Argumentation and Smart Contracts

Dr Nikolaos Spanoudakis

http://users.isc.tuc.gr/~nispanoudakis

CONTENTS

- Intro to Argumentation
- Preference-based argumentation – Gorgias
- Intro to smart contracts
- An application

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

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

- 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

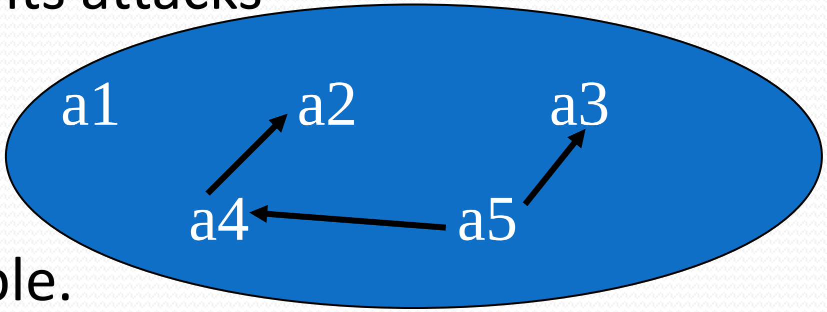
- Premises → Conclusion

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

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.

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

- 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

Given the Common Sense Knowledge:

(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

(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)**

Smart Contracts

An introduction

Smart Contracts

- Recently, the technique of smart contracts has emerged as a way to specify programs that enforce agreements between two or more parties, which can be
  - rules to govern transactions [Delmolino et al., 2016],
  - enforce contractual clauses [Idelberger et al., 2016], and
  - monitor quality of service (QoS) characteristics (e.g. performance, availability, security) [Bunse et al., 2012].

Examples of smart contracts

- supporting cryptocurrency protocols
- executable Service Level Agreements (SLAs)
- wallet applications
- crowdfunding services
- smart cards
- …

Properties for smart contracts

- Economists stress two properties important to good contract design:
  - *observability* by principals and
  - *verifiability* by third parties such as auditors and adjudicators.
- From the traditions behind contract law and the objectives of data security, we derive a third objective,
  - *privity*.
- Be careful, small letters are hidden in the system
- However, most contractual disputes involve an unforeseen or unspecified eventuality [Szabo, 1997]

Gorgias and Smart Contracts

Where we find out how argumentation caters for the execution of smart contracts – including new features

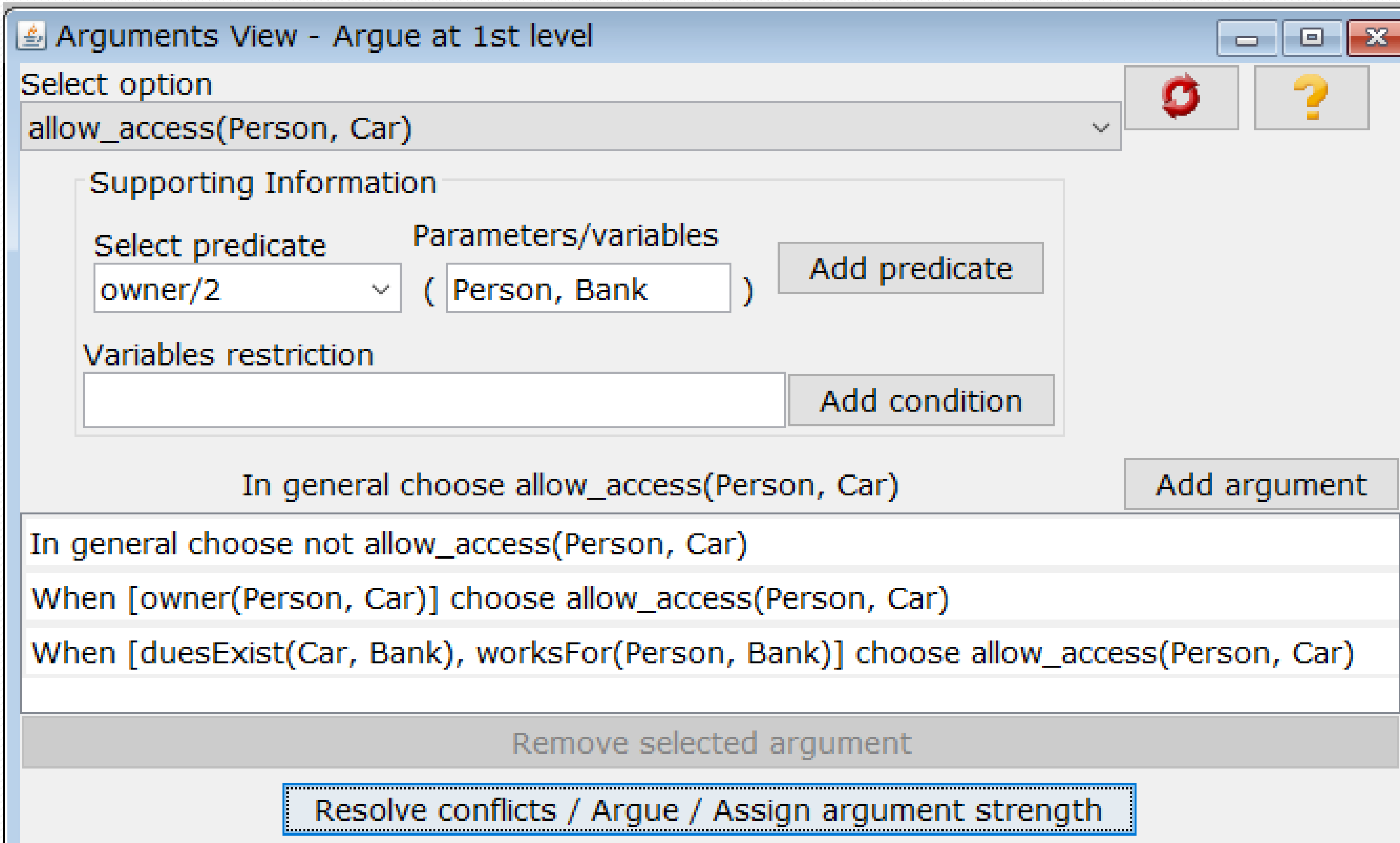
We will use SODA and Gorgias to do a small show case

