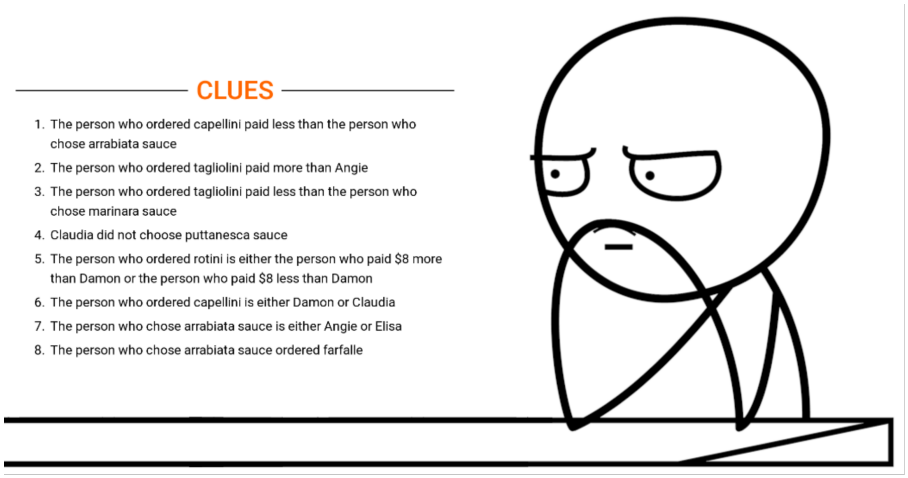
Step-wise Explanations of Constraint Satisfaction Problems

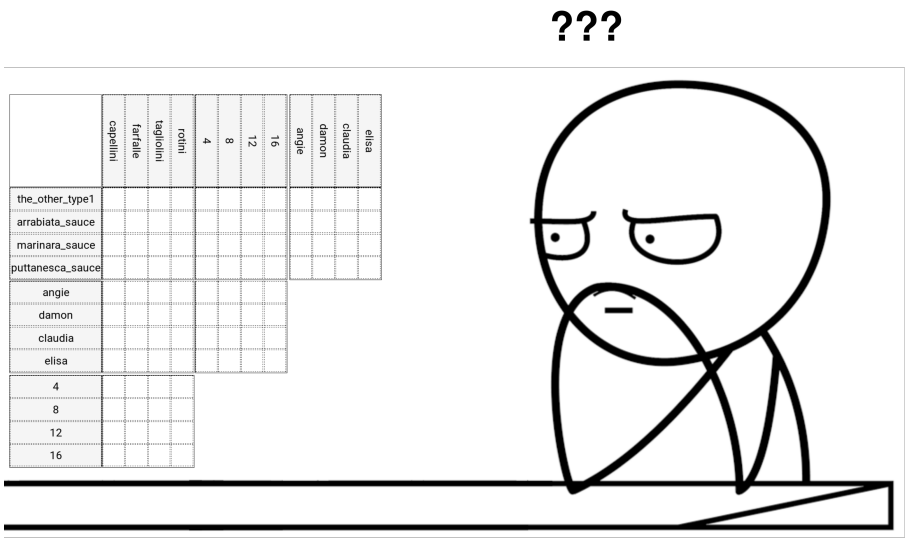
Bart Bogaerts[2] Emilio Gamba[1] Tias Guns[1]

