Preference-based argumentation in practice

Nikolaos Spanoudakis, PhD
School of Production Engineering and Management,
Applied Mathematics and Computers Laboratory,
Technical University of Crete
@: nispanoudakis@tuc.gr
www: http://users.isc.tuc.gr/~nispanoudakis

With special thanks to Prof. A. Kakas (University of Cyprus)

CONTENTS

- Abstract Argumentation and defeasible logic
- Preference-based argumentation - Gorgias
- Methodology – Software Development for Argumentation
- Application Level Explanations for Argumentation-based Decision Making

Abstract Argumentation and Defeasible Logic

Why do I need argumentation for my applications?

Which argument prevails?

Human like Systems
Why Argumentation?

- Argumentation is native to human reasoning
  - Cognitive Psychology
    - "The function of human reasoning is argumentative" (Mercier & Sperber, 2011)
  - Behaviour Economics
    - "human judgment diverges from rational expectations" (Thaler, 2016)
- Knowledge captured as arguments

Logic Programming and Logic

- Logic programming's foundation is classic/mathematical logic
  - Strict/Formal logic
- It is adequate for problems with strict requirements
  - Like any other framework/programming language!

Strict Logic

- A statement either is or isn’t a logical conclusion
  - If a statement is a logical conclusion (or solution to a problem) then it is still a logical conclusion when we add any new knowledge!
  - E.g. Once proven, mathematical theorems hold forever!
  - Thus, we say that classical logic is monotonic

Remember VIKI from “I robot”?

Strict Logic

- A statement either is or isn't a logical conclusion
  - If a statement is a logical conclusion (or solution to a problem) then it is still a logical conclusion when we add any new knowledge!
  - E.g. Once proven, mathematical theorems hold forever!
  - Thus, we say that classical logic is monotonic
- However, when we reason with common sense, new information leads us to change our conclusion
  - non monotonic reasoning (McCarthy, 1980)

Non monotonic logic

- Common sense rules are not strict
  - They are "For the most part" or "Usually" rules DEFAULT RULES
  - A rule, p :- q, is interpreted as (prolog notation)
    - "Usually, if we know that q holds then p holds"
    - fly(X) :- bird(X) , holds "for the most part"

Defeasible Knowledge

- Properties of the world
  - “Usually, birds fly”
    - “Usually, penguins don’t fly”
    - “Usually, nestlings don’t fly”
    - “Usually, hurt birds don’t fly”
- Results of actions
  - “Usually, when we shoot a bird, it is hurt”
    - “Usually, when the gun is not loaded, the bird is not hurt”
      - “Usually, the gun is loaded”
    - “Usually, when the bird is far away, it is not hurt”

What color is my banana?

What color is my banana?

- I have it in my bag…

What color is my banana?

- I have it in my bag…

What color is my banana?

- I have it in my bag…
- It is too ripe…

What color is my banana?

- I have it in my bag…
- It is too ripe…

What color is my banana?

- I have it in my bag…
- It is too ripe…
- I spilled blue color on it…

What color is my banana?

- I have it in my bag…
- It is too ripe…
- I spilled blue color on it…

Defeasible Knowledge (2)

- Results of actions
  - “Usually, when we move something, then it gets at a new position”
  - at(Object, Pos2) :- move(Object, Pos1, Pos2)
    Default Rule!
- State maintenance – Knowledge inertia
  - at(Object, Pos, T2) :- at(Object, Pos, T1) , T2>T1.
    - E.g. at(my_car, car_park, 5pm) :- at(my_car, car_park, 9am)
- Knowledge inertia for any property:
  - holdsAt(Property,T2) :- holdsAt(Property,T1), T2>T1

What is an argument?

- An argument is a link between
  - Some premises
  - A conclusion supported by it

- Premises → Conclusion

Fundamental Concept – Valid argument

- Based on the informal meaning:
  - "A valid argument is one whose counter-arguments are not valid"
  - "A valid argument is one whose counter-arguments are, or rendered by it, not valid"
- Formalized through Abstract Argumentation: <Args, Attack> (or <Arg,Att,Def>) from AI
  - Args is a set of arguments
  - Attack (and Defense) is (are) the counter-argument relation

Acceptable Arguments

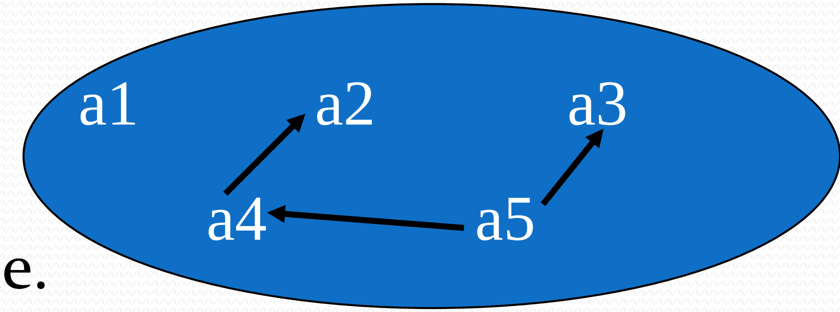
- Informally…
- What is a good/acceptable argument?
  - An argument that builds a coherent case for its position.
  - An argument that can defend itself against all its counter-arguments
    - An argument that renders its counter-arguments incoherent/invalid
  - We call this an **Admissible** argument

Abstract Argumentation (1)

- An abstract argumentation framework is a pair of a set T of arguments and an attacking relation on arguments
  - AF=<Args, Att>, where Att is a binary relation on Args
- S $\subseteq$ Args is an Admissible Argument iff
  - S it does not attack itself (i.e. it is conflict free), and
  - S attacks (counter-attacks/defends) all its attacks
- This is a simple but powerful definition

Abstract Argumentation (2)