Requirements for a car lock

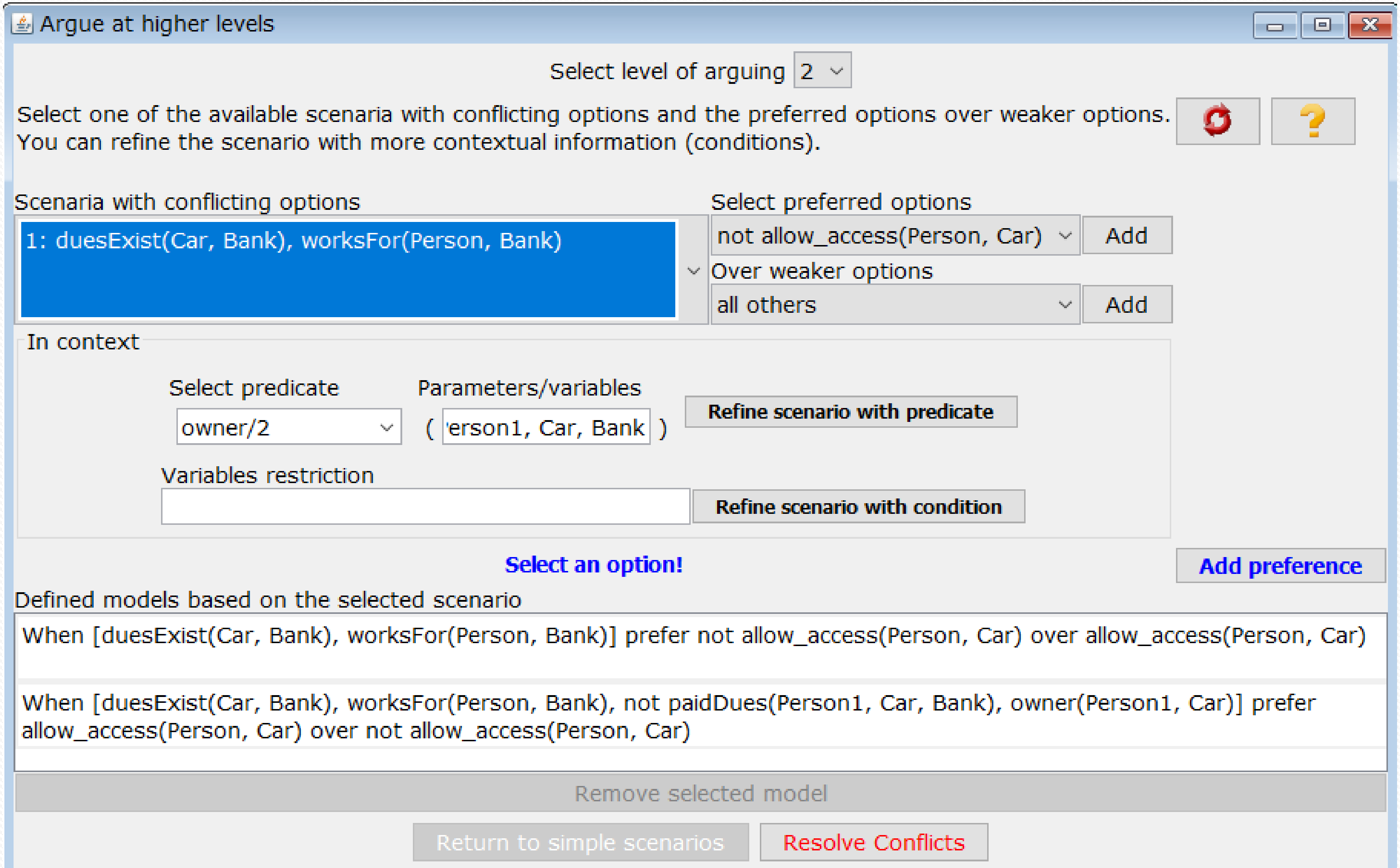
- A lock to selectively let in the owner and exclude third parties;
- A back door to let in the creditor;
- Creditor back door switched on only upon nonpayment for a certain period of time; and
- The final electronic payment permanently switches off the back door.