[1]Data Analytics Laboratory, Vrije Universiteit Brussel, Belgium

[2]Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium

Motivation

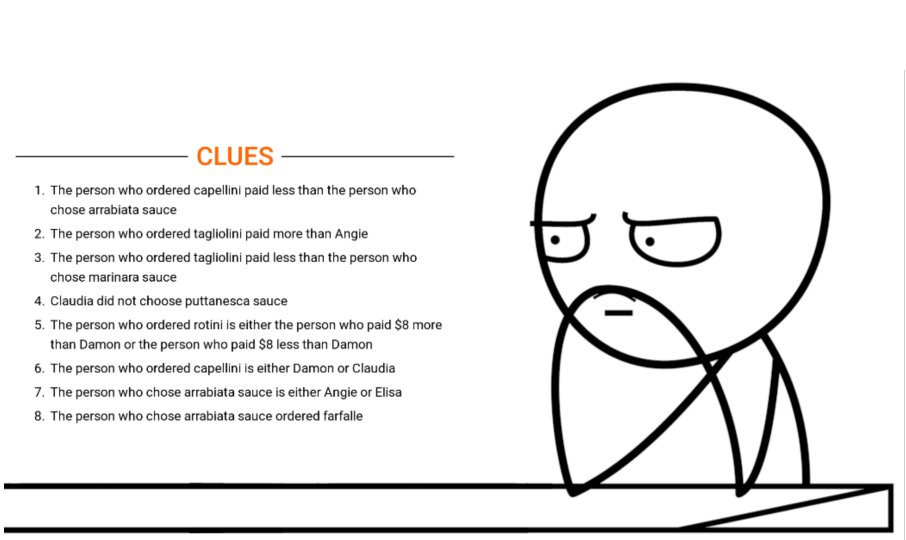


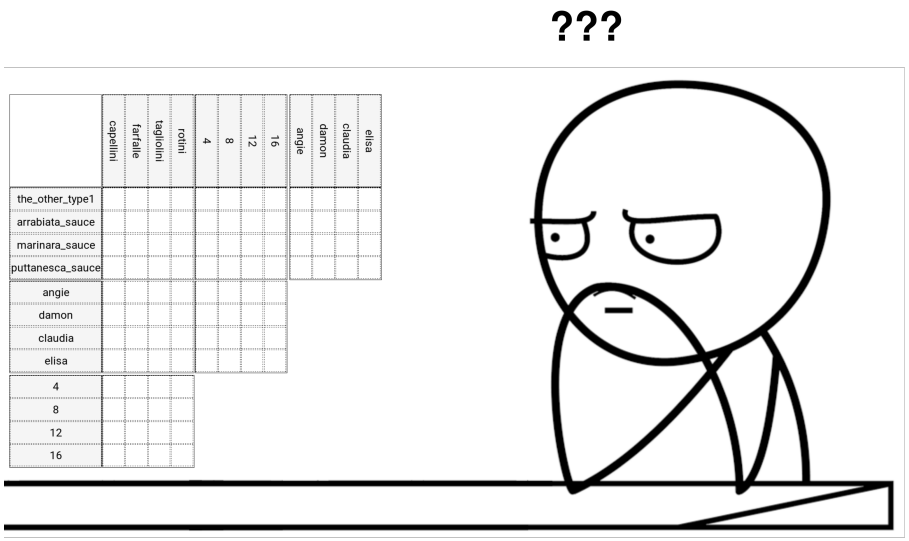


[1] https://freuder.wordpress.com/pthg-19-the-third-workshop-on-progress-towards-the-holy-grail/

[2] Figure src: https://www.jing.fm/iclipt/bRTTbx/

Motivation



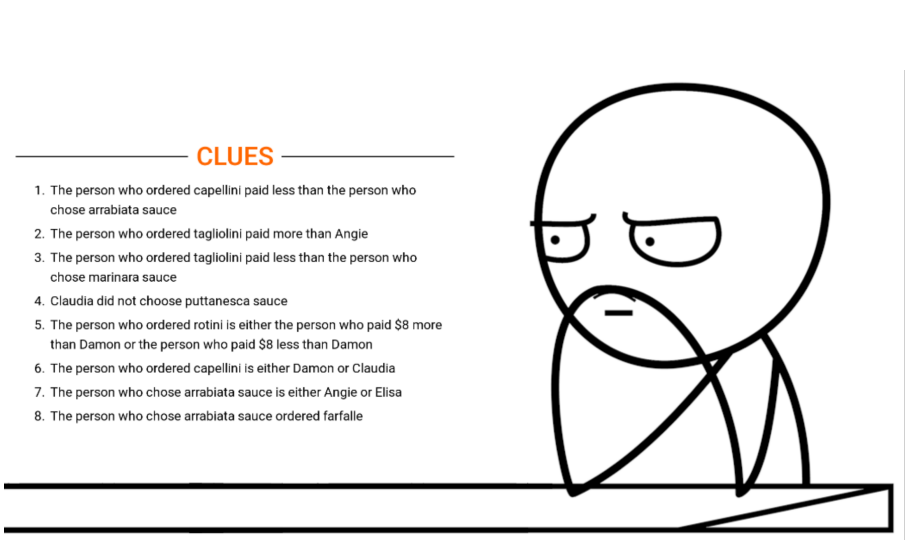


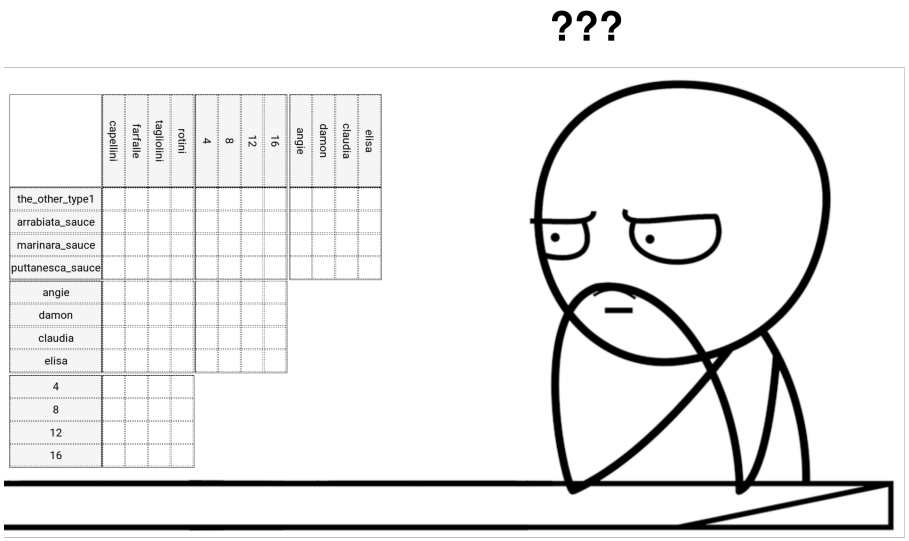
- 2019 Holy Grail Challenge: Zebra puzzles [1] [2]
  - Parse puzzle clues and translate these into CSP

