Iterative Constraint Reasoning - Dynamic Constraint Reasoning in Time Space

Brown-Bag Session
October 25th, 2022

Lothar Hotz

Rainer Herzog

Stephanie von Riegen

Hamburger Informatik Technologie-Center

Universität Hamburg, Germany

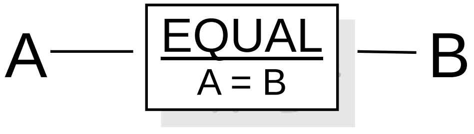
Sudoku as we know it

		4	**6**		**9**			**2**
9								
	3			**5**		**6**	**4**	
2					**5**	**7**		**4**
	5						**2**	
3		**9**	**2**					**1**
	4	**2**		**8**			**1**	
								7
6			**3**		**7**	**2**		

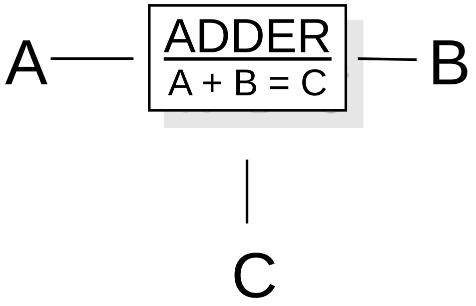
5	8	**4**	**6**	1	**9**	3	7	**2**
9	2	6	4	7	3	1	8	5
1	**3**	7	8	**5**	2	**6**	**4**	9
2	6	8	1	9	**5**	**7**	3	**4**
4	**5**	1	7	3	8	9	**2**	6
3	7	**9**	**2**	6	4	8	5	**1**
7	**4**	**2**	9	**8**	6	5	**1**	3
8	9	3	5	2	1	4	6	**7**
6	1	5	**3**	4	**7**	**2**	9	8

Each field a variable with domain {1…9}.
Sudoku rules as constraints.
Given numbers as requirements.

Constraints



- Constraints represent relations between variables
- Constraint-Net
- Constraints are multi-directional
- Holy Grail of programming:
  - the user states the problem, the computer solves



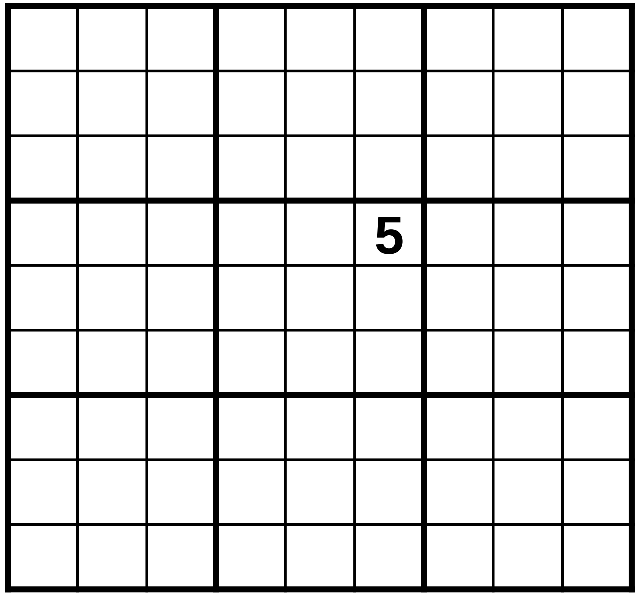
Sudoku as we know it

		4	**6**		**9**			**2**
9								
	3			**5**		**6**	**4**	
2					**5**	**7**		**4**
	5						**2**	
3		**9**	**2**					**1**
	4	**2**		**8**			**1**	
								7
6			**3**		**7**	**2**		

