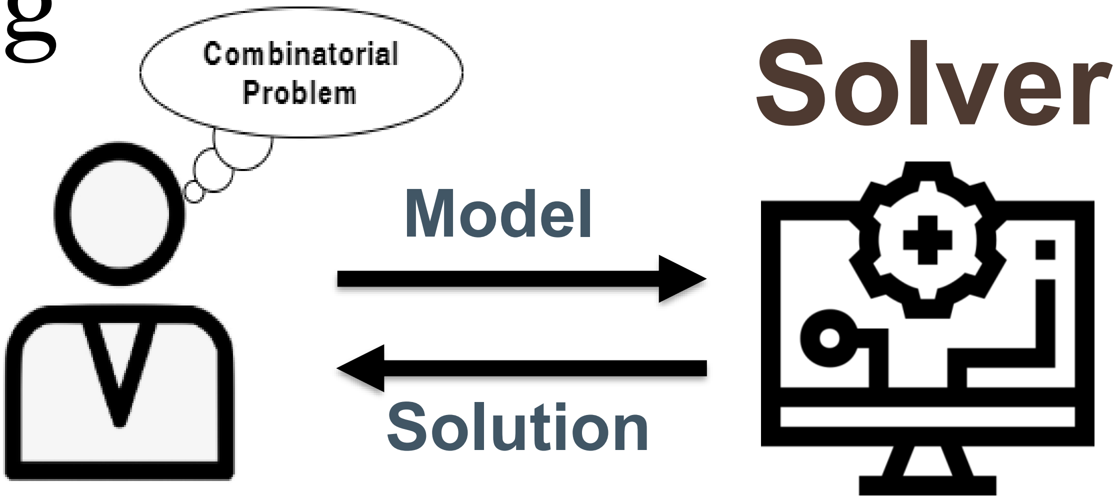
ML-based Generalization Queries for Interactive Constraint Acquisition

Dimos Tsouros, Senne Berden, Tias Guns

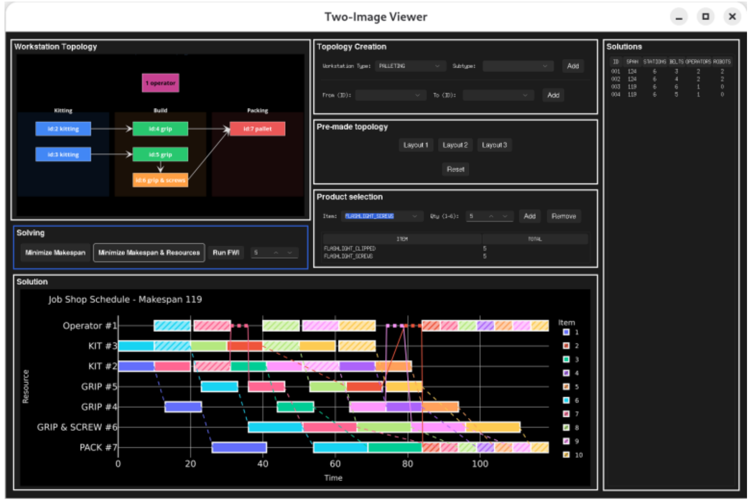
19/07/2026

Constraint Modeling

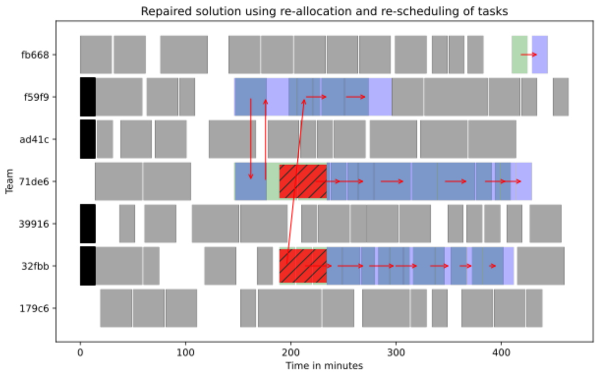
Model + Solve paradigm



- Decision Variables
- Domains
- Constraints
- Objective (optional)



	Week 1							Week 2							Total shifts
	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
name															
Megan	F	D	D	D	D	F	F	D	D	F	F	D	D	D	9
Katherine	D	D	D	D	D	F	F	D	D	F	F	D	D	F	9
Robert	D	D	D	F	F	D	D	F	F	D	D	D	F	F	8
Jonathan	D	D	F	F	F	D	D	D	D	D	F	F	F	F	7
William	F	D	D	D	D	F	F	D	D	F	F	D	D	D	9
Richard	D	D	D	D	D	F	F	D	D	F	F	F	D	D	9
Kristen	F	F	D	D	D	F	F	D	D	D	F	F	D	D	8
Kevin	D	D	F	F	F	F	F	F	D	D	D	D	D	F	7
Cover D	5/5	7/7	6/6	5/4	5/5	2/5	2/5	6/6	7/7	4/4	2/2	5/5	6/6	4/4	14



Constraint Solving technologies are powerful, but modeling requires expertise in:

- Domain Knowledge
- Mathematical Skills
- Solver Formalisms

H.F.R.I.
Hellenic Foundation for Research & Innovation

Making Constraint Solving More Accessible

Modelling Tools

MiniZinc XCSP CPMpy Essence prime

Constraint Acquisition

Users provide or classify (non-)solutions

From Natural Language

Textual descriptions of combinatorial problems

Making Constraint Solving More Accessible

Modelling Tools

CPMpy

Essence prime

Modeling is not a one-shot procedure!