[1] https://freuder.wordpress.com/pthg-19-the-third-workshop-on-progress-towards-the-holy-grail/

[2] Figure src: https://www.jing.fm/iclipt/bRTTbx/

Motivation



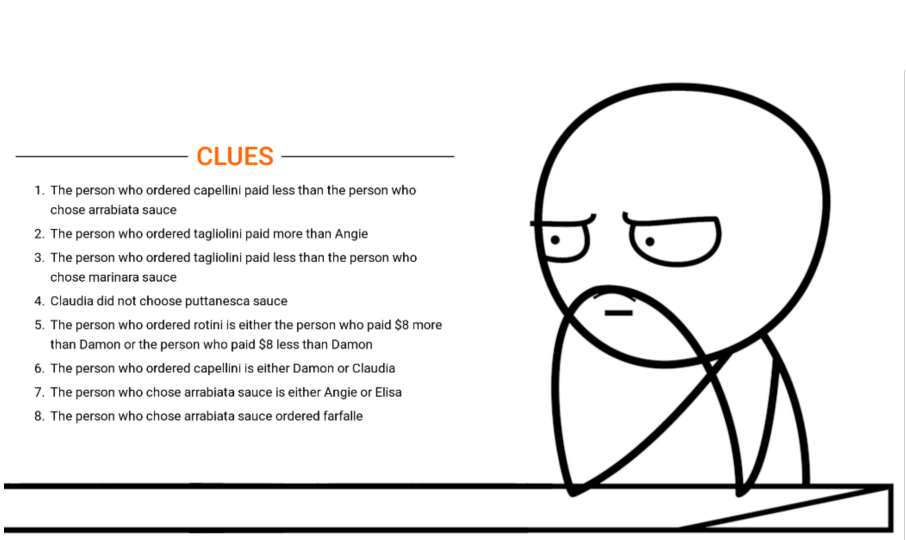


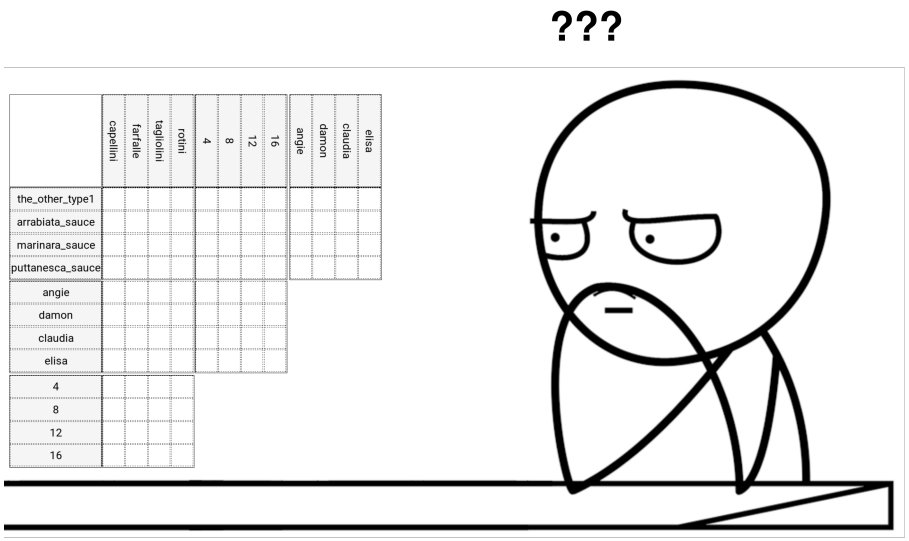
- 2019 Holy Grail Challenge: Zebra puzzles [1] [2]
  - Parse puzzle clues and translate these into CSP
  - Solve CSP automatically

[1] https://freuder.wordpress.com/pthg-19-the-third-workshop-on-progress-towards-the-holy-grail/

[2] Figure src: https://www.jing.fm/iclipt/bRTTbx/

Motivation





- 2019 Holy Grail Challenge: Zebra puzzles [1] [2]
  - Parse puzzle clues and translate these into CSP
  - Solve CSP automatically
  - Explain in a **human-understandable** way how to solve this puzzle

[1] https://freuder.wordpress.com/pthg-19-the-third-workshop-on-progress-towards-the-holy-grail/

[2] Figure src: https://www.jing.fm/iclipt/bRTTbx/

Motivation

Context ?

- *Constraint solving*
  - Problem specification is an explicit *model-based representation* of the problem
  - Opportunity to explain inference steps in terms of this representation

Motivation

Context ?