[Szabo, 1997]

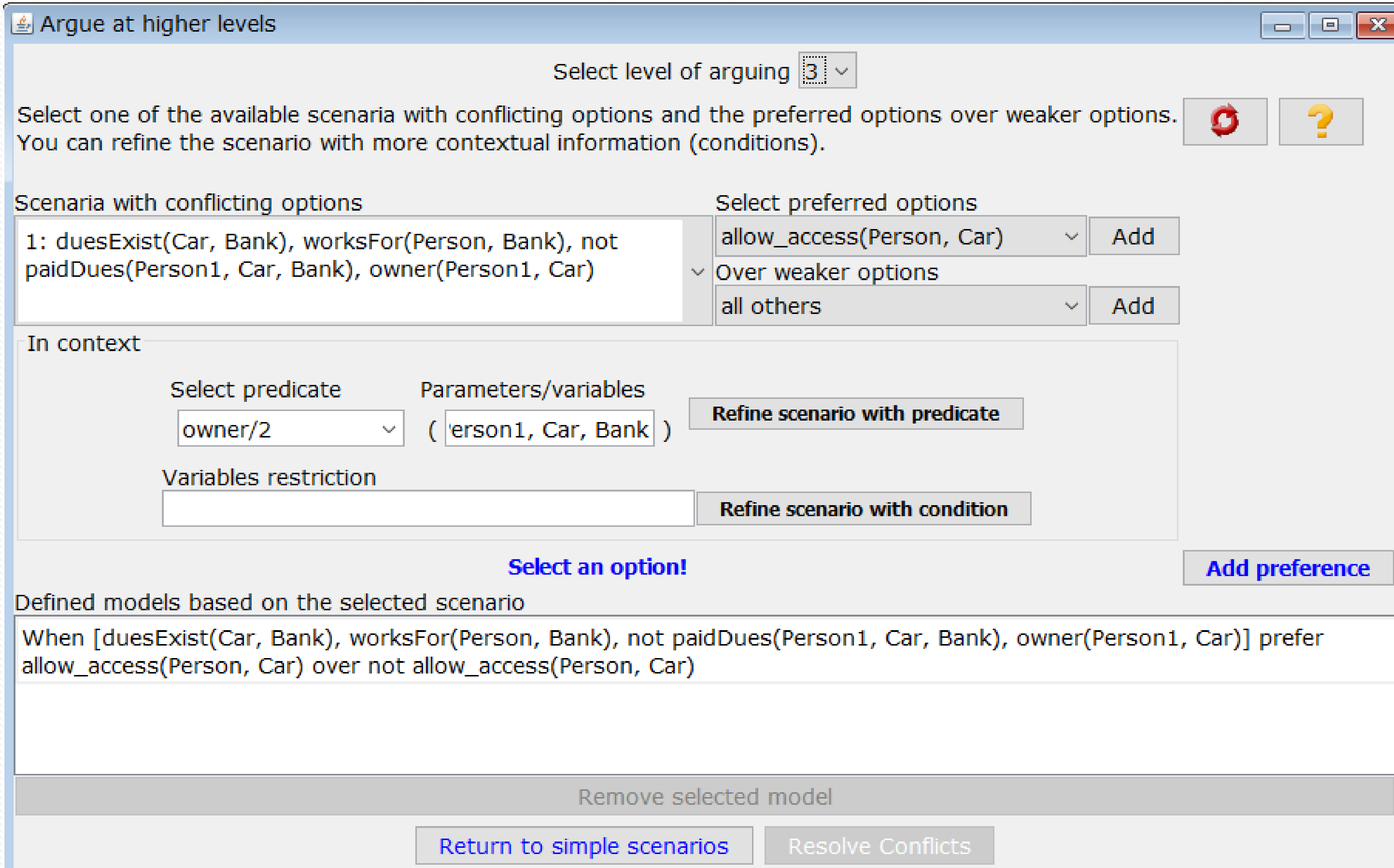
Add object level arguments