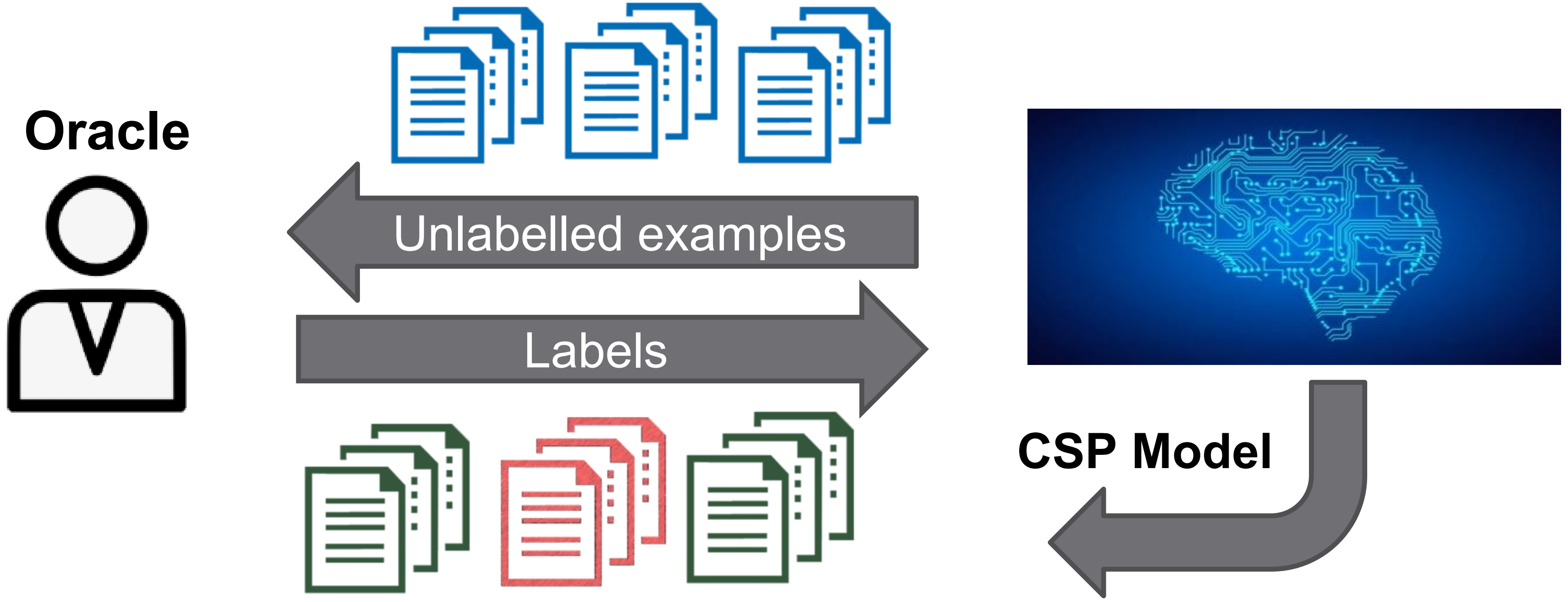
Constraint Acquisition

Users provide or classify (non-)solutions

From Natural Language

Textual descriptions of combinatorial problems

Interactive Constraint Acquisition (CA)



Membership query

1	1	3	4
3	2	1	1
2	2	3	1
2	3	4	3

Answer: Negative in both of them (a constraint is violated)