- *Constraint solving*
  - Problem specification is an explicit *model-based representation* of the problem
  - Opportunity to explain inference steps in terms of this representation

How ?

- *Explain* (strong, complex) propagation in simple steps with a use case on Logic Grid Puzzles (a.k.a Zebra Puzzle, Einstein puzzle)

Motivation

Context ?

- *Constraint solving*
  - Problem specification is an explicit *model-based representation* of the problem
  - Opportunity to explain inference steps in terms of this representation

How ?

- *Explain* (strong, complex) propagation in simple steps with a use case on Logic Grid Puzzles (a.k.a Zebra Puzzle, Einstein puzzle)

End-Goal ?

- *Interactive* constraint solving

Our Contributions

- Formalize the step-wise explanation problem;
- Propose an algorithm (agnostic of actual propagators, consistency level, etc.);
- Propose heuristics for guiding the search for explanations;
- Experimentally demonstrate feasibility.

Outline

- Preliminaries
  - What is a logic grid puzzle?
- Step-wise Explanations
  - Problem definition
    - ★ Explaining an inference step.
    - ★ The explanation-production problem!
    - ★ A non-redundant explanation?
  - Explanation algorithm
- Example explanation
- Conclusion & Future work

Preliminaries

- A logic grid puzzle instance consists of:
  - Set of **Clues**
  - Sets of entities that need to be linked
  - Each entity of one type is linked to *exactly one* entity of each other type (**Bijectivity**)
  - The links are *consistent* (**Transitivity**)

Preliminaries

- A logic grid puzzle instance consists of:
  - Set of **Clues**
  - Sets of entities that need to be linked
  - Each entity of one type is linked to *exactly one* entity of each other type (**Bijectivity**)
  - The links are *consistent* (**Transitivity**)

- **Objective:**
  - Find the *unique* assignment I that satisfies all of the constraints T (clues, bijectivity & transitivity)

Preliminaries

- A logic grid puzzle instance consists of:
  - Set of **Clues**
  - Sets of entities that need to be linked
  - Each entity of one type is linked to *exactly one* entity of each other type (**Bijectivity**)
  - The links are *consistent* (**Transitivity**)

- **Objective:**
  - Find the *unique* assignment I that satisfies all of the constraints T (clues, bijectivity & transitivity)

- *Example*

  "*Angie chose arrabiate sauce and Elisa did not chose pesto*"

Preliminaries

- A logic grid puzzle instance consists of:
  - Set of **Clues**
  - Sets of entities that need to be linked
  - Each entity of one type is linked to *exactly one* entity of each other type (**Bijectivity**)
  - The links are *consistent* (**Transitivity**)

- **Objective:**
  - Find the *unique* assignment I that satisfies all of the constraints T (clues, bijectivity & transitivity)

- *Example*

  "*Angie chose arrabiate sauce and Elisa did not chose pesto*"

Problem Definition

Explaining an inference step

Properties

- Combination of *already derived facts* **E** and *clues+constraints* **S**
- *Non-Redundant*
  - *None* of the facts or constraints can be removed while still explaining the newly derived information
- Interpretability (difficulty) quantified by cost function f, ex:
  - Promote the use of simple constraints
  - Penalize the use of multiple constraints (combination of constraints)

Problem Definition

Explanation-Production Problem

Generating the explanation sequence

- Given a *theory* (clues+constraints),
- Starting from an *initial assignment* I_0 (empty or partially filled grid),
- Find a non-redundant explanation sequence:

$$I_0 \xrightarrow[(E_1,S_1)]{expl.} I_1 \xrightarrow[(E_2,S_2)]{expl.} \ldots \xrightarrow[(E_n,S_n)]{expl.} I_n$$

- Minimize a predefined aggregate (max, average) over the costs of the explanation sequence

Non-redundant explanation

...in practice

- Smallest set of constraints and facts together explain a new fact
- Can be reduced to finding a Minimal Unsat Core/Subset (a.k.a MUC or MUS)
- Standard MUS extraction algorithm:
  - Subset Minimal (not further reducable), but not cardinality-minimal (the smallest one or optimal one)

(Greedy) Algorithm

- Generate candidate explanations using unsat-core extraction (one at a time)
- Iteratively grow the number of constraints used
- Prune search based on optimistic approximation of cost

(Greedy) Algorithm

... applied to logic grid puzzles

- Visual explanation interface
- Cost function:
  - Single implicit axiom: very cheap
  - Single constraint or 1 clue: less cheap
  - Multiple constraints: very expensive

DEMO

`https://bartbog.github.io/zebra/pasta/`

Algorithm

	capellini	farfalle	tagliolini	rotini	4	8	12	16	angie	damon	claudia	elisa
the_other_type1												
arrabiata_sauce		✔			-					-	-	
marinara_sauce			-		-							
puttanesca_sauce											-	
angie	-		-					-				
damon				-								
claudia												
elisa	-											
4			-									
8												
12												
16			-									