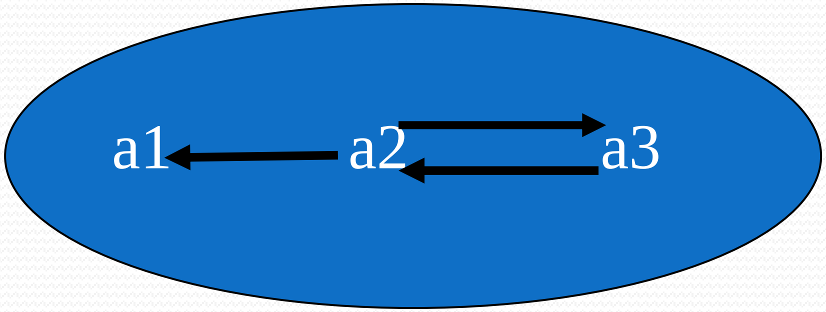
- S $\subseteq$ Args is an Admissible Argument iff
  - S it does not attack itself (i.e. it is conflict free), and
  - S attacks (counter-attacks) all its attacks



- Example
  - {a2} and {a3} are not admissible.
  - But {a2, a5} is admissible.
  - {a1}, {a5} are admissible.
  - {a1,a2,a5} is maximally admissible.

Abstract Argumentation (3)

- S $\subseteq$ Args is an Admissible Argument iff
  - S it does not attack itself (i.e. it is conflict free), and
  - S attacks (counter-attacks) all its attacks
- Quiz: Which arguments are admissible?



  - {a2} and {a3} are admissible.
  - {a1} is not admissible.
  - {a1, a3} is admissible (maximally).

Example of Argumentation

- "Sellers who deliver on time are trustworthy"
  - a1={trusted(Seller) :- timely(Seller)}
- "Sellers who deliver wrong are not trustworthy"
  - a2={¬ trusted(Seller) :- wrong_delivery(Seller)}
- Suppose we "observe":
  - timely(bob): a1 supports trusted(bob).
  - wrong_delivery(bob): a2 supports ¬trusted(bob).
  - a1 attacks a2 and vice-versa.
- "Sellers who are trusted get large orders"
  - a= {large_orders(X) :- trusted(X)}
  - A={a1, a} supports large_orders(bob)
  - B={a2} attacks A - **B undercuts A**

Building admissible arguments Dialectics

- Dialectic Nature of Validity:
  - Consider attacks and find defences.

1. Find an argument Δ that supports the position (query) we want.
2. Check Δ is not self-attacking.
3. Consider attacks, A, against Δ.
4. Attack/Defend each A by argument D.
5. Add D to Δ to give new Δ'= Δ U D.
6. Repeat from 2nd step with Δ'.

Argumentation: Foundations

- Logical Entailment via argument acceptability:
  - Existence of an acceptable argument for conclusion ϕ.
    - Credulous entailment
  - Non-Existence of an acceptable argument for $\neg\phi$.
    - Sceptical entailment
- Classical Logic can be used as a realization of Abstract Argumentation

Preference Based Argumentation - Gorgias

Each set of rules is an argument for the (logical) conclusions it supports.

Arguments supporting contrary conclusions are conflicting – are counter-arguments.

Which argument prevails?

Preference Based Argumentation

- Logic Programming Rules & Priorities
- An extension of Logic Programming
- Arguments are sets of rules
- Attacks between arguments are defined via:
  - Conflicts between conclusions of arguments
  - Strength relation on the subsets of rules, used in each argument to derive the conflicting conclusion, based on the priority relation between the individual rules in the subsets.

An Example

(r1): fly(x)←bird(x)

(r2): ¬fly(x) ←penguin(x)

(r3): penguin(x) ←walkslikepeng(x)

(r4): ¬penguin(x) ← ¬flatfeet(x)

(r5): bird(x) ← penguin(x)

(r6): bird(tweedy)

(r7): walkslikepeng(tweedy)

(r8): ¬flatfeet(tweedy)

? fly(tweedy**)**

Argument for:
A1 ={r6, r1}

Against A1:
A2 ={r7, r3, r2}

Against A2:
A3 = {r8, r4}

Yes, fly(tweedy**) can be supported by A1 U A3. (credulous)**

With preferences (priorities)

(r1): fly(x)←bird(x)

(r2): ¬fly(x) ←penguin(x)

(r3): penguin(x) ←walkslikepeng(x)

(r4): ¬penguin(x) ← ¬flatfeet(x)

(r5): bird(x) ← penguin(x)

(r6): bird(tweedy)

(r7): walkslikepeng(tweedy)

(r8): ¬flatfeet(tweedy)

(r9): r2 > r1

(r10): r4 > r3

? fly(tweedy**)**

Argument for:
A1 ={r6, r1}

Against A1:
A2 ={r7, r3, r2, r9}

Against A2:
A3 = {r8, r4, r10}

Yes, fly(tweedy**) can be supported by A1 U A3. (skeptical)**

The Attacking Relation

- Specifies when a subset S1 (of the given theory T) attacks another subset S2
- An attacking relation is realized via:
  - 1) Inconsistent conclusions
  - 2) A local Priority Relation (<):
    - Encodes locally the relative strength of sentences/rules in the theory: r < r' means that r has lower priority than r'.
    - This lifts up to a global strength relation on arguments
    - It can be reasoned, just like any other predicate

Argumentation Summary

- An argument is:
  - A set of sentences/rules, S, in some background logic (L, ⊢): from which we can derive a conclusion (i.e. $S \vdash \phi$)
- Attacking Relation:
  - Specifies that one argument (i.e. a set S1 of rules) attacks another S2 when they have some contrary conclusion and S1 is “as strong” as S2.
- An argument S is Admissible:
  - S is conflict free (i.e. it does not attack itself) and
  - S attacks (counter-attacks) all its attacks
- Credulous or Skeptical Reasoning:
  - A conclusion holds in one or all admissible/acceptable extensions

The Gorgias framework

Gorgias is a general argumentation framework that combines the ideas of preference reasoning and abduction (Kakas and Moraitis, 2003)

http://www.cs.ucy.ac.cy/~nkd/gorgias

Decision Making in Argumentation

- A decision problem consists of:
  - A set of *Options*.
  - A set of *Values* that parametrize the options.
  - *Object level Arguments*: a structure, e.g., a rule of conditions (could be empty) that makes an option available or not in a given situation.
  - *Preferences* that give relative strength to the arguments for the various options

Writing rules in Gorgias

- The language for representing the theories is given by rules with the syntax in formula:

  rule(Signature, Head, Body).

