Explainable AI via Argumentation: Theory & Practice

Antonis Kakas antonis@ucy.ac.cy
University of Cyprus, Cyprus
Nikos Spanoudakis nispanoudakis@tuc.gr
Technical University of Crete, Greece

Co-founders (with Pavlos Moraitis) of Argument Theory
https://www.argument-theory.com/

ESSAI 2024 School: 22-26 July, Athens

Lecture 2
Authoring Arg-based Knowledge/Systems

- SoDA Methodology (in practice)
  - Options, Scenarios and Scenario-based Preferences
  - Group Preferences and Contraindications
- Examples of Call Assistant & Travel Assistant
  - From Nat. Language to Hierarchies of Scenario-based Preferences
- Gorgias Code Generation from SBPs (Examples)
  - Gorgias Cloud
  - Queries: Answers & Explanations in Gorgias Cloud
- Preparation for rAIson (create account).
- Further Discussion of Student Projects
  - Exercise to expand Call Assistant SBPs and Code

Software Development for Argumentation (SoDA)

facilitates the principled modeling of real-life problems

SoDA Methodology – summary

- 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?

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

- A very important domain based on legislation.
- E.g. EU law in Cyprus:
  - Law 138(I)/2001: Personal Data Protection
  - Law N. 1(I)/2005: Patient Rights
- Possible options for a decision:
  - Full Access
  - Partial Access
  - Read Only Access
  - Restricted Read Access
  - Suspended Access
  - No Access
- A real-world problem addressed using Gorgias and the medica app:
  - http://medica.cs.ucy.ac.cy/home_page.php

Medical records access form

Patient Id: *

123

File Id: *

10

Select the person requesting Access to Medical Records: *

Docto

Is there a written consent from the Patient? *

Yes

Is there an order from the Medical Association *

Yes

Access Reason: *

- Publish
- Treatment purpose
- Data Processing Purposes
- Personal Use
- Educational Purposes
- Research Purposes

SUBMIT

Your level of access is: Limited Plus Access

Legislation	Explanation
The Safeguarding and the Patients' Rights Protection Act 2004, Article 15, paragraph 2b	**The doctor has limited plus access to the owner's data for therapy/medical use.**

Do you need higher level of access for the same purpose?

Do you need higher level of access for other purpose?

Medical Data Access – with SoDA

- Requirements of a decision problem
  in high level form, (controlled) natural language:
  - *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.*

Medical Data Access–Scenario-based Preferences

- 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.*

Phase 1:
Mark Decision Factors and Preference Keys to aid the modeling process

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario

SoDA step 1:
What are the options? Identify and place on columns.

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access

SoDA step 1:
What are the options? Identify and place on columns

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access

SoDA step 2:
What are the object level arguments? Define initial scenarios in level 1. Mark the enabled options

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access
1a	true		X
1b	Owner	X	

SoDA step 2:
What are the object level arguments? Define initial scenarios in level 1. Mark the enabled options

Object level arguments level is 1 indicating that they are at the bottom of the hierarchy of scenarios. To differentiate between them we can add a letter after the level.

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access
1a	true		X
1b	Owner	X	

SoDA step 3:
What are the possible scenarios given the object-level arguments?

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access
1a	true		X
1b	Owner	X	
2	Owner, true	X	

SoDA step 3: What are the possible scenarios given the object-level arguments?

The combination of scenarios creates a next level in the hierarchy of scenarios (starting from the initial scenario).

Primary choice in the combination of the two object level arguments

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Refinement

Level	Scenario	Full access	No access
1a	true		X
1b	Owner	X	
2	Owner, true	X	

No need to repeat true

SoDA step 4:
What are the contexts that refine the scenarios?

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access
1a	true		X
1b	Owner	X	
2	Owner	X	
3	Owner, Taking critical tests		X

SoDA step 4:
What are the contexts that refine the scenarios?

Successive scenario refinements and combinations indicate a next level in the hierarchy of scenarios (starting from the initial scenario).