CLUES

1. The person who ordered capellini paid less than the person who chose arrabiata sauce
2. The person who ordered tagliolini paid more than Angie
3. The person who ordered tagliolini paid less than the person who chose marinara sauce
4. Claudia did not choose puttanesca sauce
5. The person who ordered rotini is either the person who paid $8 more than Damon or the person who paid $8 less than Damon
6. The person who ordered capellini is either Damon or Claudia
7. The person who chose arrabiata sauce is either Angie or Elisa
8. The person who chose arrabiata sauce ordered farfalle
9. Logigram Constraint
   - Transitivity constraint
   - Bijectivity
   - Combination of logigram constraints

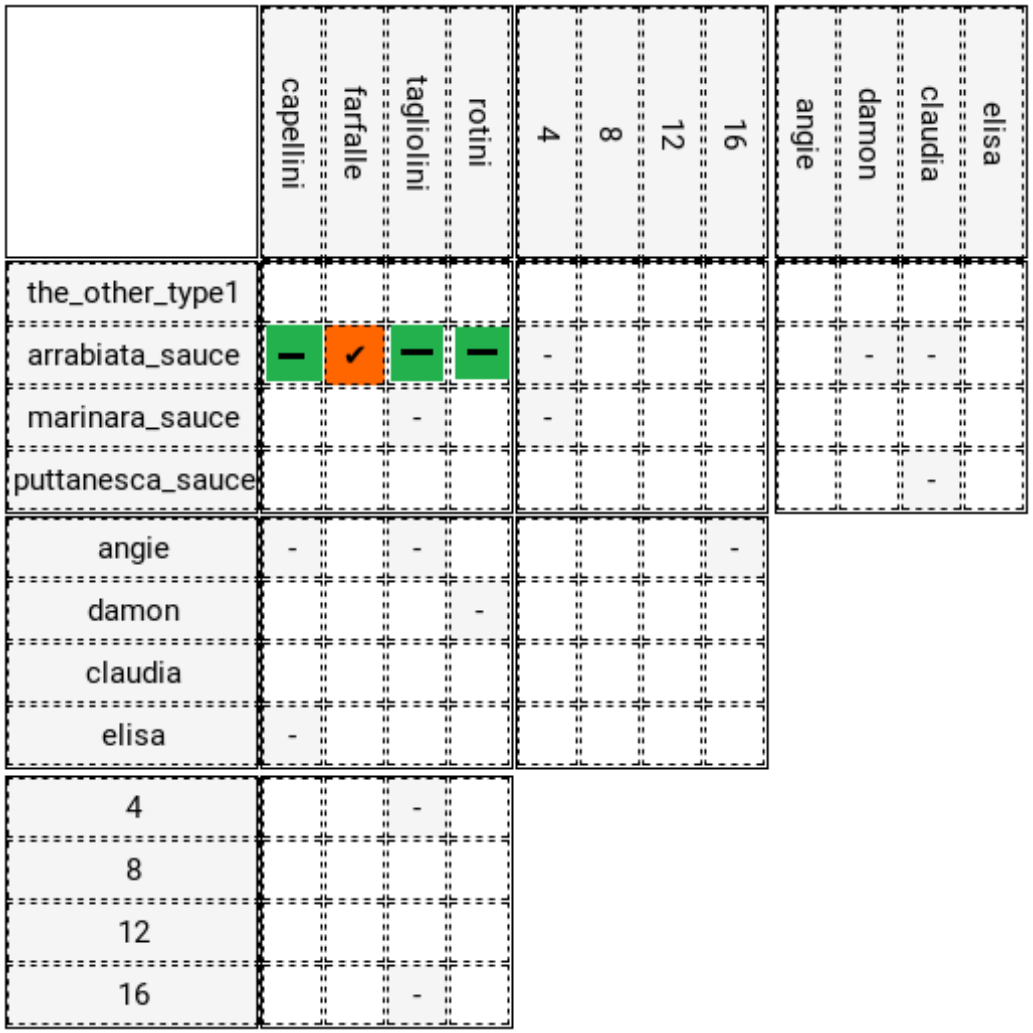
Algorithm

	capellini	farfalle	tagliolini	rotini	4	8	12	16	angie	damon	claudia	elisa
the_other_type1		−										
arrabiata_sauce		✓			-					-	-	
marinara_sauce		−	-		-							
puttanesca_sauce		−									-	
angie	-		-					-				
damon				-								
claudia												
elisa	-											
4			-									
8												
12												
16			-									

CLUES

1. The person who ordered capellini paid less than the person who chose arrabiata sauce
2. The person who ordered tagliolini paid more than Angie
3. The person who ordered tagliolini paid less than the person who chose marinara sauce
4. Claudia did not choose puttanesca sauce
5. The person who ordered rotini is either the person who paid $8 more than Damon or the person who paid $8 less than Damon
6. The person who ordered capellini is either Damon or Claudia
7. The person who chose arrabiata sauce is either Angie or Elisa
8. The person who chose arrabiata sauce ordered farfalle
9. Logigram Constraint
   - Transitivity constraint
   - Bijectivity
   - Combination of logigram constraints