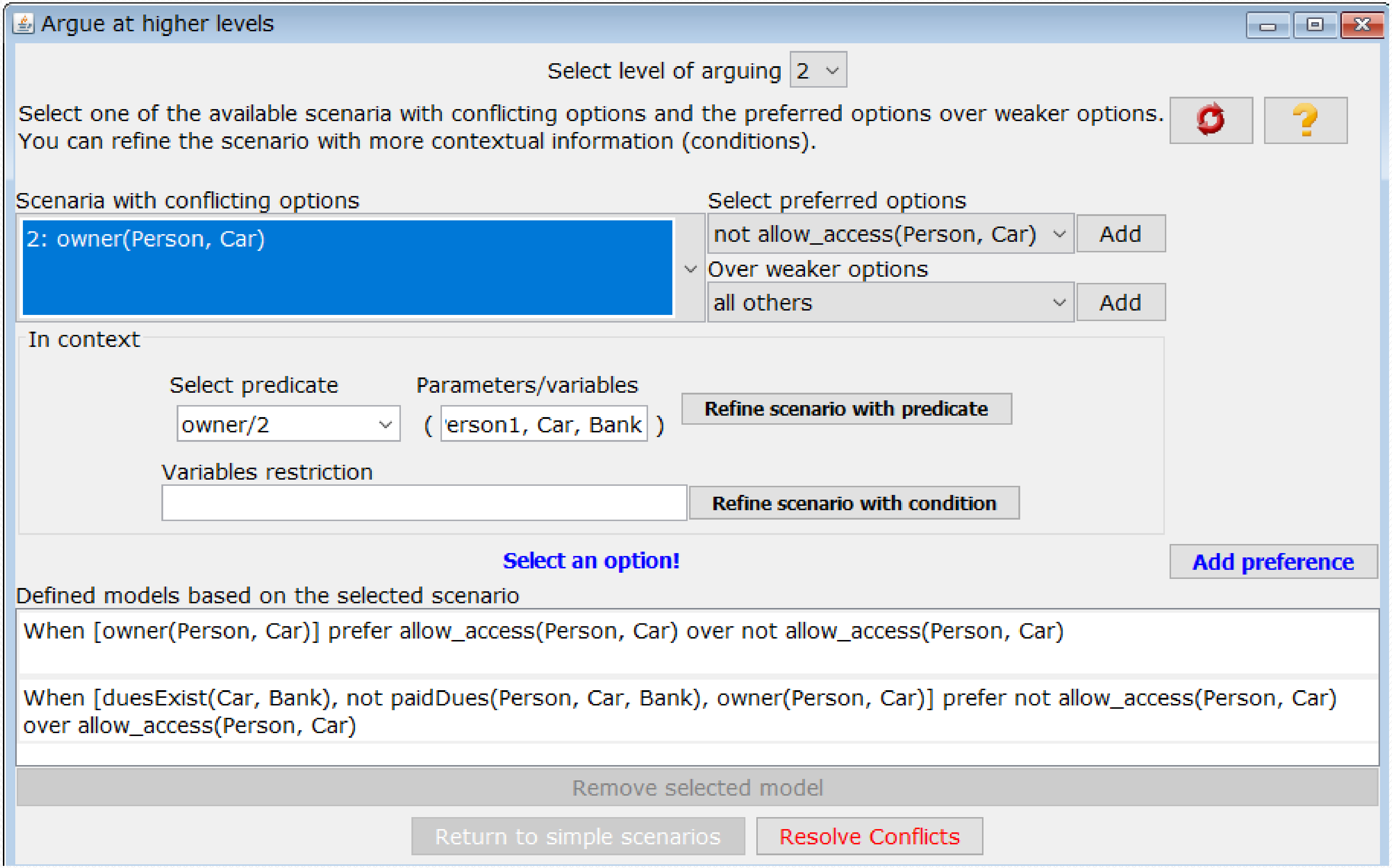
Argue for creditor (1)