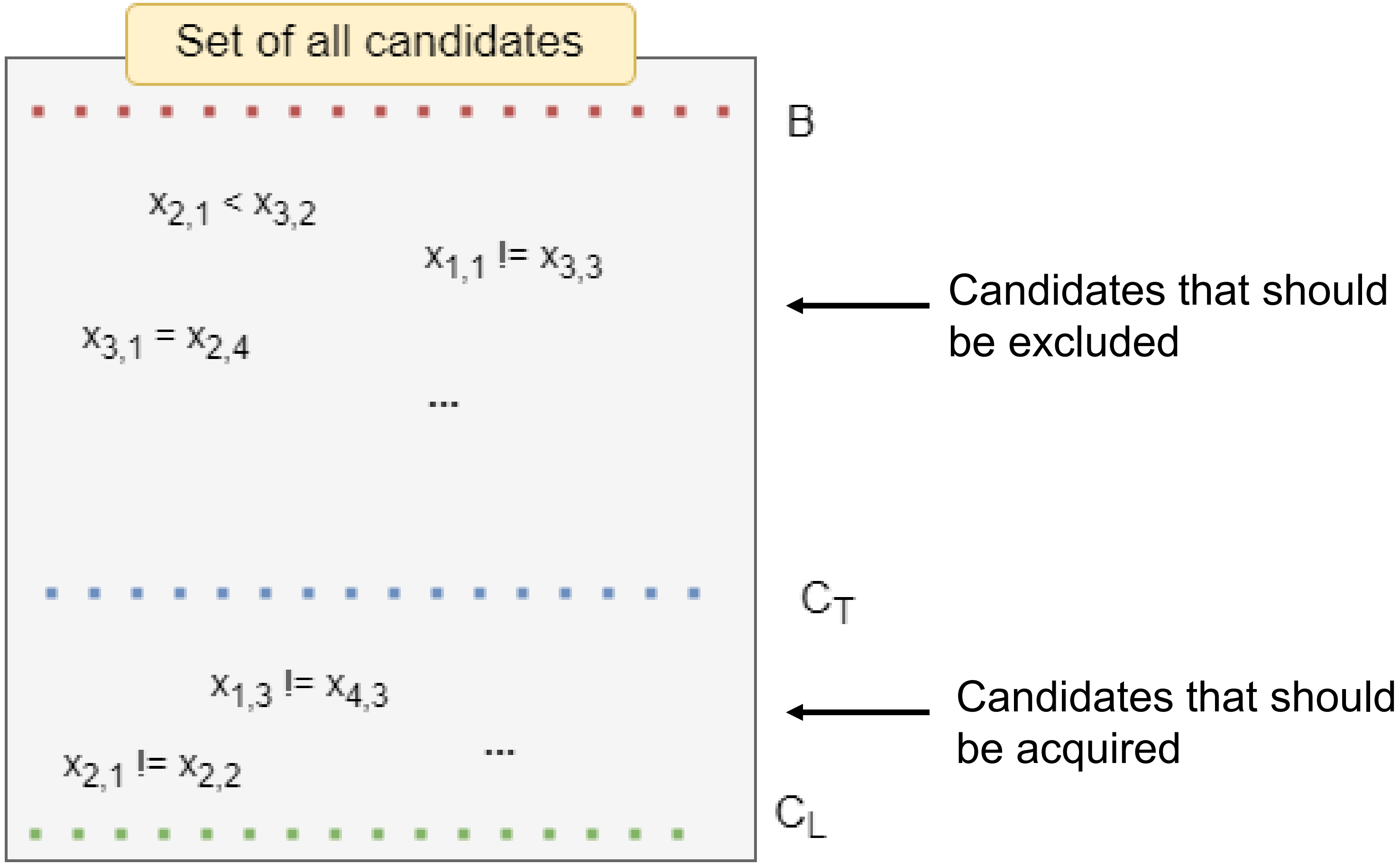
Partial query

1	1	-	4
3	-	1	-
-	-	-	-
2	-	-	-

Learning Constraints through Partial Queries, C. Bessiere et al., AIJ, 2023

Interactive CA

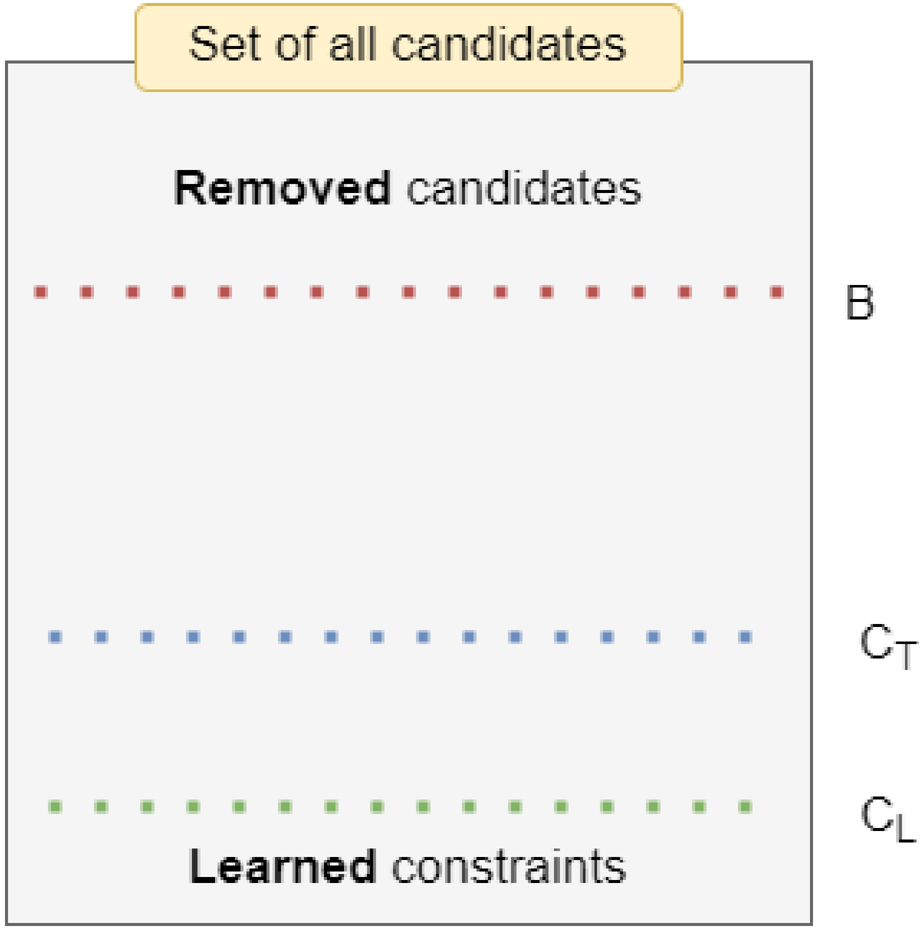
- **B**: set of (remaining) candidate constraints
- **C_T**: target set of constraints
- **C_L**: learned set of constraints