Medical Data Access–Scenario-based Preferences

- 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.*
- SBPs:

Level	Scenario	Full access	No access
1a	true		X
1b	Owner	X	
2	Owner	X	
3	Owner, Taking critical tests		X

SoDA step 5:
Is the model complete?

YES!

Guidelines: When defining SBPs

- Start with object level (initial) scenarios
  - They enable (unlock) the options
- Continue with Combinations
  - Two scenarios are combined to form a composite scenario
  - The composite scenario supports the union of the options supported by the combined scenarios
  - The combined scenario elements is the union of the elements of the combined scenarios

Guidelines: When defining SBPs (cont.)

- Continue with a refinement
  - At least one element is added to an existing scenario to define a refined (next level) scenario
  - The refined scenario supports a genuine subset of options supported by the previous level scenario
  - If an existing scenario supports more than one refinements (branches), then the modeler may choose to continue in another table of SBPs

Example Problems (1)

- **Contextual Decision Policy/Making**
  - **Example D1: Call Assistant (Personal Policy)**

"Normally, allow a call. When at work deny a call from an unknown number. When busy at work also deny a call from a known number unless it is an emergency family call. Always allow a call from my manager. "

Options: allow a call, deny a call.

Example Problems (1)

- **Contextual Decision Policy/Making**
  - **Example D1: Call Assistant (Personal Policy)**

"Normally, allow a call. When at work deny a call from an unknown number. When busy at work also deny a call from a known number unless it is an emergency family call. Always allow a call from my manager. "

Options: allow a call, deny a call.

Mark Decision Factors and Preference Keys to aid the modeling process

Call Assistant–Scenario-based Preferences (1)

- Normally, allow a call. When at work deny a call from an unknown number. When busy at work also deny a call from a known number unless it is an emergency family call. Always allow a call from my manager.
- SBPs

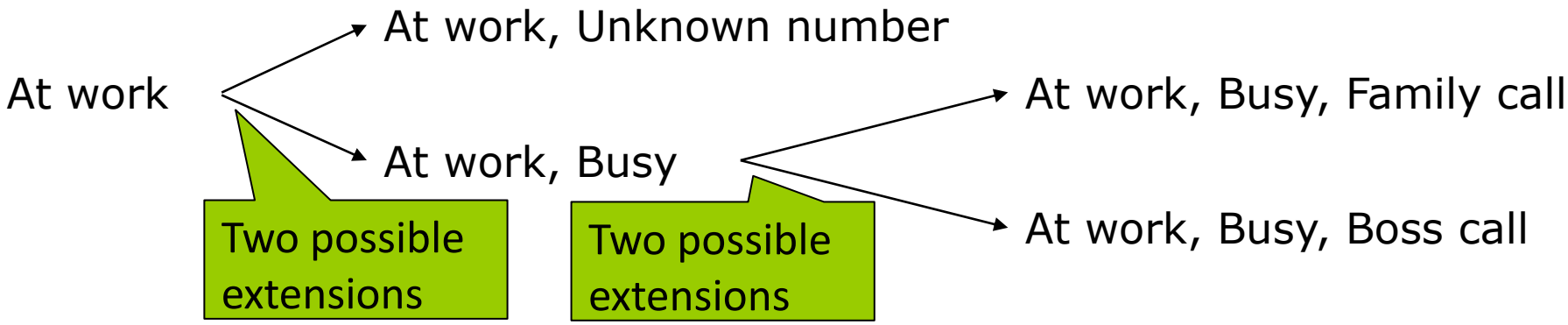
Level	Scenario	Deny call	Allow call
1	At work	X	X
2a	At work, Unknown number	X	

Level	Scenario	Deny call	Allow call
1	At work	X	X
2b	At work, Busy	X	
3ba	At work, Busy, Family call		X

Level	Scenario	Deny call	Allow call
1	At work	X	X
2b	At work, Busy	X	
3bb	At work, Busy, Boss call		X