- where *Head* is a literal, *Body* is a list of literals and *Signature* is a compound term composed of the rule name with selected variables from the Head and Body of the rule.

Adding preferences

- The predicate *prefer/2* is used to capture the higher priority relation ">" defined in the theoretical framework. It should only be used as the head of a rule. Using the rules syntax we can write:

  rule(Signature, prefer(Sig1, Sig2), Body).

- which means that the rule with signature *Sig1* has higher priority than the rule with signature *Sig2*, provided that the preconditions in the *Body* hold

Complements

- A literal's negation is considered by default as conflicting with the literal itself. A negative literal is a term of the form

  neg(L).

- There is also the possibility to define conflicting predicates that are used as heads of rules using the *complement/2* predicate:

  complement(Head1, Head2).

Methodology – Software Development for Argumentation (SoDA)

facilitates the principled modeling of real life problems

Decision Making via Argumentation

- Policy Options, e.g. different levels of access
- Policy Preferences
  - Dynamic preferences over changing environment of the application of the policy
  - Multi-Level preferences over different CONTEXTS of policy
- General form of Preferences:
  - "Normally, in SITUATION prefer O_i, but in particular CONTEXT prefer O_j."
    - "Generally, don't give access but for the owner give full access."
    - "Generally, allow full access to owner but when critical tests suspend access. "

Medical Data Access

- Requirements:
  - *Generally, don't give access but for the owner give full access.*
  - *Generally, allow full access to owner but when he is taking critical tests suspend access.*
- Code in GORGIAS:

```
rule(d1(Agn), access(Agn, DataID, full_access), []):- owner(Agn,
   DataID), dataItem(DataID).
rule(d2(Agn), access(Agn, DataID, no_access), []):- dataItem(DataID).
rule(hpr_12(Agn), prefer(d1(Agn), d2(Agn)),[]).
rule(hpr_21(Agn), prefer(d2(Agn), d1(Agn)),[critical_tests(Agn)]).
rule(hpr_21_12(Agn), prefer(hpr_21(Agn), hpr_12(Agn)),[]).
```

SoDA Methodology

(Kakas et al., 2019; Spanoudakis et al., 2016)

- In SoDA we consider the following ordered questions:
  1. What is the decision problem? What are the options?
  2. What are the object level arguments (what conditions unlock the options, also type the parameters)?
  3. What are the possible scenarios given the object-level arguments?
  4. What are the contexts that refine the scenarios?
  5. Is the model/representation complete?
  6. How do we extend the model?
     - With new refined contexts (in existing scenarios)
     - With new scenarios.

Application Level Explanations for Argumentation-based Decision Making

Work presented at ArgXAI 2022

Nikolaos I. Spanoudakis[1], Antonis C. Kakas[2] and Adamos Koumi[2]
[1]School of Production Engineering and Management, Technical University of Crete
[2]Department of Computer Science, University of Cyprus

Outline

- Introduction - Contributions
- Building explanations from Gorgias preference-based argumentation framework results
- Application examples
- Conclusion and Future Work

Argumentation and eXplainable Artificial Intelligence

- Argumentation and eXplainable Artificial Intelligence (XAI) are closely related (Vassiliades et al., 2021)
- The application of computational argumentation to XAI is supported by its strong theoretical and algorithmic foundations and the flexibility it affords (a wide variety of argumentation frameworks).
- AFs include ways to specify arguments and dialectical relations between them, as well as semantics to evaluate the dialectical acceptability or strength of arguments, while differing (sometimes substantially) in how they define these components (Cyras et al., 2021)

Contributions

- We show how the returned results of the dialectical argumentation reasoning within the Gorgias framework can be exploited to provide human-readable explanations that are
  - Attributive
  - Contrastive
  - Actionable
- These results, can be manipulated by applications to produce case-based human readable explanations.

A gorgias theory example (requirements)

- An agent will normally buy an article that it needs. In the case that it is low on funds, the agent will not buy an item that is not urgently needed.

A gorgias theory example (code)

```
rule(r1(X), buy(X), []):- need(X).
rule(r2(X), neg(buy(X)), [neg(urgentNeed(X))]).
rule(pr1(X), prefer(r2(X), r1(X)), [lowOnFunds]).
rule(pr2(X), prefer(r1(X), r2(X)), []).
rule(c1(X), prefer(pr1(X), pr2(X)), []).
abducible(urgentNeed(X), []).
abducible(neg(urgentNeed(X)), []).
```

A gorgias query example

- Suppose that we know that the agent needs a bag and it believes it is low on funds
  - need(bag)
  - rule(b1, lowOnFunds, [])
- Let's try the query to not buy the bag
  - neg(buy(bag))
- It is valid, the argument is
  - ass(neg(urgentNeed(bag))), c1(bag), b1, pr1(bag), r2(bag)

> The way to generate a belief. It can be argued for or against

Explanation generator

- Internal explanation:
- b1, r2(bag), pr1(bag), c1(bag), ass(neg(urgentNeed(bag)))
  - rule(r1(X), buy(X), []):- need(X).
  - rule(r2(X), neg(buy(X)), [neg(urgentNeed(X))]).
  - rule(pr1(X), prefer(r2(X), r1(X)), [lowOnFunds]).
  - rule(pr2(X), prefer(r1(X), r2(X)), []).
  - rule(c1(X), prefer(pr1(X), pr2(X)), []).
  - abducible(urgentNeed(X), []).
  - abducible(neg(urgentNeed(X)), []).