Interactive CA

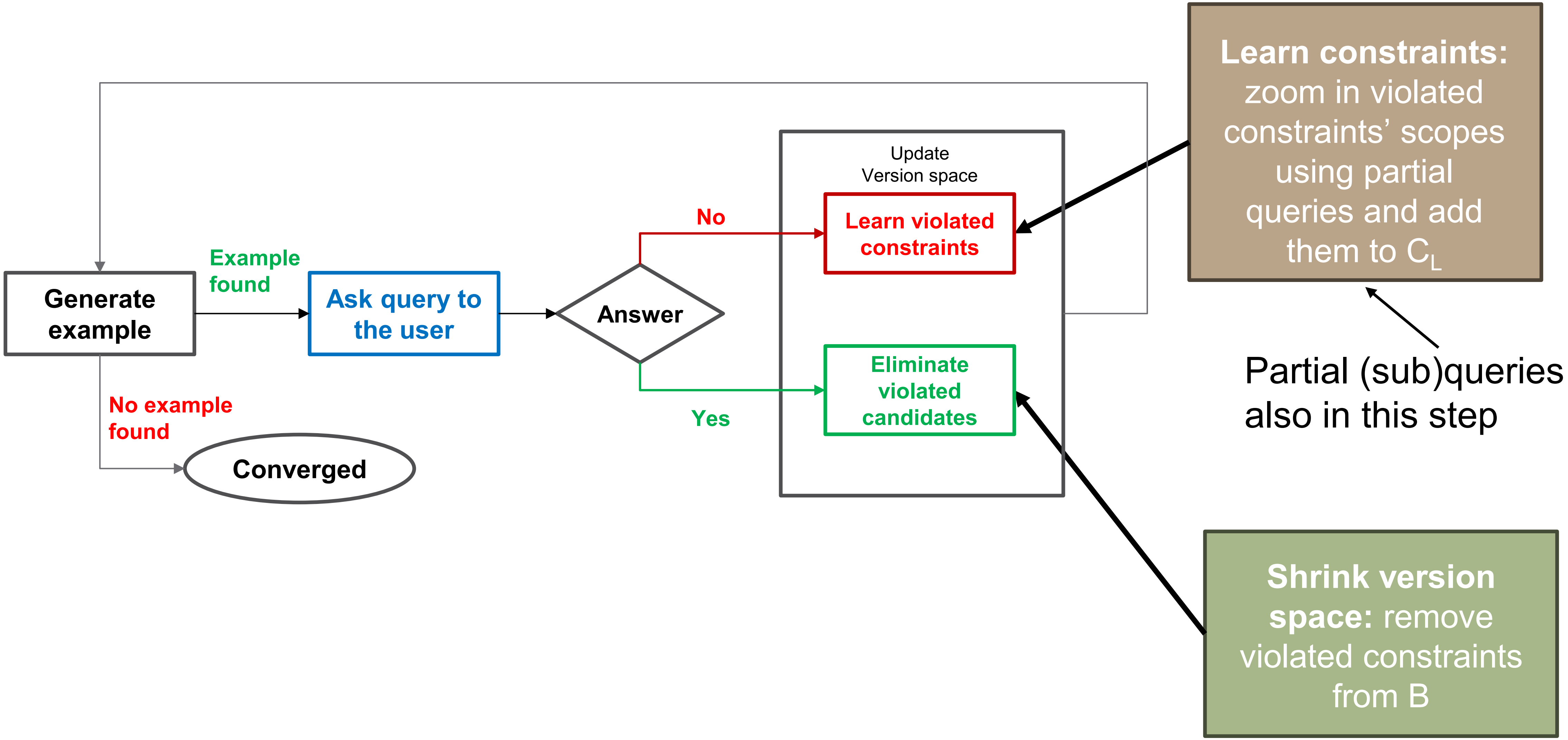
During the learning process:

- Constraints are removed from *B*
- Constraints are added to C_L



- **B**: set of (remaining) candidate constraints
- $\mathbf{C_T}$: target set of constraints
- $\mathbf{C_L}$: learned set of constraints

Interactive Constraint Acquisition



Learning Constraints through Partial Queries, C. Bessiere et al., AIJ, 2023

Main challenge for human users: Slow learning

Standard interactive CA learns one ground constraint at a time. Even when many constraints share the same underlying **pattern**.

Example from Exam Timetabling

$course[i,j] \neq course[k,l]$ → Broader rule: all courses should be in different slots

- Learning each constraint separately is slow and repeats work.
- Learning the pattern can **reduce** the number of queries.

Generalization queries in interactive CA

Existing type-based generalization

- Generalization based on user-given variable types (e.g. row, column)
- Ask whether a learned ground constraint should hold for every combination of variables with this sequence of types.

learned ground constraint cell(1,1) ≠ cell(1,9)

Which sequence of types?