Call Assistant–Scenario-based Preferences (1)

- Context evolution



- SBPs

Level	Scenario	Deny call	Allow call
1	At work	X	X
2a	At work, Unknown number	X	

Level	Scenario	Deny call	Allow call
1	At work	X	X
2b	At work, Busy	X	
3ba	At work, Busy, Family call		X

Level	Scenario	Deny call	Allow call
1	At work	X	X
2b	At work, Busy	X	
3bb	At work, Busy, Boss call		X

Call Assistant–Scenario-based Preferences (2)

- Normally, allow a call. When at work deny a call from an unknown number. When busy at work also deny a call from a known number unless it is an emergency family call. Always allow a call from my manager.
- SBPs

Level	Scenario	Deny call	Allow call
1	At work	X	X
2a	At work, Unknown number	X	

Level	Scenario	Deny call	Allow call
1	At work	X	X
2b	At work, Busy	X	
3b	At work, Busy, Family call		X

Call from my manager is considered as an object level argument

Level	Scenario	Deny call	Allow call
1	Boss call	X	X
2c	Boss call, At work	X	

Example Problems (2)

- **Contextual Decision Policy/Making**
  - **Example D2: Travel Assistant (Personal Policy)**

"For long distance travel it is possible to use all means of transport. If the bus stop is near, I prefer to get the bus. If it is a cold day, I can take the metro or a taxi. If the bus stop is near and it is a cold day, I prefer to take the metro, except if it rains, in which case I will take a taxi. I do not take the taxi when I am short on funds."

Options: take a taxi, take the bus, take the metro.

Travel assistant–Scenario-based Preferences (1)

- For long distance travel it is possible to use all means of transport. If the bus stop is near, I prefer to get the bus. If it is a cold day, I can take the metro or a taxi. If the bus stop is near and it is a cold day, I prefer to take the metro, except if it rains, in which case I will take a taxi. I do not take the taxi when I am short on funds.
- SBPs

Level	Scenario	metro	taxi	bus
1	Visit friend	X	X	X
2a	Visit friend, bus stop nearby			X

Level	Scenario	metro	taxi	bus
1	Visit friend	X	X	X
2b	Visit friend, cold	X	X	
3b	Visit friend, cold, bus stop nearby	X		
4b	Visit friend, cold, bus stop nearby, rains		X	
5b	Visit friend, cold, bus stop nearby, rains, short on funds	X		

Travel assistant–Scenario-based Preferences (2)

- For long distance travel it is possible to use all means of transport. If the bus stop is near, I prefer to get the bus. If it is a cold day, I can take the metro or a taxi. If the bus stop is near and it is a cold day, I prefer to take the metro, except if it rains, in which case I will take a taxi. I do not take the taxi when I am short on funds.
- SBPs

Level	Scenario	not taxi	metro	taxi	bus
1	Visit friend		X	X	X
2a	Visit friend, bus stop nearby				X

Level	Scenario	not taxi	metro	taxi	bus
1	Visit friend		X	X	X
2b	Visit friend, cold		X	X	
3b	Visit friend, cold, bus stop nearby		X		
4b	Visit friend, cold, bus stop nearby, rains			X	

Level	Scenario	not taxi	metro	taxi	bus
1	Short on funds	X			
2c	Short on funds, Visit friend	X			

Translating/Mapping hierarchies of SBPs to Gorgias Argumentation Theories

Gorgias is an open source general argumentation framework that combines the ideas of preference reasoning and abduction

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

Decision Making in Argumentation (see Lecture 1)

- 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.
  - *Priority Arguments* that give relative strength to the arguments for the various options

Writing arguments in Gorgias source code

- The language for representing the arguments is given by sentences 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 (optional) selected variables from the Head and Body of the argument.

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 an argument. Using the “rule” syntax we can write:

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

- which means that the argument with signature *Sig1* has higher priority than the argument with signature *Sig2*, when 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).

