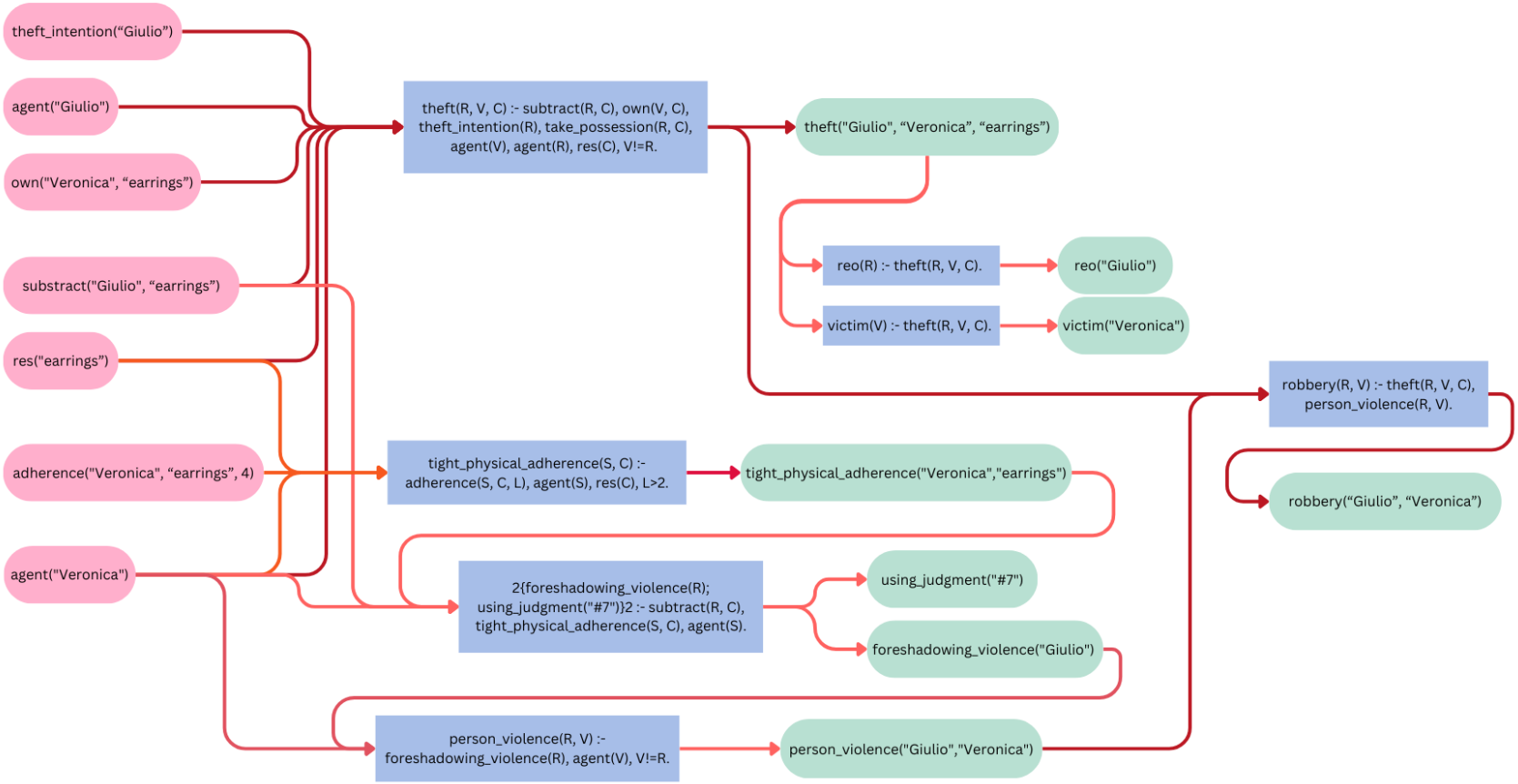
Decision n. 17348, 18th April 2014, Section II, Court of Cassation

```
agent("Mario").
agent("Giorgio").
theft_intention("Mario").
subtract("Mario", "necklace").
res("necklace").
own("Giorgio", "necklace").
unknown_adherence("Giorgio",
"necklace").
```

The ASP-solver output 4 stable model containing:

- `adherence("Giorgio","necklace", 1).`
  `theft_snatch("Mario","Giorgio").`
- `adherence("Giorgio","necklace", 2).`
  `theft_snatch("Mario","Giorgio").`
- `adherence("Giorgio","necklace", 3).`
  `robbery("Mario","Giorgio").`
- `adherence("Giorgio","necklace", 4).`
  `robbery("Mario","Giorgio").`

Explainability (inherited from xASP)



Explainability can be applied to decision taken by ASP and/or by humans (adding a denial: "it is impossible that this verdict does not hold").

Inductive Learning (with ILASP)

A learning task is a tuple (B, S, E^+, E^-) where

- B (background knowledge) is a Logic (ASP) Program
- E^+ and E^- are the set of positive and negative examples
- S is a set of rules that denotes the hypothesis space, This is the main difference wrt sub-symbolic learning framework: we must help the learning.

ILASP computes $H \subseteq S$ coherent with B, E^+, E^-.