- *Not fully expressive*
- *Many failing generalization attempts*

Var1: column1 + Var2: row1 ?	Var1: row1 + Var2: column9 ?
Var1: column1 + Var2: column9	Var1: row1 + Var2: row1 √

Constraint Specification (CS)

H.F.R.I. Hellenic Foundation for Research & Innovation

Capturing **problem-level requirements**

«All variables in the same row should have a different value”

```
Foreach row ∈ all_rows:
    Foreach cell1, cell2 ∈ all_pairs(row):
        c ← cell1 ≠ cell2
```

- **What kind of constraint?** ≠
- **Which variables?** Same row
- **Which combinations?** All pairs

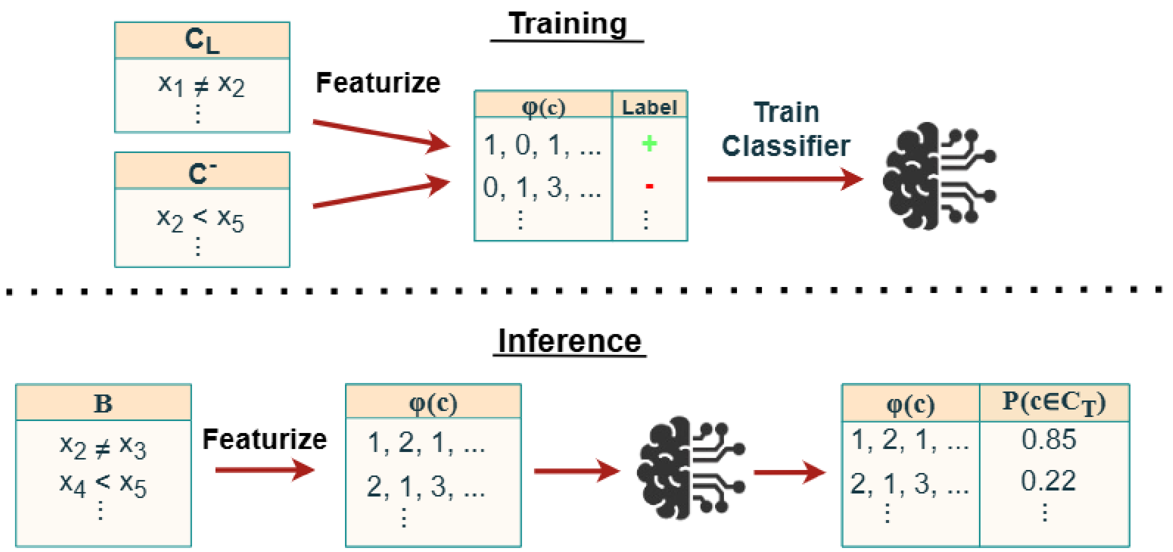
Elements of a CS

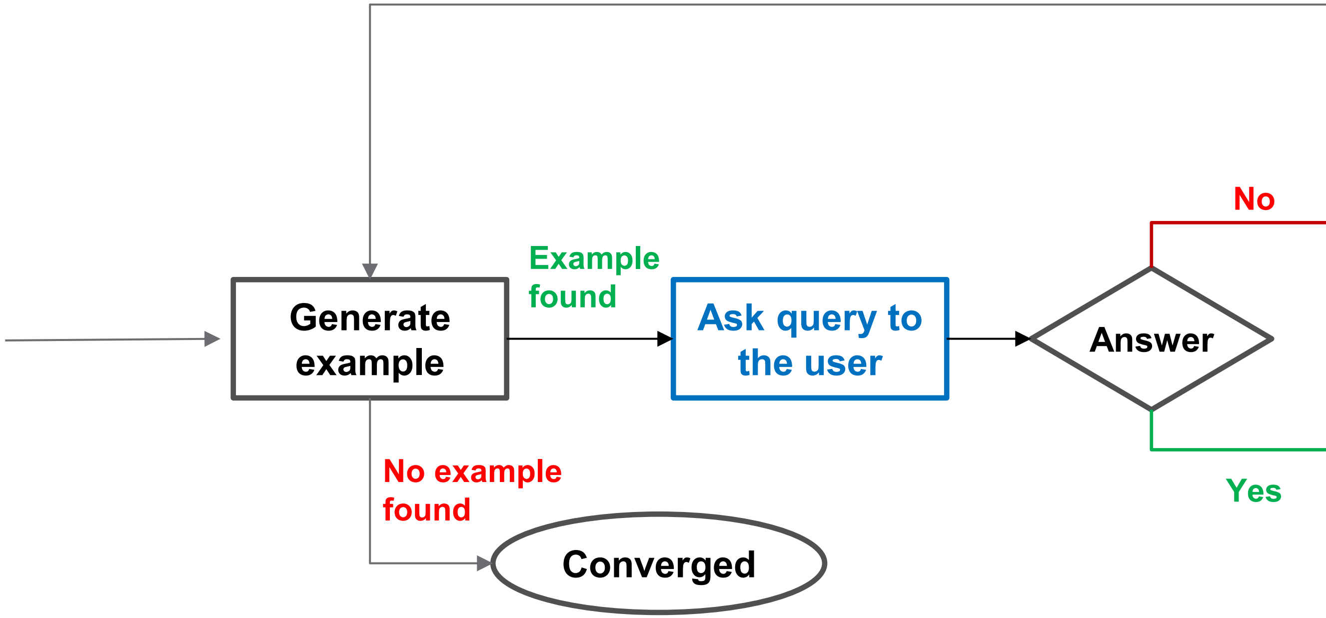
```
Foreach Y ∈ 𝒫(V_T):
  Foreach scope ∈ combinations(Y, arity(r),
                                        S):
    c ← (rel(c) = r, var(c) = scope)
```

- **Relation** r: What constraint?
- **Partitioning** function G: How to group the variables?
- Sequence **conditions** S: Which combinations in each group?