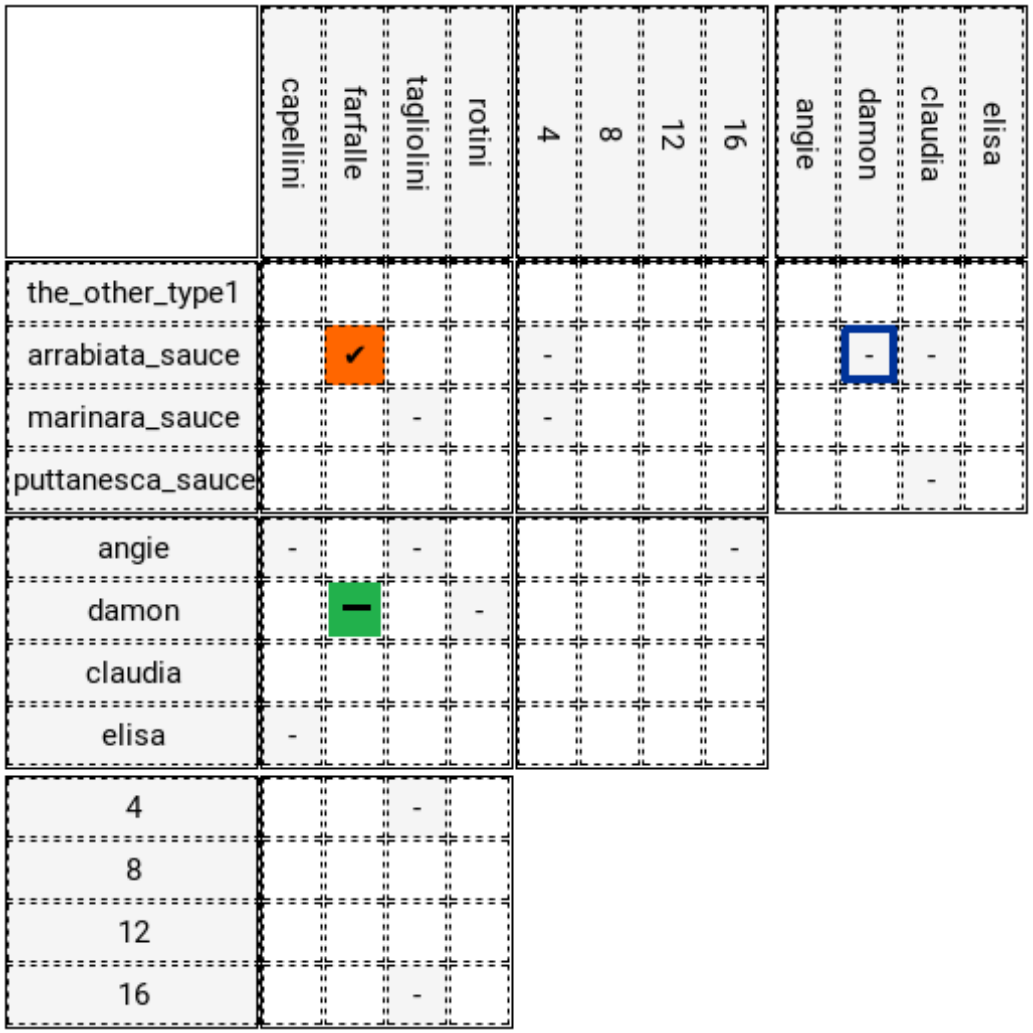
Algorithm



CLUES

1. The person who ordered capellini paid less than the person who chose arrabiata sauce
2. The person who ordered tagliolini paid more than Angie
3. The person who ordered tagliolini paid less than the person who chose marinara sauce
4. Claudia did not choose puttanesca sauce
5. The person who ordered rotini is either the person who paid $8 more than Damon or the person who paid $8 less than Damon
6. The person who ordered capellini is either Damon or Claudia
7. The person who chose arrabiata sauce is either Angie or Elisa
8. The person who chose arrabiata sauce ordered farfalle
9. Logigram Constraint
   - Transitivity constraint
   - Bijectivity
   - Combination of logigram constraints