Attributive

Contrastive

Actionable

Explanation generator

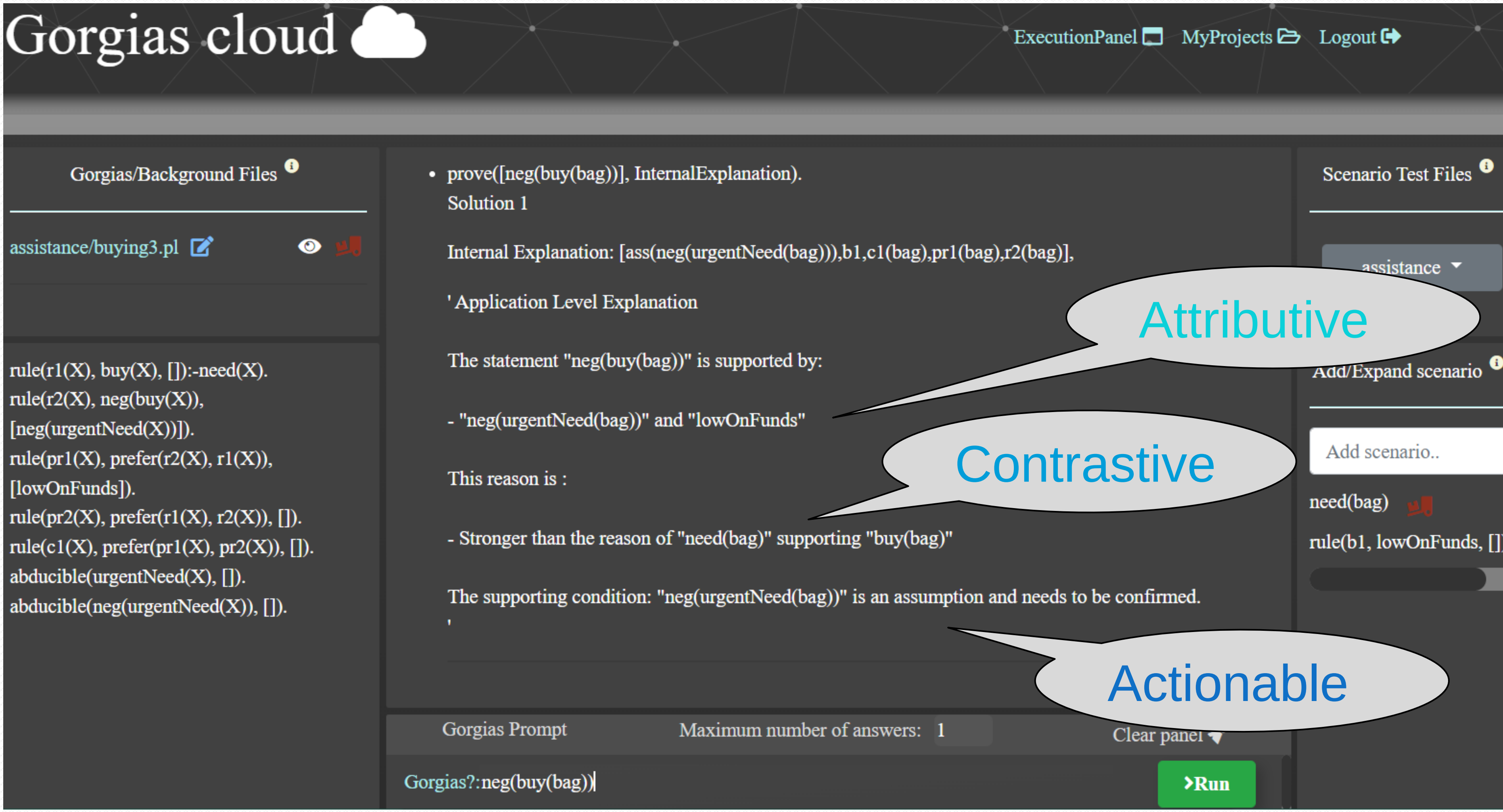
Attributive

Contrastive

Actionable

- Internal explanation:
- b1, r2(bag), pr1(bag), c1(bag), ass(neg(urgentNeed(bag)))
- The statement "neg(buy(bag))" is supported by: - "neg(urgentNeed(bag))" and "lowOnFunds"
- This reason is : - Stronger than the reason of "need(bag)" supporting "buy(bag)"
- The supporting condition: "neg(urgentNeed(bag))" is an assumption and needs to be confirmed.

Gorgias Cloud Application-Level Explanations



A Social Media Application Example

- The agent browses social media content
- Sets each item's priority to:
  - Important
  - Normal/default
  - Hide
- Policy:
  - … Posts that come from the user's manager are **important** regardless of whether they are positive or negative. … **Hide** politics posts from the user's manager when negative. …

A Social Media Application Example

Social Media Assistant

Post Title	Presidential election
Post Characteristics	• The post comes from my manager. • The post is on politics. • The post topic is negative.

Hide post

Explanation:

Generally when the post comes from your manager, we prefer to set the priority to important, but because **the post topic is in on politics and it is negative** **<u>we hide the post.</u>**