S can be given explicitly or implicitly via mode declarations, such as: `modeh modeha modeb` which state that a certain atom can appear in the head, in the head as an aggregate, in the body of a rule. ILASP is called with a bound on the number of literals in the body for each learned rule.

Inductive Learning (with ILASP)

A learning task is a tuple (B, S, E^+, E^-) where

- B (background knowledge) is a Logic (ASP) Program
- E^+ and E^- are the set of positive and negative examples
- S is a set of rules that denotes the hypothesis space, This is the main difference wrt sub-symbolic learning framework: we must help the learning.

ILASP computes $H \subseteq S$ coherent with B, E^+, E^-.

S can be given explicitly or implicitly via mode declarations, such as: `modeh` `modeha` `modeb` which state that a certain atom can appear in the head, in the head as an aggregate, in the body of a rule. ILASP is called with a bound on the number of literals in the body for each learned rule.

Inductive Learning (with ILASP)

```
lesioni(V1, V2) :- intent_to_harm(V1, V2); cause_illness(V1, V2).
percosse(V1, V2) :- not cause_illness(V1, V2);
                    intent_to_harm(V1, V2); cause_pain(V1, V2).
cause_illness(V1, V2) :- illness(V3); cause(V1, V2, V3).
cause_illness(V1, V2) :- item(V4); cause(V1, V2, V3, V4); illness(V3).
cause_pain(V1, V2) :- cause(V1, V2, pain).
cause_pain(V1, V2) :- item(V4); cause(V1, V2, pain, V4).
illness(V3) :- physical_illness(V3).
illness(V3) :- mental_illness(V3).
```

Search space of 861 rules, plus background knowledge, ILASP runs in 30 seconds (averaged over 100 runs).

Conclusions and Future Works

In the future the system XAI-LAW (read “ASSAI”) would:

- Assist judges in having possible scenarios to choose from,
- Help lawyers constructing situations of ambiguity that could lead to minor penalties

Novelties:

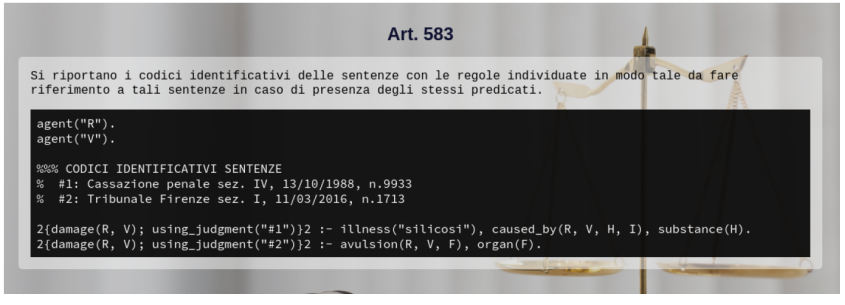
- Using/Specializing XASP tools for explainabiliy (need to be checked with lawiers)
- Using ILASP (Inductive Learning of Answer Set Programs) to learn on previous judgments and automatically/semi automatically evolving the ystem

QUESTIONS?


Verso l'automazione del Ragionamento Legale
Articoli
Sentenze


Home
Articoli
Sentenze
Libro Secondo, Titolo XII - Dei delitti contro la persona:
Sentenze artt. 575 - 579
Sentenze artt. 581 - 582
Sentenze art. 583
Sentenze artt. 584 - 588
Sentenze artt. 589 - 589 bis
Sentenze artt. 595 - 609 bis
Sentenze artt. 610 - 614
Libro Secondo, Titolo XIII - Dei delitti contro il patrimonio:
Sentenze artt. 624 - 624 bis - 628
https://benedettas.github.io/ragionamentolegale/sentenzeartt575579.html


Home
Sentenze
Articolo 575
Codice Penale Art. 575 - Omicidio
Chiunque cagiona la morte di un uomo è punito con la reclusione non inferiore ad anni ventuno.
reo(R, "murder") :- murder(R, V).
victim(V, "murder") :- murder(R, V).
murder(R, V) :- death_intention(R), not consent(V), cause_death(R, V), agent(R), agent(V), R != V.
Articolo 579
Codice Penale Art. 579 - Omicidio del consenziente
Chiunque cagiona la morte di un uomo, col consenso di lui, è punito con la reclusione da sei a quindici anni.
reo(R, "consenting_murder") :- consenting_murder(R, V).
victim(V, "consenting_murder") :- consenting_murder(R, V).
consenting_murder(R, V) :- death_intention(R), consent(V), cause_death(R, V), agent(R), agent(V), R != V.


Art. 583
Si riportano i codici identificativi delle sentenze con le regole individuate in modo tale da fare riferimento a tali sentenze in caso di presenza degli stessi predicati.
agent("R").
agent("V").
%%% CODICI IDENTIFICATIVI SENTENZE
% #1: Cassazione penale sez. IV, 13/10/1988, n.9933
% #2: Tribunale Firenze sez. I, 11/03/2016, n.1713
2{damage(R, V); using_judgment("#1")}2 :- illness("silicosi"), caused_by(R, V, H, I), substance(H).
2{damage(R, V); using_judgment("#2")}2 :- avulsion(R, V, F), organ(F).