Can we ***generalize*** *the learned ground constraints to constraint specifications?*

Existing in CA:
Using ML to guide query generation



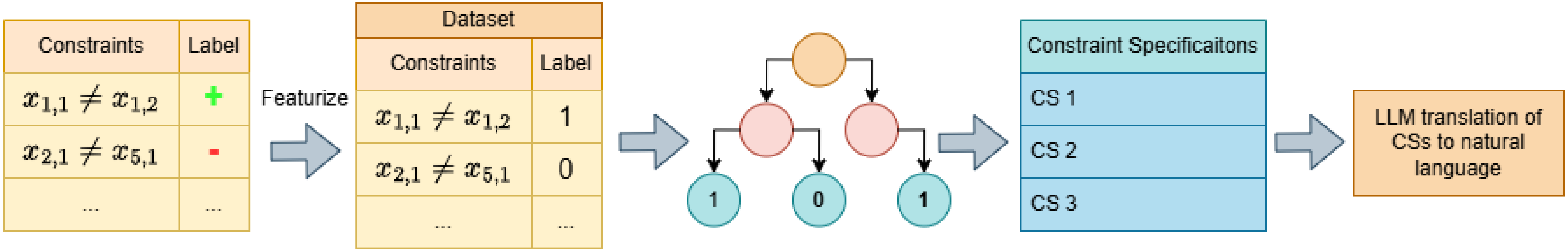


The classifier is learning the patterns to predict constraints

Let's use what the classifier learns to generalize

Framework for extracting constraint specifications using ML

1. Build dataset of constraints during acquisition
2. Train **a rule-based** classifier
3. Extract Constraint Specifications
4. Query the user in natural language



GQ-Gen: Generalization query generation

Extract CSs using ML

Feature Representation of Constraints

Feature group	Example features
Relation	relation name, constants, arity
Partitioning	same row / column / latent dims
Conditioning	index avg, min/max, distances
Label	constraint is in target C_T or not

Feature groups: building blocks for extracting CSs from *positive* decision rules.

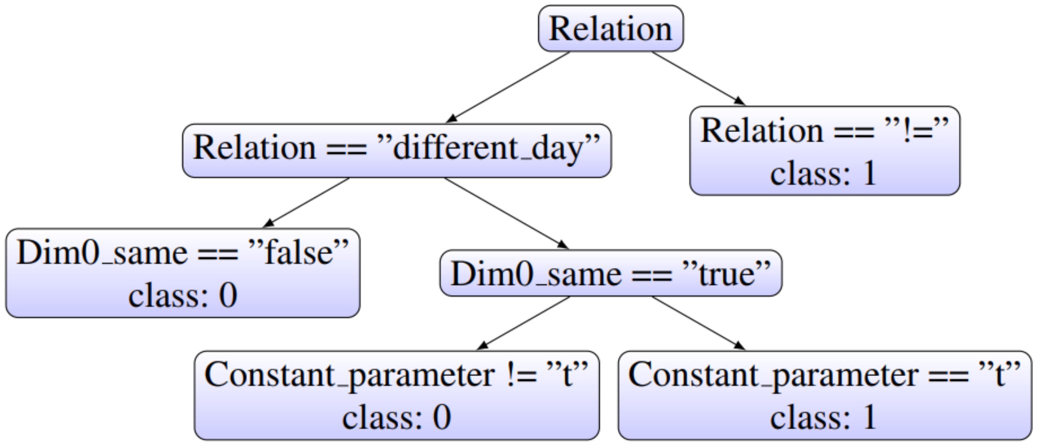
Extracting CSs

1. *Get the positive decision rule*
2. *Check what features it involves*
3. *Build the CS by putting the features in their respective group*

Example – Exam Timetabling

H.F.R.I. Hellenic Foundation for Research & Innovation

1. Decision Tree learned



2. Decision Rules

```
r1: Relation == "different_day"
& Dim0_same == "false"
then 0
r2: Relation == "different_day"
& Dim0_same == "true"
& Constant_parameter != "t"
then 0
r3: Relation == "different_day"
& Dim0_same == "true"
& Constant_parameter == "t"
then 1
r4: Relation == "!=" then 1
```

3. Extracted Constraint Specifications

```
Foreach row ∈ all_rows:
  Foreach scope ∈ all_pairs(row):
    c ← (rel(c) = "different_day(t)",
                var(c) = scope)
```