Medical Data Access – Translation to Gorgias

- 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(r1(Agn), access(Agn, no_access), []).
rule(r2(Agn), access(Agn, full_access), []) :- owner(Agn).
rule(p21(Agn), prefer(r2(Agn), r1(Agn)), []).
rule(p12(Agn), prefer(r1(Agn), r2(Agn)), []):- critical_tests(Agn).
rule(c12_21(Agn), prefer(p12(Agn), p21(Agn)), []).
complement(access(Agn, no_access), access(Agn, full_access)).
complement(access(Agn, full_access), access(Agn, no_access)).
```

Medical Data Access – Translation to Gorgias

- 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(r1(Agn), access(Agn, no_access), [])
rule(r2(Agn), access(Agn, full_a                wner(Agn).
rule(p21(Agn), prefer(r2(                gn)), []).
rule(p12(Agn),                 r2(Agn)), []):- critical_tests(Agn).
rule(c12                 efer(p12(Agn), p21(Agn)), []).
                 (access(Agn, no_access), access(Agn, full_access)).
    plement(access(Agn, full_access), access(Agn, no_access)).
```

The hard way!

Scenario-based Preferences to Gorgias code

- SBPs:

Level	Scenario	Full access	No access
1	true		X
1	Owner	X	
2	Owner	X	
3	Owner, Taking critical tests		X

- Code in GORGIAS: Take SBP tables one after the other

```
rule(r1(Agn), access(Agn, no_access), []).
rule(r2(Agn), access(Agn, full_access), []):- owner(Agn).
rule(p21(Agn), prefer(r2(Agn), r1(Agn)),[]).
rule(p12(Agn), prefer(r1(Agn), r2(Agn)),[]):- critical_tests(Agn).
rule(c12_21(Agn), prefer(p12(Agn), p21(Agn)),[]).
```

Call Assistant–SBPs to code (1)

- SBPs:

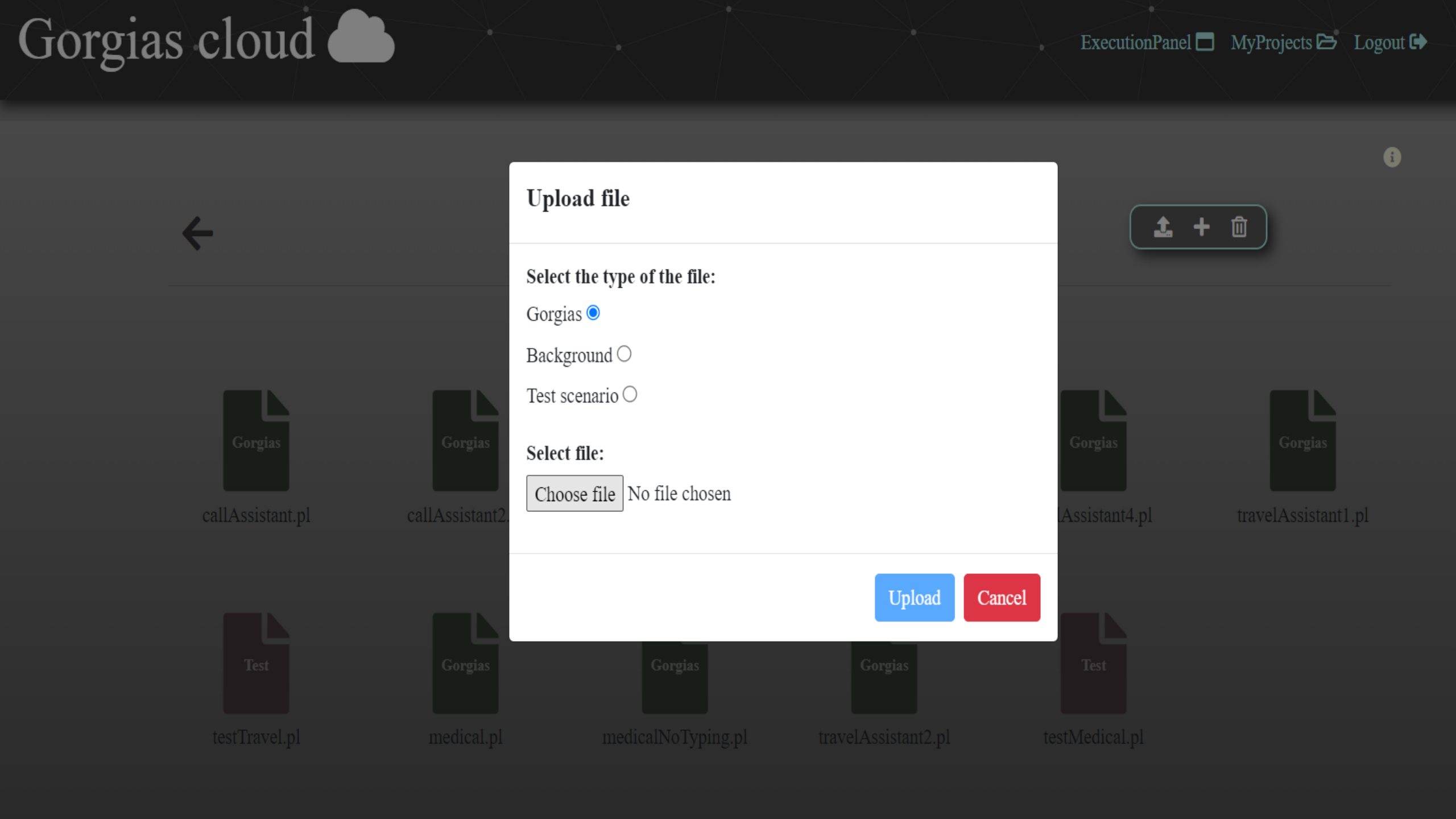
Level	Scenario	Deny call	Allow call
1	At work	X	X
2a	At work, Unknown number	X	
2b	At work, Busy	X	
3ba	At work, Busy, Family call		X
3bb	At work, Busy, Boss call		X