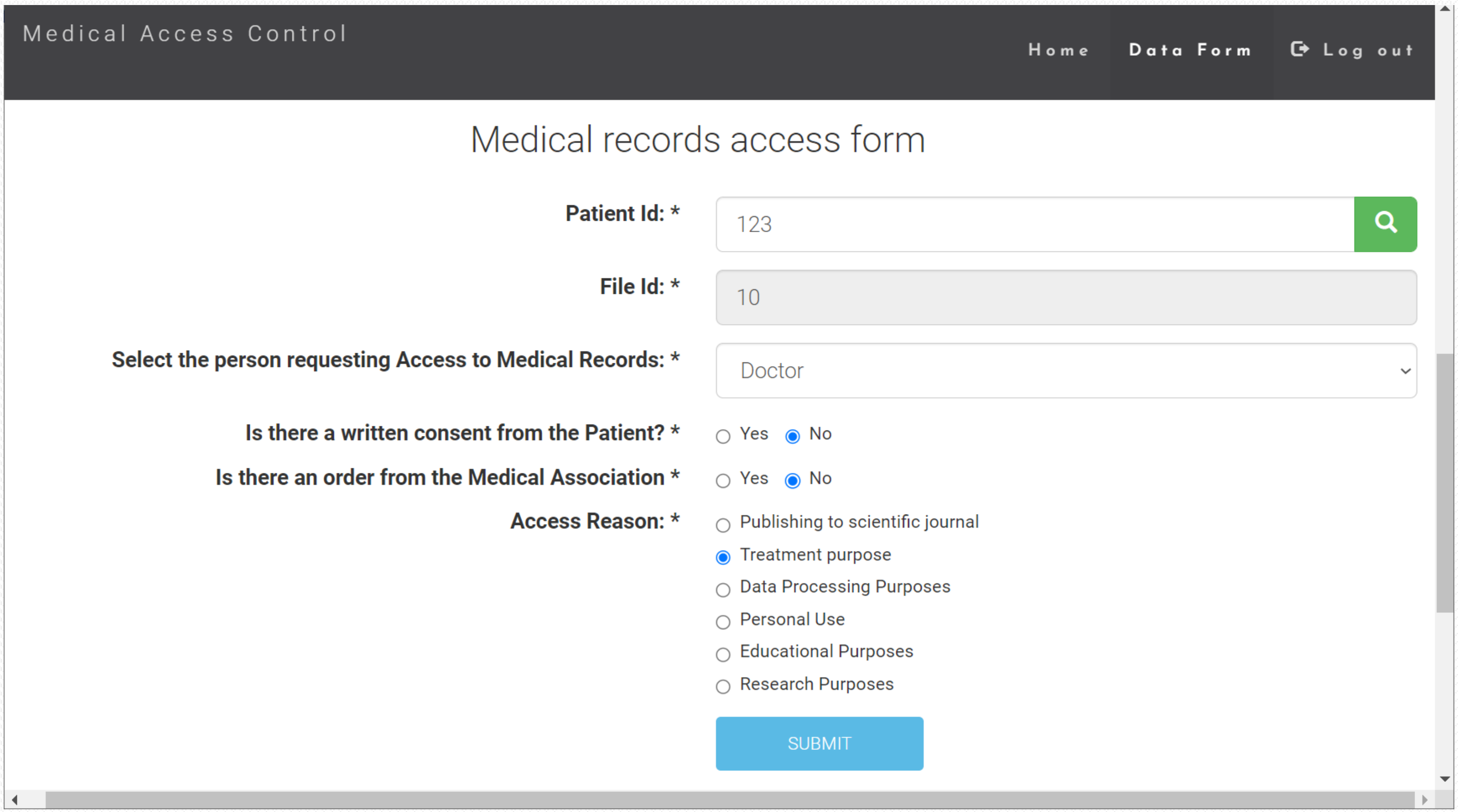
Contrastive

Attributive

Application Example 1: Medica

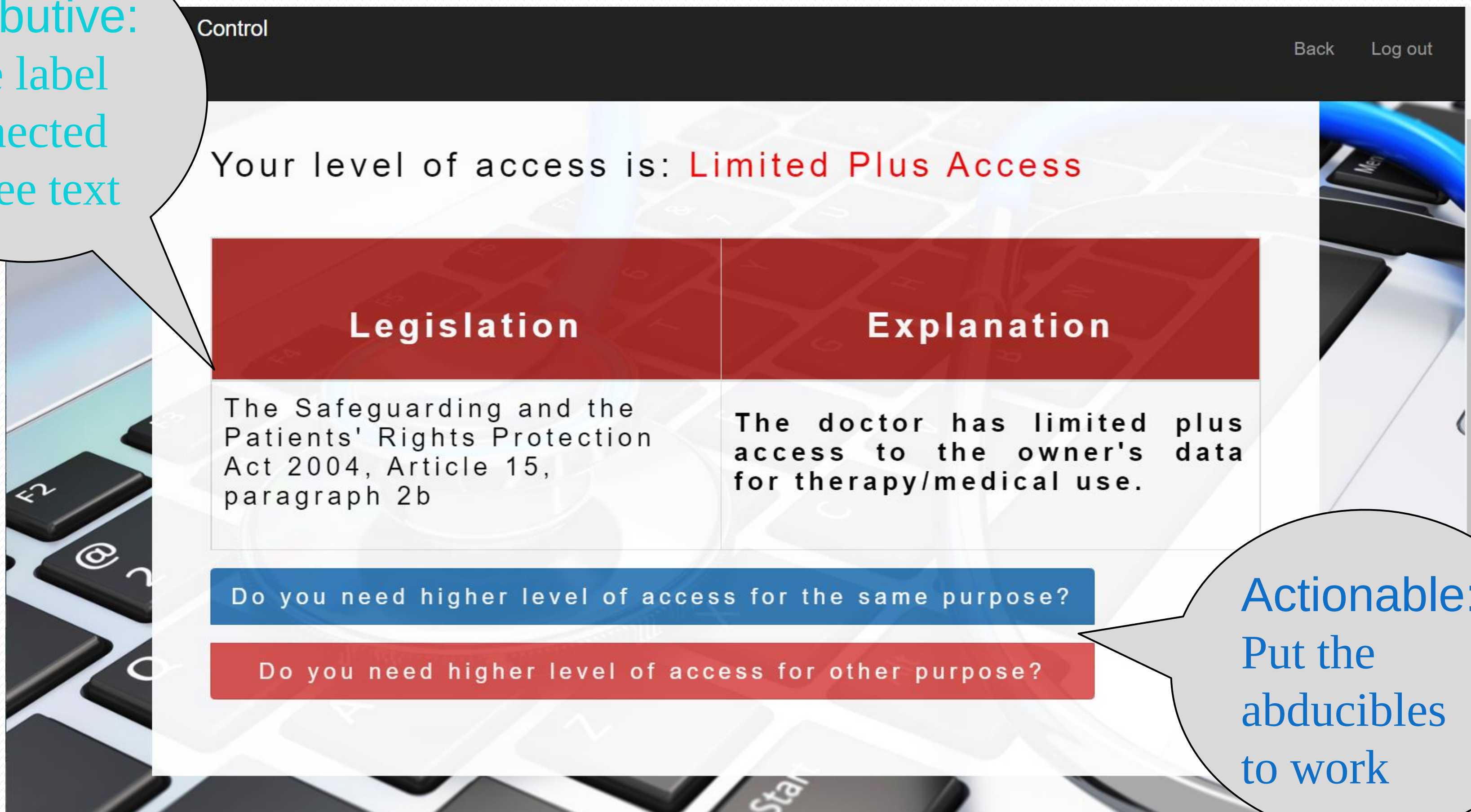
- This system aims to aid the decisions of administrative personnel in the health domain
- They need to decide what information can be disclosed to a person asking for it
- EU legislation defines the access rights