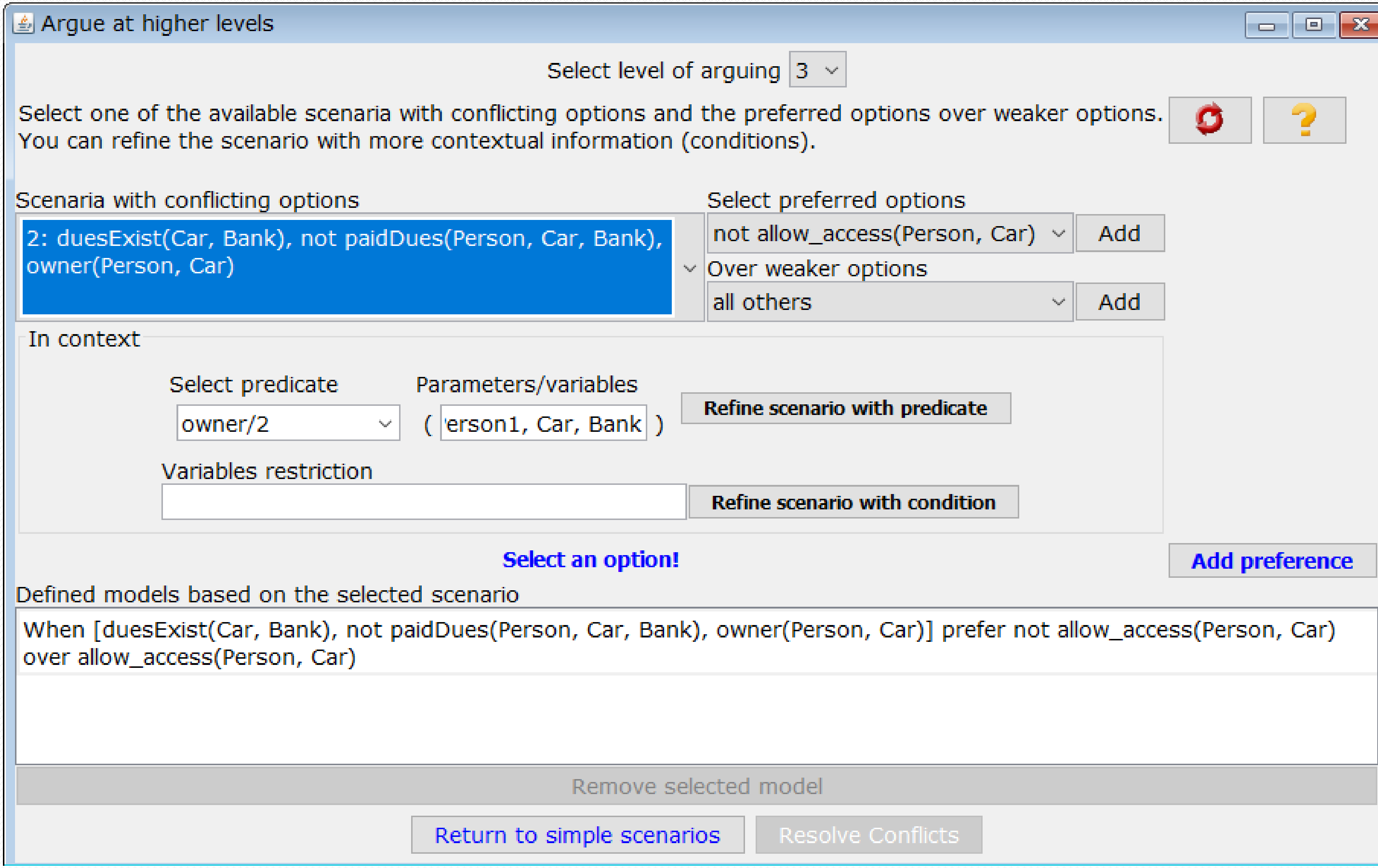
Argue for creditor (2)