- Code in GORGIAS:

```
rule(r1, allow_call, []):-at_work.
rule(r2, deny_call, []):-at_work.
rule(p21a, prefer(r2, r1), []):- at_work, unknown_number.
rule(p21b, prefer(r2, r1), []):- at_work, busy.
rule(p12ba, prefer(r1, r2), []):- at_work, busy, family_call.
rule(c12ba, prefer(p12ba, p21b), []):- at_work, busy, family_call.
rule(p12bb, prefer(r1, r2), []):- at_work, busy, boss_call.
rule(c12bb, prefer(p12bb, p21b), []):- at_work, busy, boss_call.
```

Hands on : Gorgias Cloud

- Live Demo Gorgias Cloud
  - Upload code
  - Run scenarios
  - Read and discuss the explanations


Gorgias cloud
ExecutionPanel
MyProjects
Logout
Upload file
Select the type of the file:
Gorgias
Background
Test scenario
Select file:
Choose file
No file chosen
Upload
Cancel
Gorgias
callAssistant.pl
Gorgias
callAssistant2.
Gorgias
Assistant4.pl
Gorgias
travelAssistant1.pl
Test
testTravel.pl
Gorgias
medical.pl
Gorgias
medicalNoTyping.pl
Gorgias
travelAssistant2.pl
Test
testMedical.pl

Gorgias cloud

ExecutionPanel

Gorgias/Background Files

essai2024/callAssistant.pl

at_work.
family_call.
busy.