Application Example 1: Medica



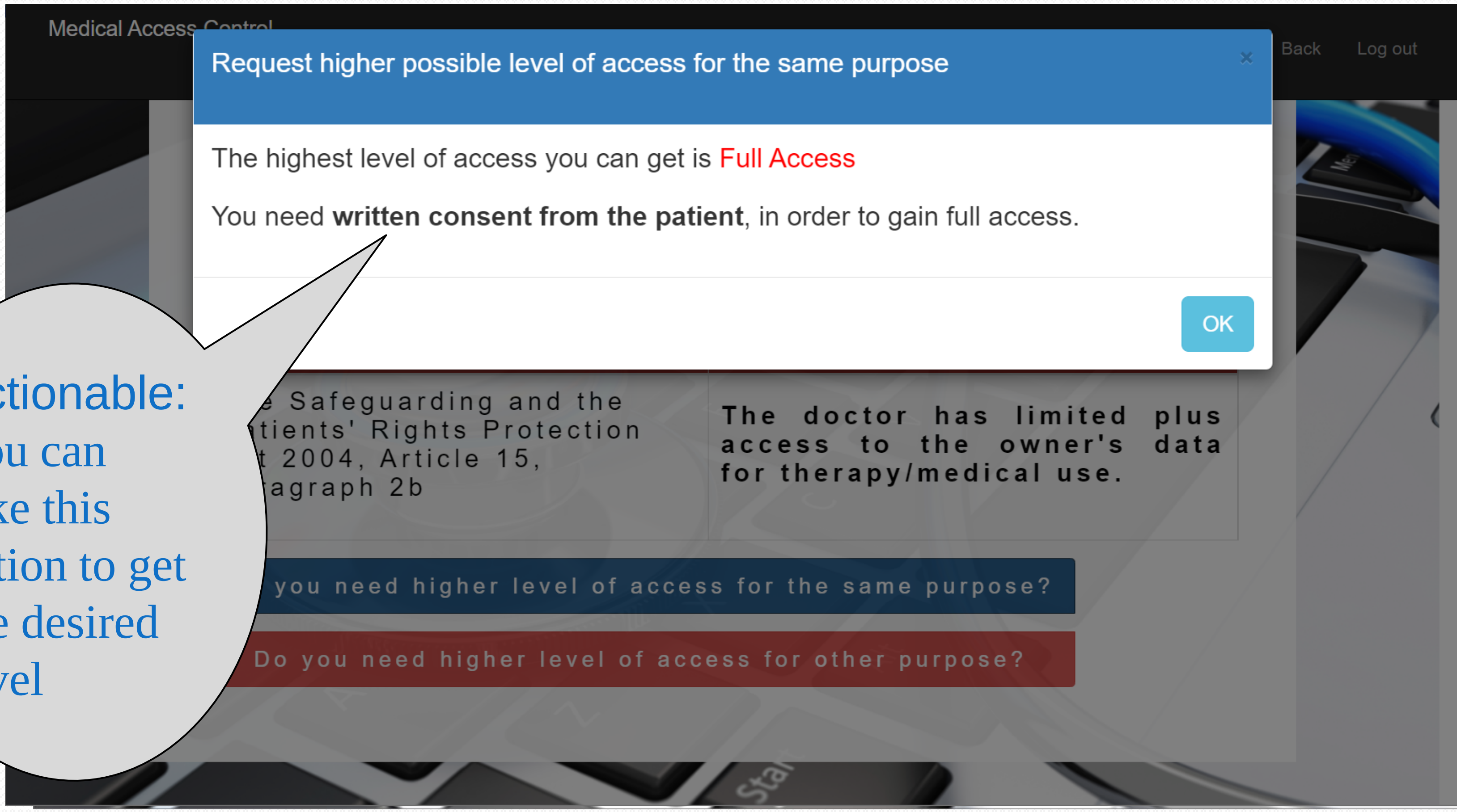
Application Example 1: Medica

Attributive:
Rule label connected to free text



Actionable:
Put the abducibles to work

Application Example 1: Medica

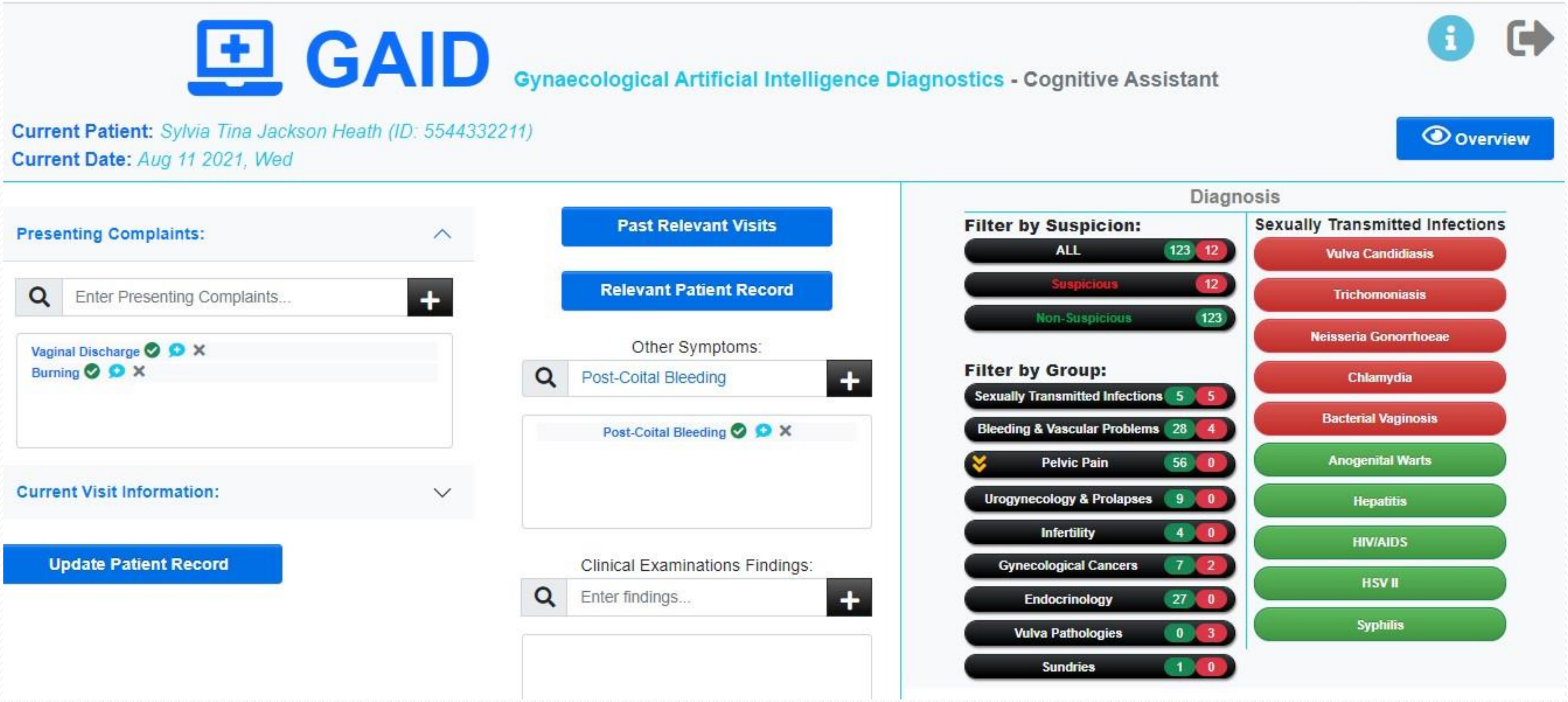


Actionable: You can take this action to get the desired level

Application Example 2: GAID

- GAID: Gynecological AI Diagnostic Assistant
- This system aims to aid a medical doctor in disease diagnosis
- The system gets the tests and symptoms of a patient
- Doctor's knowledge determines the outcome

Application Example 2: GAID



Application Example 2: GAID

Diagnosis:

Anogenital Warts

Attributive

EXPLANATION | RELATED INFO | ICD-10

Under the information **Vaginal Burning** it is ***recommended*** that you investigate **Anogenital Warts**.

This decision is ***supported*** by: **Inter-Menstrual Bleeding**.

The following further information ***strengthens*** this decision: **Dysuria**.

But each of the following information:

- **Image: No indication of small cauliflower-shaped lumps**
- **History: Vaccinated with HPV before first intercourse**

indicates, that this disease may not be possible and could be **excluded**.

Contrastive

Actionable: Possible issues to investigate in order to exclude other possible diseases

Argumentation-based Decision Making as a Service

- The previously presented applications used the Gorgias Cloud as a service (Spanoudakis et al., 2023).
- Thus, the apps call a service submitting to it the context, the query and the gorgias file they want to use.
- Similar services (non xAI enabled) are offered by other research groups such as
  - arg-tech.org (Reed et al., 2017; Snaith and Reed, 2012)
  - tweetyproject.org (Thimm, 2017)

Conclusions

- We have delved into explainable AI-base decisions that are
  - Attributive
  - Contrastive
  - Actionable
- Still more work is needed
  - NLP: generate predicates and arguments from human generated text
  - Generate arguments explanations (Attributive, Contrastive or Actionable) in free text

Register to Gorgias Cloud for the hands-on session tomorrow

The Gorgias Cloud System is open for academic use. Register at:

https://aiasvm1.amcl.tuc.gr:8087/

After registering check your email for a link to confirm it (take care it could be in the junk email folder)