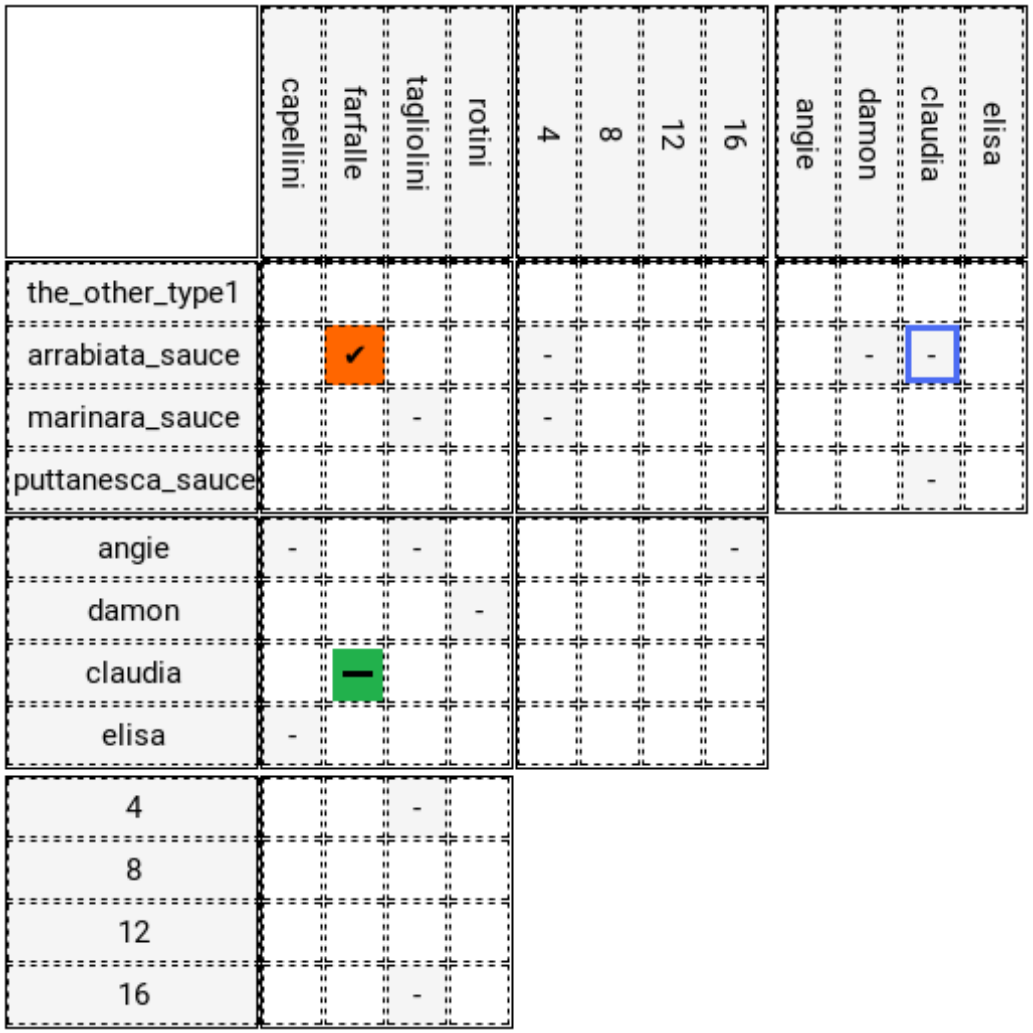
Algorithm



CLUES

1. The person who ordered capellini paid less than the person who chose arrabiata sauce
2. The person who ordered tagliolini paid more than Angie
3. The person who ordered tagliolini paid less than the person who chose marinara sauce
4. Claudia did not choose puttanesca sauce
5. The person who ordered rotini is either the person who paid $8 more than Damon or the person who paid $8 less than Damon
6. The person who ordered capellini is either Damon or Claudia
7. The person who chose arrabiata sauce is either Angie or Elisa
8. The person who chose arrabiata sauce ordered farfalle
9. Logigram Constraint
   - Transitivity constraint
   - Bijectivity
   - Combination of logigram constraints