- prove([allow_call], InternalExplanation).
  Solution 1

  Internal Explanation: [c12ba,p12ba,r1],

  ' Application Level Explanation

  The statement "allow_call" is supported by:

  - "at_work"

  This reason is :

  - Stronger than the reason of "at_work" and "busy" supporting "deny_call" when "at_work" and "busy" and "family_call"

  '

Hands on : Gorgias Cloud

- Create accounts – for those not already done so
- Load and Run the call assistant for various scenarios
- Exercise 1: Extend the gorgias code of the Call assistant with the further requirement:
  - "Allow the call when at work I have a family call"
- Exercise 2: Extend it further with your own one sentence requirement

Exercise 1 in SBPs
Call Assistant–Scenario-based Preferences (3)

- Normally, allow a call. When at work deny a call from an unknown number. When busy at work also deny a call from a known number unless it is an emergency family call. Always allow a call from my manager.
- SBPs

Level	Scenario	Deny call	Allow call
1	At work	X	X
2a	At work, Unknown number	X	
2b	At work, Busy	X	
2c	At work, Family call		X
3c	At work, Busy, Family call		X
1	Boss call		X
2d	At work, Boss call		X

I want family call to allow the call whenever at work – not a refinement. Combination of previous level conflicting scenarios

Gorgias cloud

Gorgias/Background Files

essai2024/callAssistant4.pl

```
:-dynamic at_work/0, unknown_number/0,
boss_call/0, busy/0, family_call/0.

rule(r1, allow_call, []):-at_work.
rule(r2, deny_call, []):-at_work.
rule(p21a, prefer(r2, r1), []):- at_work,
unknown_number.
rule(p21b, prefer(r2, r1), []):- at_work, busy.
rule(p12c, prefer(r1, r2), []):- at_work,
family_call.
rule(c12c, prefer(p12c, p21b), []):- at_work,
busy, family_call.
rule(r3, allow_call, []):- boss_call.
rule(p2d, prefer(r3, r2), []):- at_work, boss_call.
```

- prove([deny_call], InternalExplanation).
  false

- prove([allow_call], InternalExplanation).
  Solution 1

  Internal Explanation: [p2d,r1,r3],

  ' Application Level Explanation

  The statement "allow_call" is supported by:

  - "at_work" and "boss_call"

  This reason is :

  - Stronger than the reason of "at_work" supporting "deny_call" when "at_work" and "boss_call"

  '

Gorgias Prompt Maximum number of answers: 1 Clear panel

Gorgias?: allow_call ›Run

Scenario Test Files

essai2024

Add/Expand scenario

Add scenario..

boss_call

at_work

busy

SoDA Methodology – summary

- 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.
     - Revisiting scenarios