The page can also be accessed from:

http://gorgiasb.tuc.gr/GorgiasCloud.html

Where you can find more info and check out the tutorials.

An Example of Argumentation Decision Policy

- **Decision policy of a seller agent**

- Normally, sell a product at its high price. You can sell a product at the lower price only if payment is cash (but normally prefer to sell high). Regular customers can be offered the low price*. In high season you must sell at high prices.

  * This could be conditional e.g. to buy 2 items, etc.

- **Options:** sell_high or sell_low

Seller agent: Scenarios

<1, {}, sell(Prd,Ag, high)>

<2, {pay_cash(Prd,Ag)}, sell(Prd,Ag, high)>
- But unlocks the possibility to sell low

<3, {pay_cash(Prd,Ag), regular(Ag)}, sell(Prd,Ag, high); sell(Prd,Ag, low)>
- Non-deterministic Scenario

<4, {pay_cash(Prd,Ag), regular(Ag), high_season}, sell(Prd,Ag, high)>

Decision policy: **seller agent**

- Object-level argument rules:

  r1: sell(Prd, Ag, high) $\leftarrow$ true
  r2: sell(Prd, Ag, low) $\leftarrow$ pay-cash(Ag, Prd)

- Default Priority: r1 > r2
- We also need to express prices are contrary
  - r3: $\neg$sell(Prd, Ag, P2) $\leftarrow$ sell(Prd, Ag, P1), P2$\neq$P1
  - Complementary relation:
    - complement(sell(Prd, Ag, high) , sell(Prd, Ag, low)).

Decision policy: **seller agent**

- Object-level argument **rules:**

  r1(Prd,Ag): sell(Prd, Ag, high) ← true
  r2(Prd,Ag): sell(Prd, Ag, low) ← pay-cash(Ag, Prd)

- Priority rules:
  - *Generally, sell at high prices:*

    R1(Prd,Ag): h-p(r1(Prd,Ag), r2(Prd,Ag)) ← true
  - Regular customers can have low price:

    R2(Prd,Ag): h-p(r2(Prd,Ag), r1(Prd,Ag)) ← regular(Ag)
  - But not at high season:

    C1(Prd,Ag): h-p(R1(Prd, Ag), R2(Prd, Ag)) ← high-season

Seller agent: Structure of Policy

- Default Policy: “Sell high”
  - For normal markets and normal customers
- Exceptional Policy: “Sell low”
  - For special markets and customers, e.g. regular customers
- Generally, Exceptional (or Special) policies dominate over the Default (or Normal) ones.
  - For normal exceptional cases, i.e. normal market
  - This is a Meta-Default policy!
- Exceptional Policy over the special policy:
  - Exceptional context of high season market.