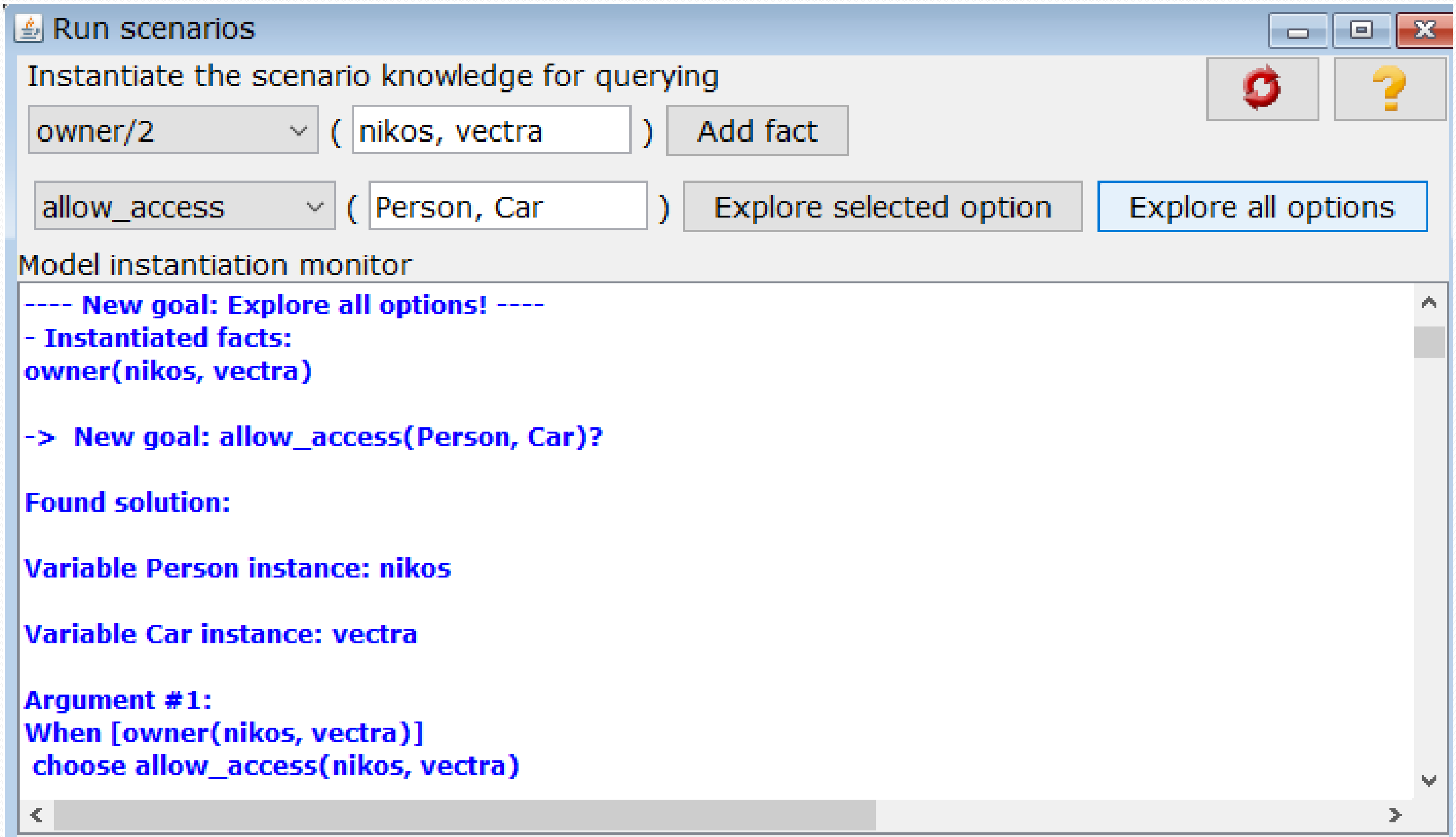
Argue for owner (1)