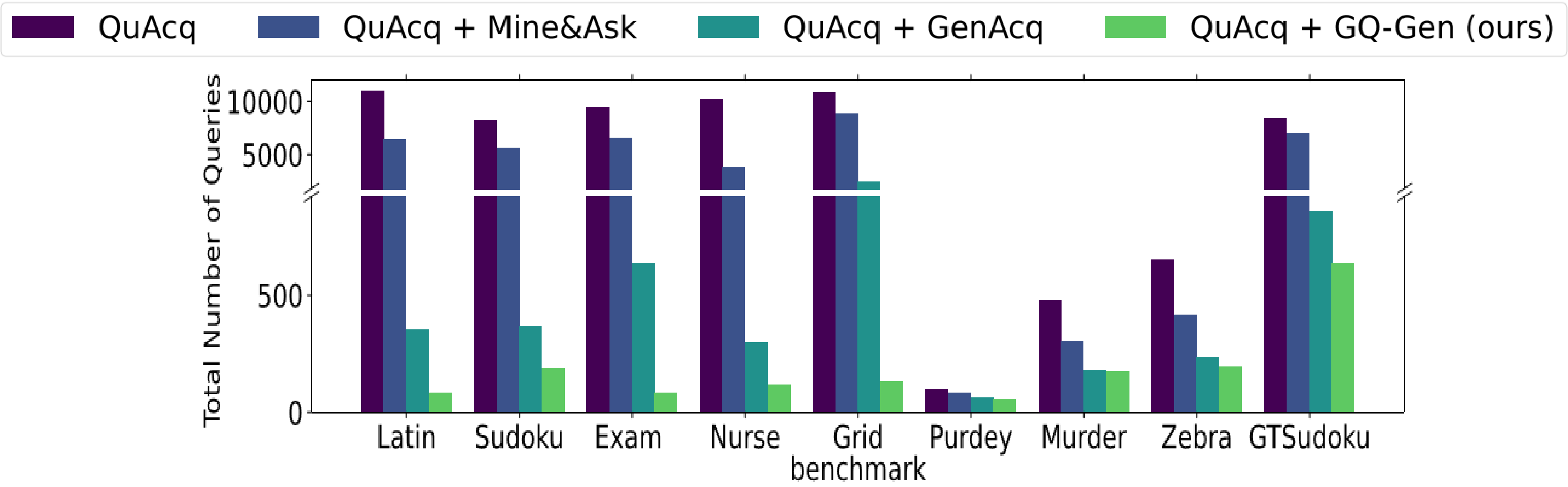
```
Foreach scope ∈ all_pairs(V):
  c ← (rel(c) = "!=", var(c) = scope)
```

4. Natural language queries

Within each semester, should every pair of exams be scheduled on different days?

In the entire timetable, should every pair of exams be assigned to different slots?

Results - # of queries



Compared against

1. QuAcq (No generalization)
2. GenAcq (type-based generalization, given types)
3. Mine&Ask (detecting types for generalization)

9 benchmarks:

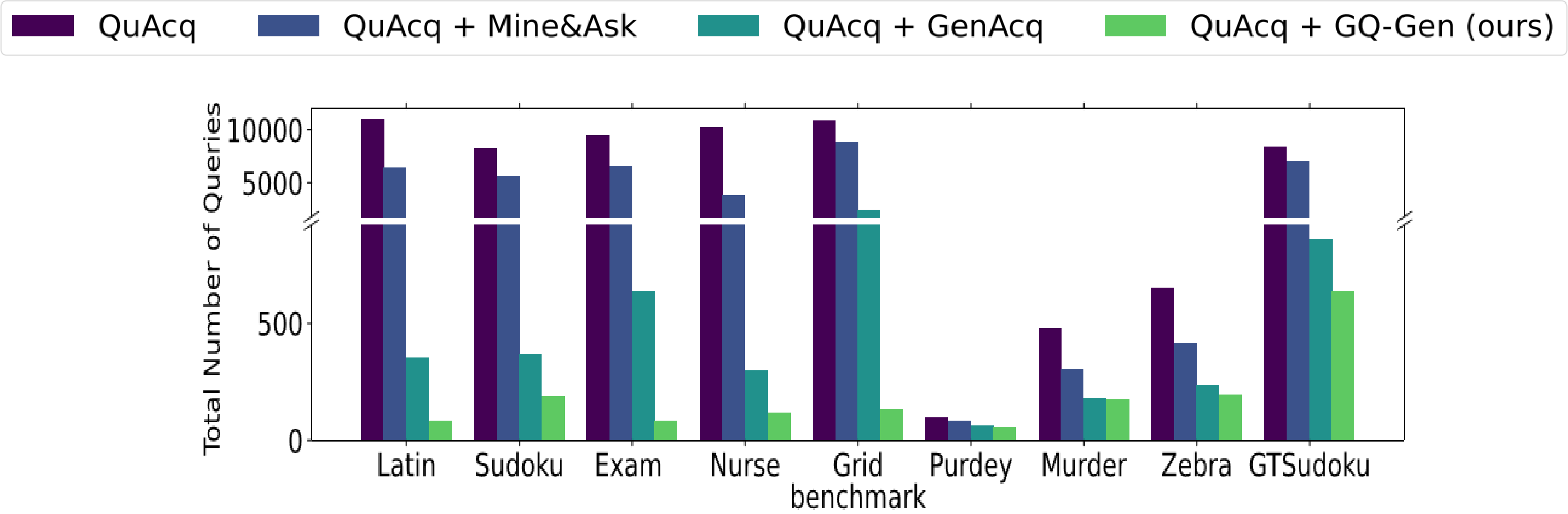
Fully structured

1. **Sudoku**
2. **Latin** squares
3. **Exam** timetabling
4. **Nurse** rostering
5. Distance-based **Grid** colouring