Seller agent: Argumentation in Scenarios

- <1, {}, sell(Prd,Ag, high)>
  - Only A={r1(p, ag)} applicable argument: supports option high.
  - Hence A is only **admissible** argument.
  - Hence **sceptical decision**: to sell high.

- <2, {pay_cash(Prd,Ag)}, sell(Prd,Ag, high)>
  - A={r1(p, ag)} supports option high price .
  - B={r2(p, ag)} supports contrary option of low price.
    - A **attacks** B and vice-versa
  - A'={r1(p, ag), R1(p,ag)} strengthens A
    - A' **attacks** B but B does **not attack** A'
  - Also B cannot be strengthened (by any applicable priority rule)
  - Hence B cannot be made admissible
  - Hence **sceptical decision**: to sell high.

Seller agent: Argumentation in Scenarios

- <3, {pay_cash(Prd,Ag), regular(Ag)}, sell(Prd,Ag, high); sell(Prd,Ag, low)> - Non-deterministic Scenario
  - B'={r2(p, ag), R2(p,ag)} strengthens B
    - A' **attacks** B' **and** B' **attacks** A'
  - Both A' and B' are admissible.
  - Hence both high and low are credulous conclusions/decisions.

- A' and B' are in conflict not only on the price but also on the priority of r1(…) over r2(…):
  - They conflict on L= h_p(r1(Prd,Ag), r2(Prd,Ag))
  - They argue about the priority or strength of rules.

Seller agent: Argumentation in Scenarios

- <4, {pay_cash(Prd,Ag), regular(Ag), high_season}, sell(Prd,Ag, high)>
  - A''={r1(p, ag), R1(p,ag), C1(p,ag)} strengthens A' (and A)
  - A'' attacks B' but not vice versa (on h_p(r1(Prd,Ag), r2(Prd,Ag)))
  - A'' admissible – No admissible argument for low.
  - Hence sceptical decision of high price

Thank you, questions?

Nikos Spanoudakis, PhD
Researcher - Teaching Staff
Applied Mathematics and Computers Laboratory,
School of Production Engineering and Management,
Technical University of Crete,
Email: nikos@amcl.tuc.gr
https://users.isc.tuc.gr/~nispanoudakis

References

Čyras, K., Rago, A., Albini, E., Baroni, P., & Toni, F. (2021). **Argumentative XAI: a survey**. *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21)*.

Dimopoulos Y., Kakas A. C. (1995), **Logic Programming without Negation as Failure**. *Proceedings of the 1995 International Symposium on Logic Programming*, pp. 369-383

Dung, P. M. (1995). **On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games**. Artificial intelligence, 77(2), 321-357.

Kakas A. C., Mancarella P., Dung P.M. (1994), **The acceptability semantics for logic programs**. *In Proceedings of the eleventh international conference on Logic programming, Pascal Van Hentenryck (Ed.)*. MIT Press, Cambridge, MA, USA, 504-519.

Kakas A. and Moraitis P. (2003), **Argumentation based decision making for autonomous agents**. *Proceedings of the second international joint conference on Autonomous agents and multiagent systems (AAMAS '03)*. ACM, New York, NY, USA, 883-890

Kakas A.C., Moraitis P., Spanoudakis N.. **Gorgias: Applying Argumentation**. *Argument & Computation*, IOS Press, Vol. 10, No. 1, 2019, pp. 55-81

McCarthy, J. (1980). **Circumscription—a form of non-monotonic reasoning**. *Artificial intelligence*, 13(1-2), 27-39.

Mercier, H., & Sperber, D. (2011). **Why do humans reason? Arguments for an argumentative theory**. *Behavioral and brain sciences*, 34(2), 57-74.

Reed, C., Budzynska, K., Duthie, R., Janier, M., Konat, B., Lawrence, J., ... & Snaith, M. (2017). **The argument web: An online ecosystem of tools, systems and services for argumentation**. *Philosophy & Technology*, 30, 137-160.

Snaith, M., & Reed, C. (2012). **TOAST: online ASPIC+ implementation**. In *Computational Models of Argument* (pp. 509-510). IOS Press.

Spanoudakis N., Constantinou E., Koumi A., Kakas A.C. (2017), **Modeling Data Access Legislation with Gorgias**. In the *30th Int. Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE 2017), Special Track on Applications of Argumentation (APPARG2017)*, Arras, France, June 27-30

Spanoudakis N., Gligoris G., Koumi, A., Kakas A.C.. (2023) **Explainable Argumentation as a Service**. *Journal of Web Semantics*, Elsevier, Vol. 76

Spanoudakis N., Kakas A.C., Moraitis P. (2016), **Applications of Argumentation: The SoDA Methodology**. *In 22nd European Conference on Artificial Intelligence (ECAI 2016)*, The Hague, Holland, 29 Aug-2 Sep

Spanoudakis N., Kakas A.C., Koumi A.. 2022. **Application Level Explanations for Argumentation-based Decision Making**. *1st International Workshop on Argumentation for eXplainable AI (ArgXAI 2022)*, Cardiff, Wales, United Kingdom, September 12th, 2022

Thaler, R. H. (2016). **Behavioral economics: Past, present, and future**. *American economic review*, 106(7), 1577-1600.

Thimm, M. (2017). **The Tweety library collection for logical aspects of artificial intelligence and knowledge representation**. KI-Künstliche Intelligenz, 31(1), 93-97.

Vassiliades, A., Bassiliades, N., & Patkos, T. (2021). **Argumentation and explainable artificial intelligence: a survey.** *The Knowledge Engineering Review*, 36, e5.