Preview : raison

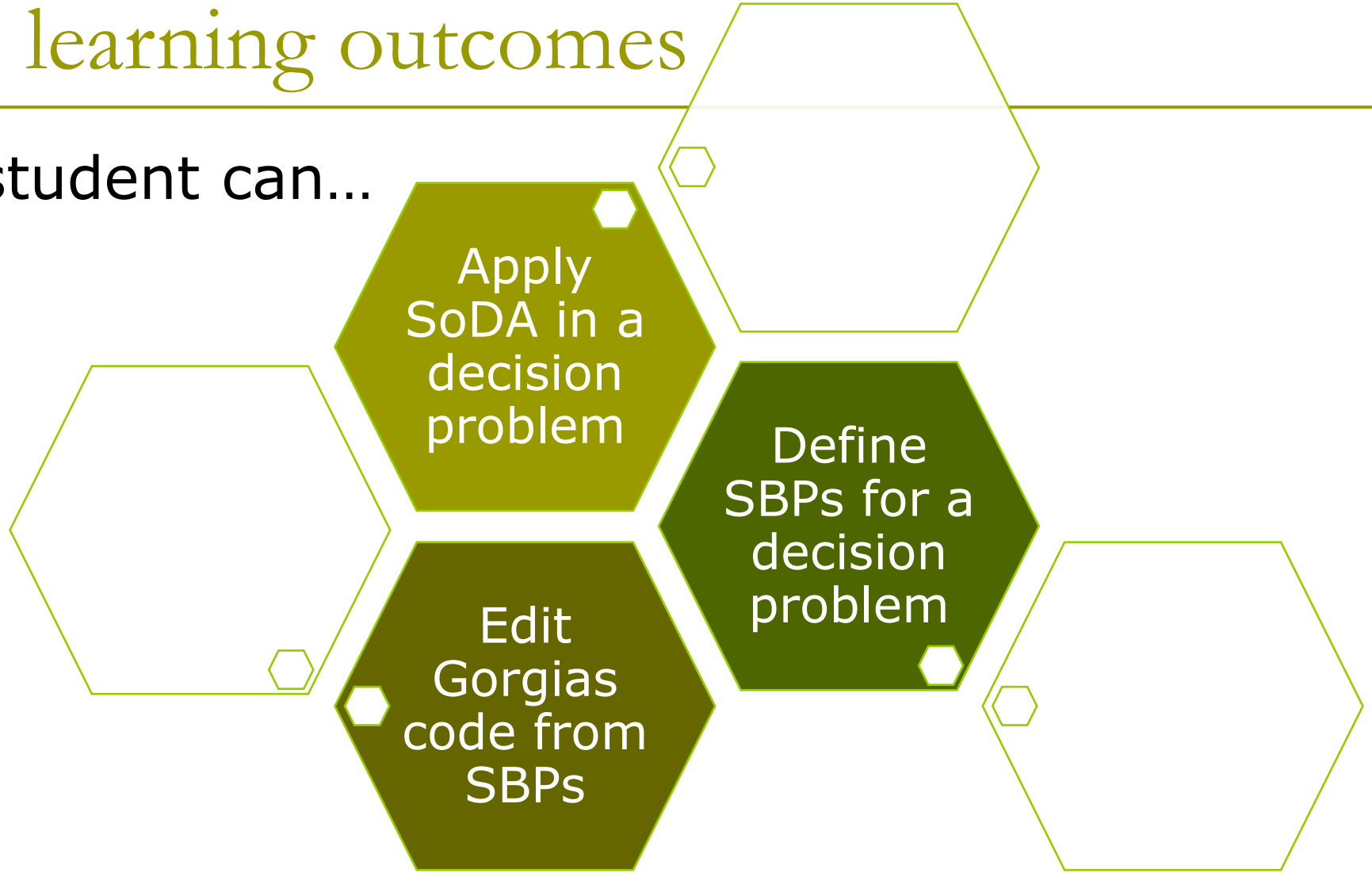
- Translation to Gorgias can be automated
- raison: Authoring SBPs
  - Create accounts
  - Visit https://ai-raison.com/ and register
- Added value
  - Work in high level – no code platform
  - Helps the stakeholder to clarify the requirements
  - Revise the requirements easily

Hands on project

- Finalize your decision policy in Natural Language.
- Extract the hierarchies of SBPs for your project policy
  - Submit both the Natural Language description and the hierarchies of SBPs to our emails with subject "Hands on – day 2"

Day 2 learning outcomes

- The student can…



Reading

- For details
  - Spanoudakis, N. I., Gligoris, G., Koumi, A., & Kakas, A. C. (2023). Explainable argumentation as a service. Journal of Web Semantics, 76, 100772.
  - Kakas, A. C., Moraitis, P., & Spanoudakis, N. I. (2019). GORGIAS: Applying argumentation. Argument & Computation, 10(1), 55-81.
  - Kakas, A., & Moraitis, P. (2003, July). Argumentation based decision making for autonomous agents. In Proceedings of the second international joint conference on Autonomous agents and multiagent systems (pp. 883-890).
  - Dimopoulos, Y., & Kakas, A. (1995). Logic programming without negation as failure. In Proceedings of the 1995 International Symposium of Logic Programming (pp. 369–383).
  - Spanoudakis, N. I., Constantinou, E., Koumi, A., & Kakas, A. C. (2017). Modeling data access legislation with Gorgias. In 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE 2017), Arras, France, June 27-30, Proceedings, Part II 30 (pp. 317-327). Springer
- See the following slides

Call Assistant–SBPs to code (2)