Partially structured

1. Purdey
2. Murder
3. Zebra
4. GTSudoku

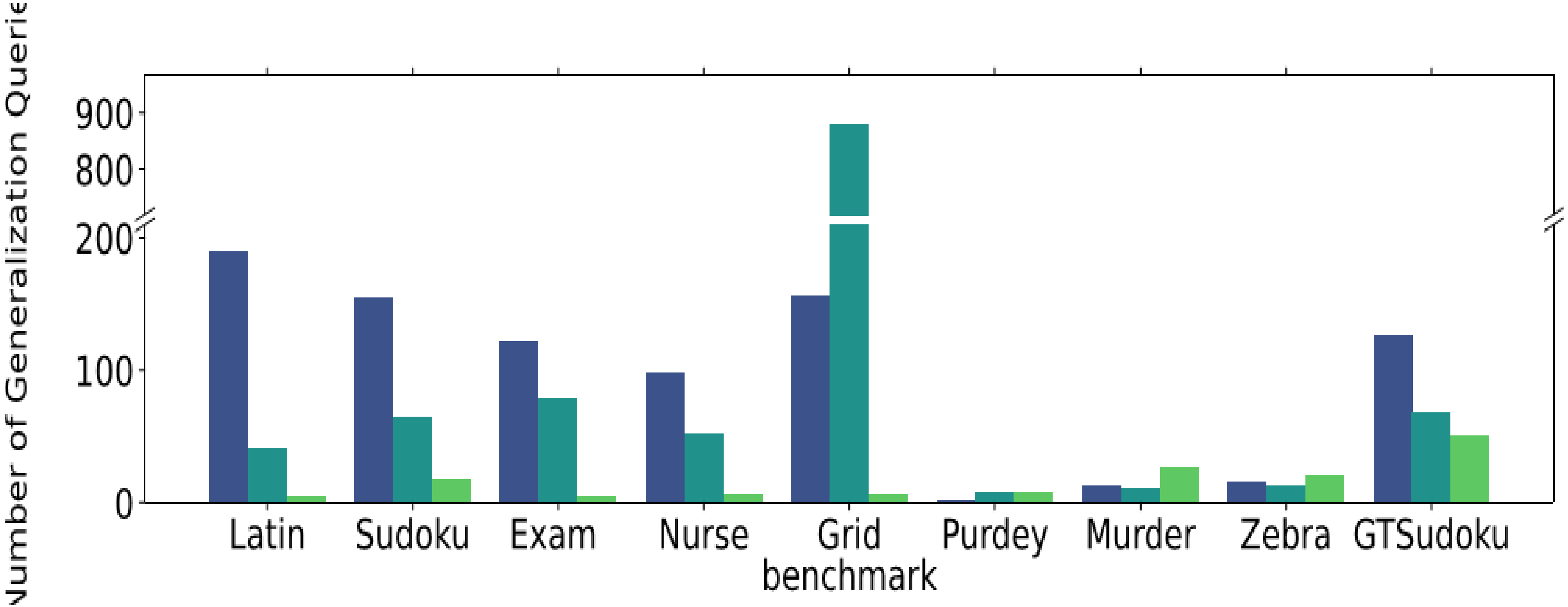
Results - # of queries



Up to 99% fewer queries vs QuAcq

Up to 95% fewer queries vs GenAcq

Results - # of Generalization Queries



# of generalization queries stays low!

Scalability

Hellenic Foundation for Research & Innovation

Scales much better than existing interactive CA methods

Additional queries in larger problem sizes are (mostly) queries to learn initial constraints before generalization

All Queries, Exam Timetabling

Nr. Constraints:	*77*	*143*	*218*	*348*	*488*	*716*	*959*	*1317*	*1699*	*2231*
QuAcq	377	734	1248	2046	2916	4338	5935	7994	10289	13108
QuAcq + GenAcq	101	149	191	233	385	374	542	557	818	826
QuAcq + GQ-Gen	**32**	**35**	**37**	**37**	**46**	**60**	**52**	**79**	**66**	**73**

All Queries, Nurse Rostering

Nr. Constraints:	*53*	*171*	*216*	*394*	*476*	*640*	*722*	*804*	*1275*	*1854*
QuAcq	403	1671	2338	4210	5368	8022	9273	10546	>15K	>15K
QuAcq + GenAcq	75	125	156	182	222	308	356	394	423	442
QuAcq + GQ-Gen	**45**	**115**	**100**	**122**	**232**	**93**	**91**	**202**	**286**	**228**

All Queries, Grid Coloring

Nr. Constraints:	*168*	*270*	*396*	*465*	*546*	*627*	*708*	*813*	*918*	*1140*
QuAcq	1229	2359	3831	4621	5628	6702	7868	9325	10852	13828
QuAcq + GenAcq	343	600	930	1138	1349	1494	1751	2017	2305	5016
QuAcq + GQ-Gen	**92**	**162**	**114**	**138**	**215**	**136**	**150**	**119**	**189**	**148**

User study

- Natural-language formulations of CS-based queries were evaluated by 19 participants.
- Results indicate that users can understand and answer the generated queries quickly and accurately.

79%	89%	69%	70%
High clarity score	correct answers	highest confidence	<30s response

Conclusions

- Pattern recognition with ML moves from guiding learning to being the **main learning component.**
- Generalization can make interactive CA substantially more query-efficient and more human-friendly.
- Next steps:
  - LLM-based interaction – extending LLMACQ
    - Generalization without explicit candidate constraints
  - Data-based generalization

Related material

PyConA

https://github.com/CPMpy/PyConA

ML-guided Interactive Constraint Acquisition

https://doi.org/10.1613/jair.1.19524

CP + ML Survey - JAIR

https://jair.org/index.php/jair/article/view/19533