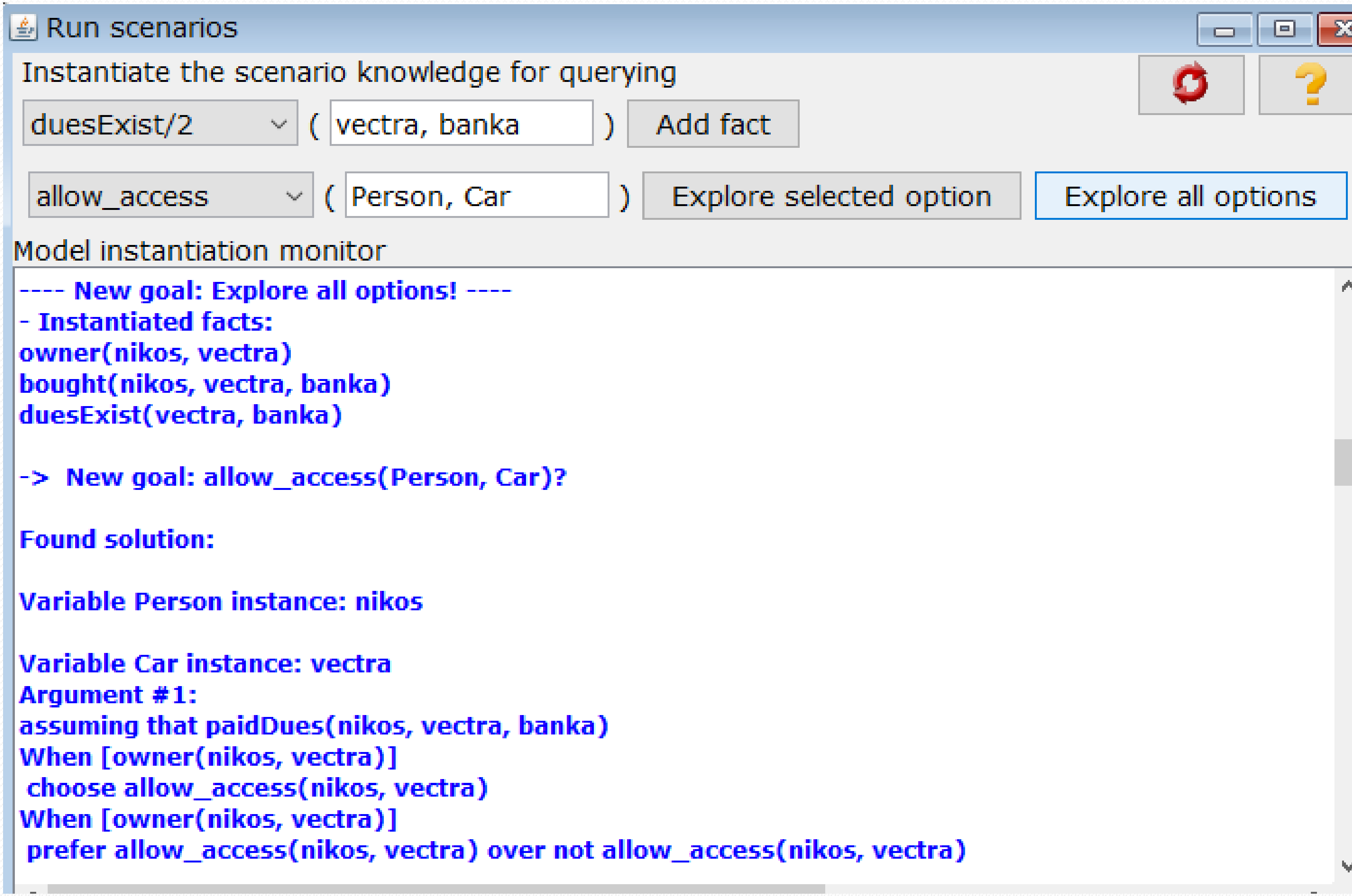
Argue for owner (2)