- SBPs:

Level	Scenario	Deny call	Allow call
1	At work	X	X
2a	At work, Unknown number	X	
2b	At work, Busy	X	
3b	At work, Busy, Family call		X
1	Boss call		X
2c	At work, Boss call		X

```
rule(r1, allow_call, []):-at_work.
rule(r2, deny_call, []):-at_work.
rule(p21a, prefer(r2, r1), []):- at_work, unknown_number.
rule(p21b, prefer(r2, r1), []):- at_work, busy.
rule(p12b, prefer(r1, r2), []):- at_work, busy, family_call.
rule(c12b, prefer(p12b, p21b), []):- at_work, busy, family_call.
rule(r3, allow_call, []):- boss_call.
rule(p32c, prefer(r3, r2), []):- at_work, boss_call.
```

Travel Assistant–SBPs to code (1)

- SBPs:

Level	Scenario	metro	taxi	bus
1	Visit friend	X	X	X
2a	Visit friend, bus stop nearby			X
2b	Visit friend, cold	X	X	
3b	Visit friend, bus stop nearby, cold	X		
4b	Visit friend, bus stop nearby, cold, rains		X	
5b	Visit friend, bus stop nearby, cold, rains, short on funds	X		

- Gorgias code

```
rule(r1, taxi, []):- visit_friend.
rule(r2, bus, []):- visit_friend.
rule(r3, metro, []):- visit_friend.
rule(p1, prefer(r2, r3), []):- visit_friend, bus_stop_nearby.
rule(p2, prefer(r2, r1), []):- visit_friend, bus_stop_nearby.
rule(p3, prefer(r3, r2), []):- visit_friend, cold.
rule(p4, prefer(r1, r2), []):- visit_friend, cold.
```

Travel Assistant–SBPs to code (1)

- SBPs:

Level	Scenario	metro	taxi	bus
1	Visit friend	X	X	X
2a	Visit friend, bus stop nearby			X
2b	Visit friend, cold	X	X	
3b	Visit friend, bus stop nearby, cold	X		
4b	Visit friend, bus stop nearby, cold, rains		X	
5b	Visit friend, bus stop nearby, cold, rains, short on funds	X		

- Gorgias code:

```
rule(c1, prefer(p3, p1), []):- visit_friend, bus_stop_nearby, cold.
rule(c2, prefer(p4, p2), []):- visit_friend, bus_stop_nearby, cold.
rule(d1, taxi, []):-visit_friend, bus_stop_nearby, cold.
rule(d2, metro, []):-visit_friend, bus_stop_nearby, cold.
rule(pd1, prefer(d1, r3), []):-visit_friend, bus_stop_nearby, cold.
rule(pd2, prefer(d2, r1), []):-visit_friend, bus_stop_nearby, cold.
rule(pd3, prefer(d2, d1), []):- visit_friend, bus_stop_nearby, cold.
```

Here I want to discriminate in a group. I have no conflict

Give priority of these object level rules over those of the initial scenario

Finally give priority to metro according to 3b

Travel Assistant–SBPs to code (1)

- SBPs:

Level	Scenario	metro	taxi	bus
1	Visit friend	X	X	X
2a	Visit friend, bus stop nearby			X
2b	Visit friend, cold	X	X	
3b	Visit friend, bus stop nearby, cold	X		
4b	Visit friend, bus stop nearby, cold, rains		X	
5b	Visit friend, bus stop nearby, cold, rains, short on funds	X		

- Code

```
rule(pd4, prefer(d1, d2), []):- visit_friend, bus_stop_nearby, cold, rains.
rule(cd1, prefer(pd4, pd3), []):-visit_friend, bus_stop_nearby, cold, rains.
rule(cd2, prefer(pd3, pd4), []):-visit_friend, bus_stop_nearby, cold, rains, short_on_funds.
rule(cd3, prefer(cd2, cd1), []):-visit_friend, bus_stop_nearby, cold, rains, short_on_funds.
```