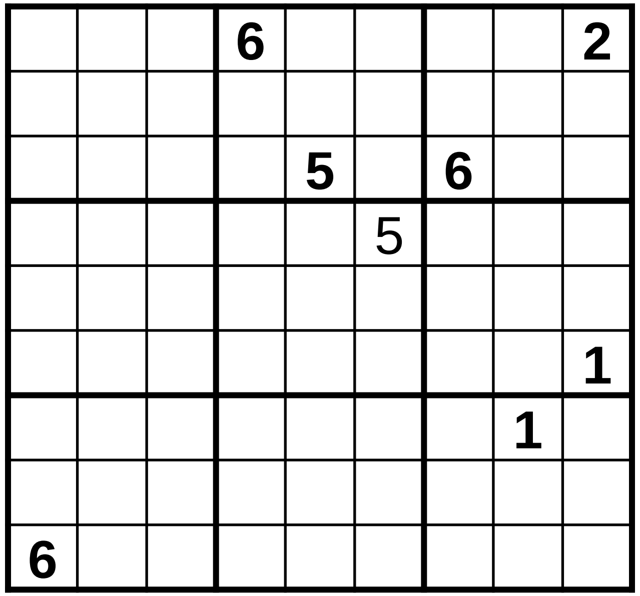
5	8	**4**	**6**	1	**9**	3	7	**2**
9	2	6	4	7	3	1	8	5
1	**3**	7	8	**5**	2	**6**	**4**	9
2	6	8	1	9	**5**	**7**	3	**4**
4	**5**	1	7	3	8	9	**2**	6
3	7	**9**	**2**	6	4	8	5	**1**
7	**4**	**2**	9	**8**	6	5	**1**	3
8	9	3	5	2	1	4	6	**7**
6	1	5	**3**	4	**7**	**2**	9	8

Each field a variable with domain {1…9}.
Sudoku rules as constraints.
Given numbers as requirements.

Iterative Sudoku

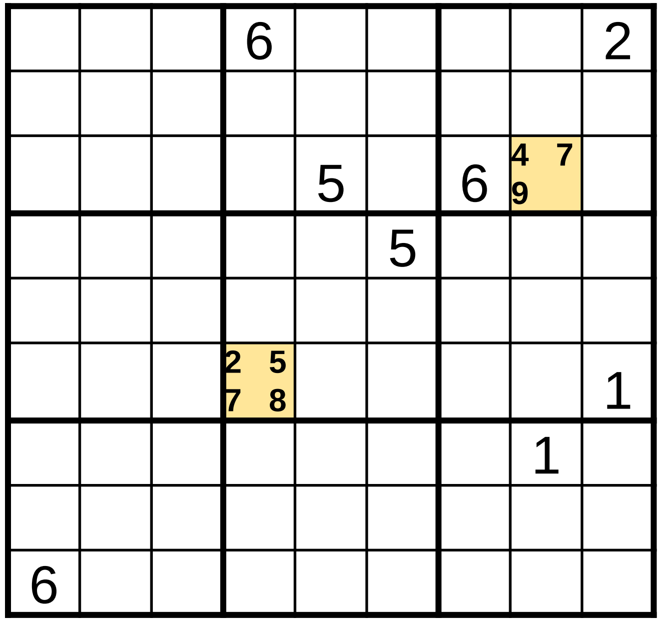


t_1



t_2

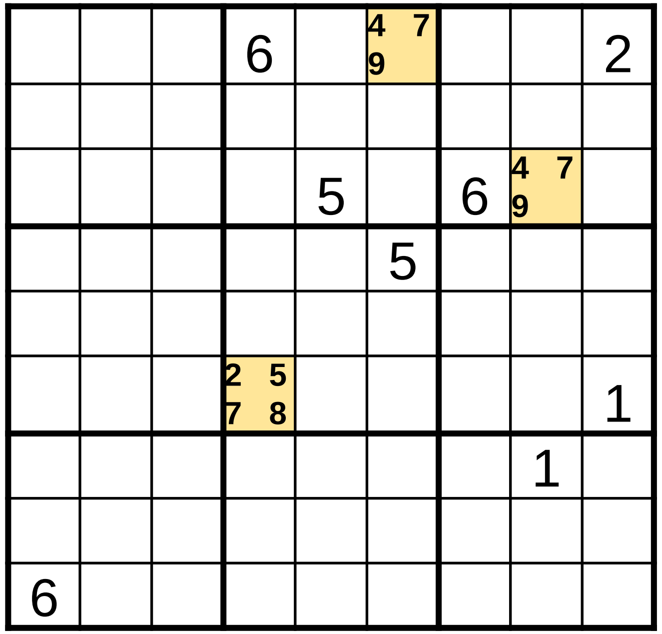
Give numbers not all together but over time



t_3

Give sets of numbers

(4 0.3) (7 0.4) (9 0.3)



t_4

Provide confidence values

Why?