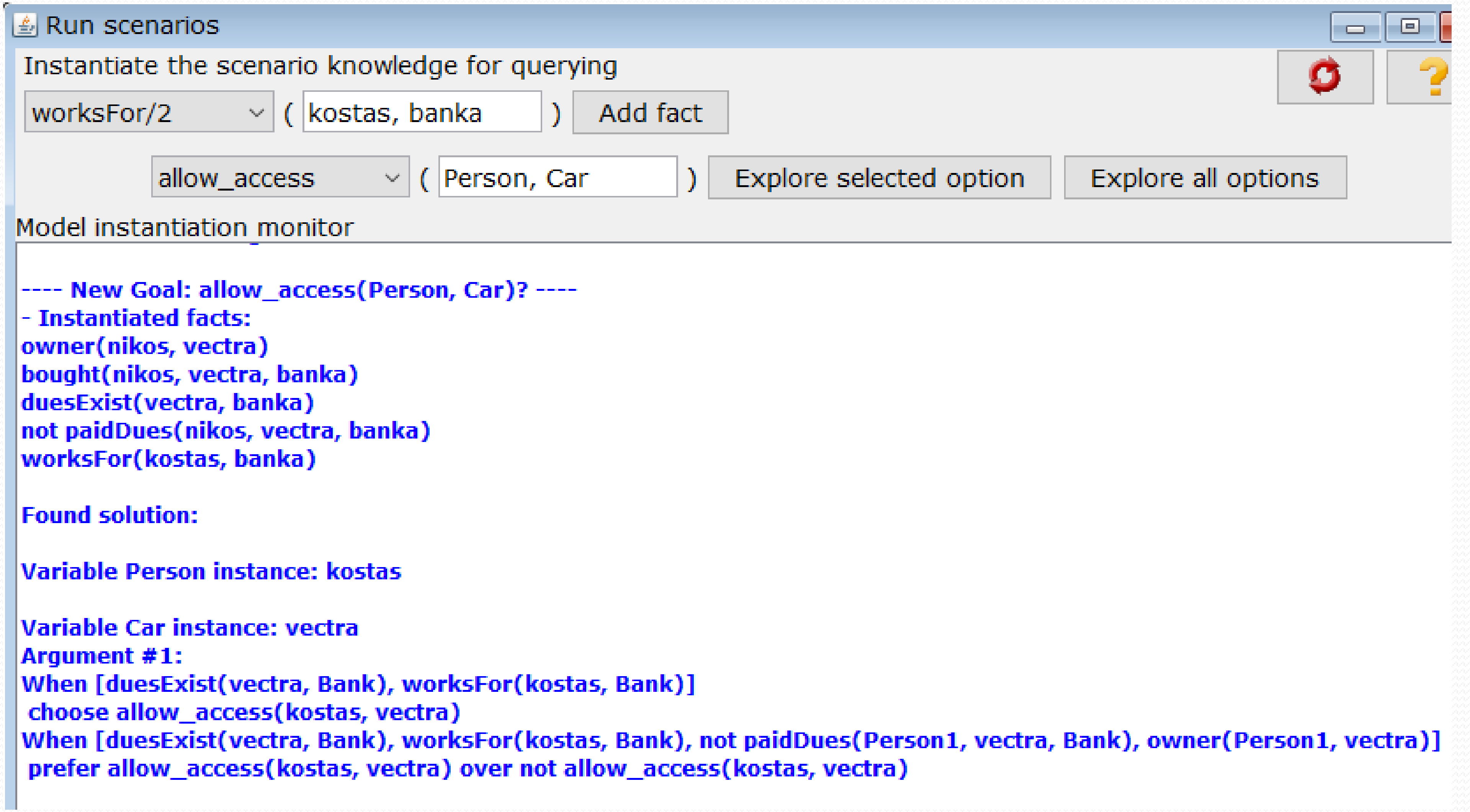
Explanation



Explanation (2)



Explanation (3)



Concluding

- Argumentation seems promising for smart contracts
  - Decisions are verifiable
  - Decisions are explainable (current work with N. Bassiliades)
- We can execute protocols defining the rules and also enforcing them (work under review with A.C. Kakas and P. Moraitis)
- Interesting for the future
  - Smart contracts for the blockchain using Gorgias
  - Smart contracts for business process domain [Mendling et al., 2018]

References

Kakas A. C., **EPL434 course**, *University of Cyprus* **(SPECIAL THANKS for Intro to Argumentation)**

Gorgias-B and Gorgias: http://gorgiasb.tuc.gr , http://www.cs.ucy.ac.cy/~nkd/gorgias/

Bunse, C., Klingert, S., Schulze, T.: **Greenslas: supporting energy-efficiency through contracts**. In: International Workshop on Energy Efficient Data Centers, pp. 54–68. Springer (2012)

Delmolino, K., Arnett, M., Kosba, A., Miller, A., & Shi, E. (2016). **Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab**. In *International Conference on Financial Cryptography and Data Security* (pp. 79-94). Springer, Berlin, Heidelberg.

Idelberger, F., Governatori, G., Riveret, R., & Sartor, G. (2016). **Evaluation of logic-based smart contracts for blockchain systems**. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web* (pp. 167-183). Springer, Cham.

Mendling, J. et al. (2018). **Blockchains for business process management-challenges and opportunities**. *ACM Transactions on Management Information Systems (TMIS), 9(1).*

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.

Szabo, N. (1997), **Formalizing and securing relationships on public networks**. *First Monday 2(9)*