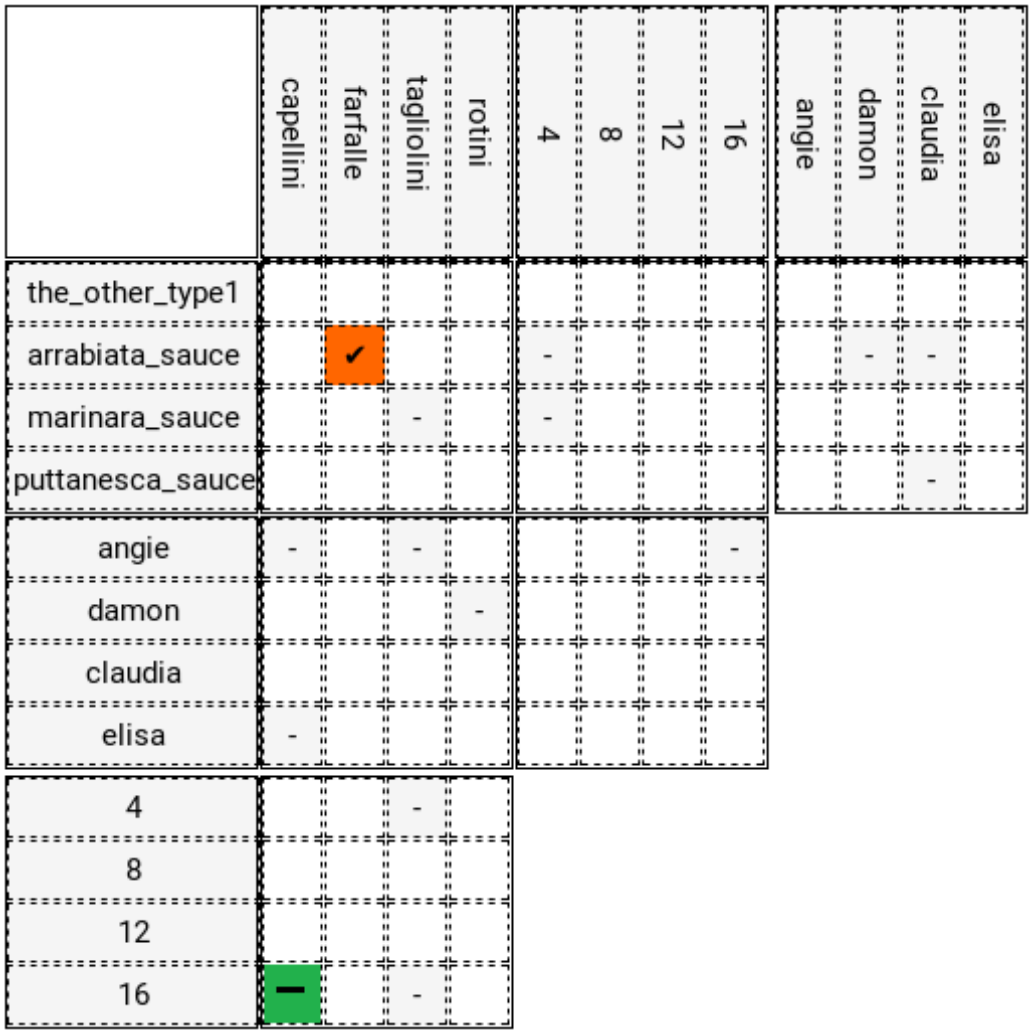
Algorithm



CLUES

1. The person who ordered capellini paid less than the person who chose arrabiata sauce
2. The person who ordered tagliolini paid more than Angie
3. The person who ordered tagliolini paid less than the person who chose marinara sauce
4. Claudia did not choose puttanesca sauce
5. The person who ordered rotini is either the person who paid $8 more than Damon or the person who paid $8 less than Damon
6. The person who ordered capellini is either Damon or Claudia
7. The person who chose arrabiata sauce is either Angie or Elisa
8. The person who chose arrabiata sauce ordered farfalle
9. Logigram Constraint
   - Transitivity constraint
   - Bijectivity
   - Combination of logigram constraints

Algorithm



CLUES

1. The person who ordered capellini paid less than the person who chose arrabiata sauce
2. The person who ordered tagliolini paid more than Angie
3. The person who ordered tagliolini paid less than the person who chose marinara sauce
4. Claudia did not choose puttanesca sauce
5. The person who ordered rotini is either the person who paid $8 more than Damon or the person who paid $8 less than Damon
6. The person who ordered capellini is either Damon or Claudia
7. The person who chose arrabiata sauce is either Angie or Elisa
8. The person who chose arrabiata sauce ordered farfalle
9. Logigram Constraint
   - Transitivity constraint
   - Bijectivity
   - Combination of logigram constraints

Contributions

In this paper, we

- ... formalize the *step-wise explanation problem*,
- ... propose a greedy algorithm (agnostic of actual propagators, consistency level, etc.),
- ... propose heuristics for *guiding* and *speeding-up the search for easy-to-understand explanations*,
- ... use logic grid puzzle to show the feasibility of the approach.

Use cases

- Teach humans how to solve a certain problem
- Quantify problem difficulty
- “Help” button
- Interactive configuration/planning/scheduling

Conclusion and Future work

Current work

- Extension to support nested explanations **?**
- Optimal explanations for inference step using unsat-core optimization **?**

Conclusion and Future work

Current work

- Extension to support nested explanations [?]
- Optimal explanations for inference step using unsat-core optimization [?]

Future work

- Optimize the explanation sequence
- Learning the optimization function (from humans)
- Interactive configuration [???]

References

Bart Bogaerts, Emilio Gamba, and Tias Guns. A framework for step-wise explaining how to solve constraint satisfaction problems, 2020.

Pierre Carbonnelle, Bram Aerts, Marjolein Deryck, Joost Vennekens, and Marc Denecker. An interactive consultant. In *BNAIC/BENELEARN*, 2019.

Maria Fox, Derek Long, and Daniele Magazzeni. Explainable planning. *arXiv preprint arXiv:1709.10256*, 2017.

Alexey Ignatiev, Mikoláš Janota, and Joao Marques-Silva. Quantified maximum satisfiability. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 250–266. Springer, 2013.

Pieter Van Hertum, Ingmar Dasseville, Gerda Janssens, and Marc Denecker. The kb paradigm and its application to interactive configuration. *Theory and Practice of Logic Programming*, 17(1):91–117, 2017.