Number detector

Role detector

Object detector

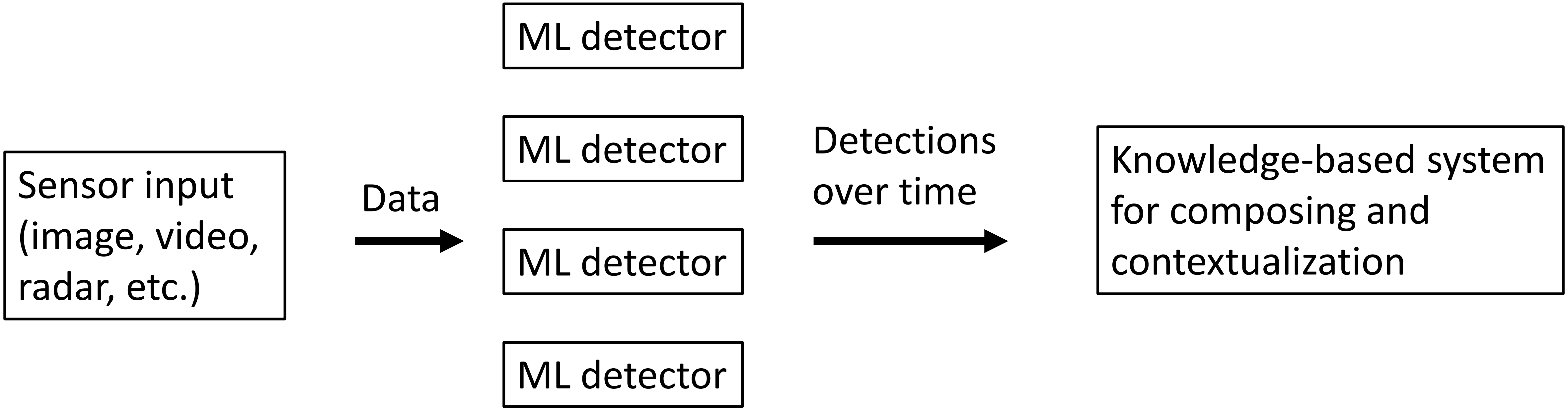
Object tracking

Action detector

Etc.

Automated capture and evaluation of a complete soccer match based on video footage

Hybrid AI Architecture

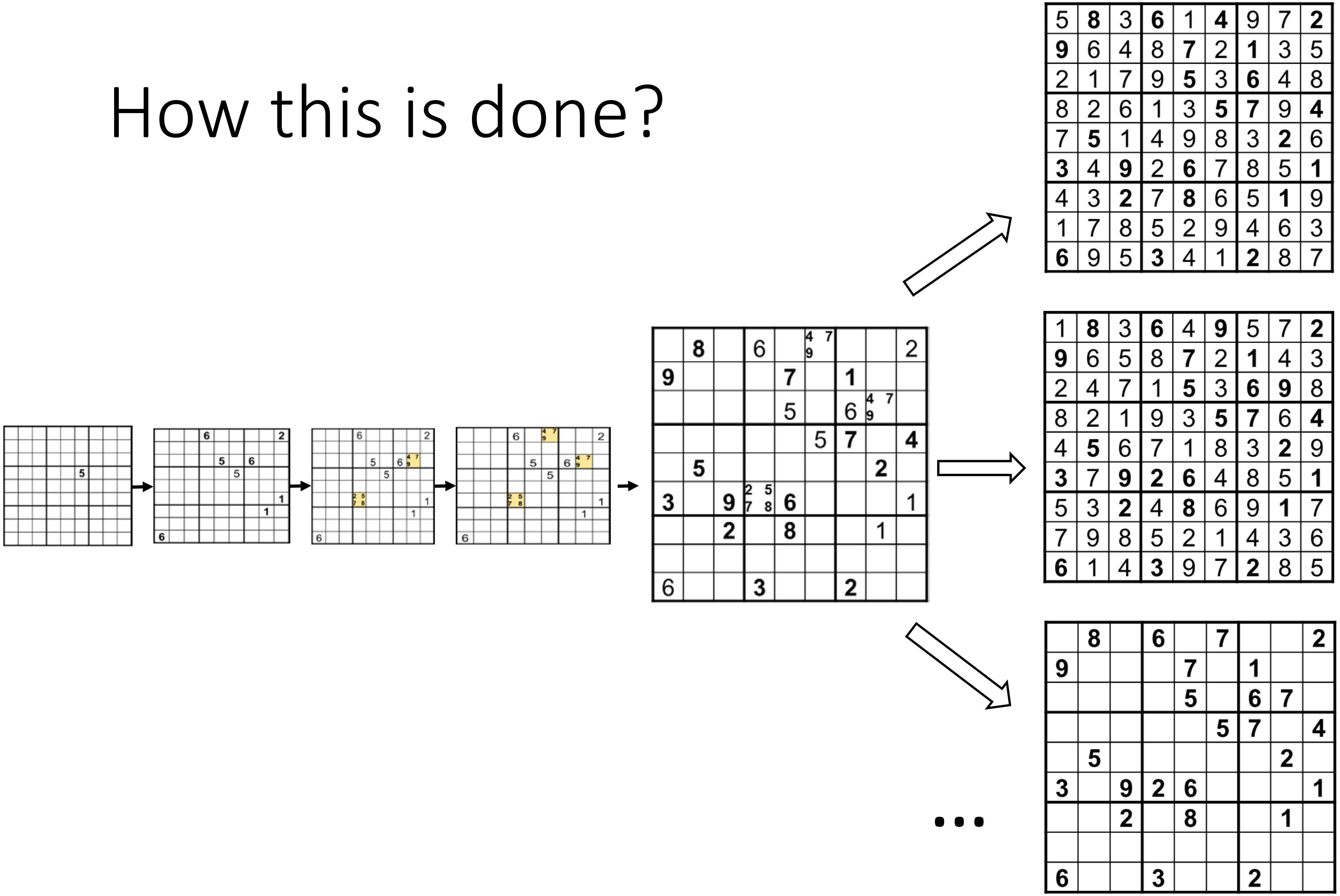


ML: Machine Learning

Iterative Constraint Reasoning

- Huge number of variables and constraints, too many for solving in one step
  - For soccer: each frame contains up to 25 actors and a huge number of positions on the field
  - Future video frames not known, i.e., requirements appear over time
  - Provide in between (non-optimal) solutions
- Iterative Sudoku as an example, perhaps a benchmark, for:
  - Combining uncertain information
  - Handling dynamically given information/data
  - Problem scaling via Sudoku puzzle size (not only 9x9)
  - Iterative constraint reasoning

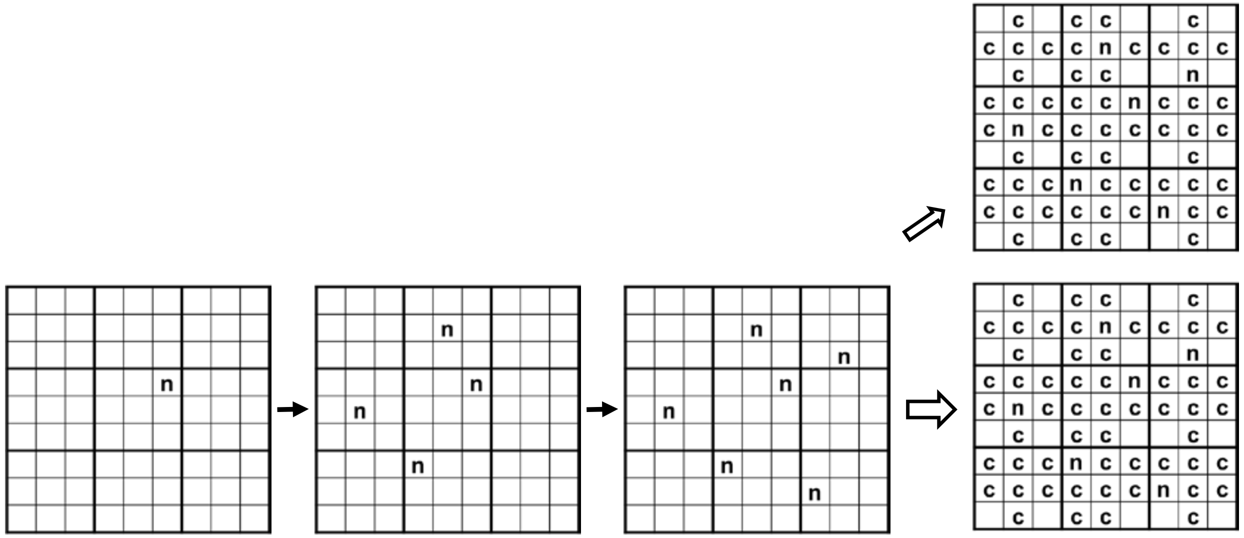
How this is done?

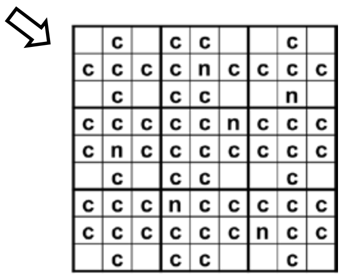


One of 11 solutions

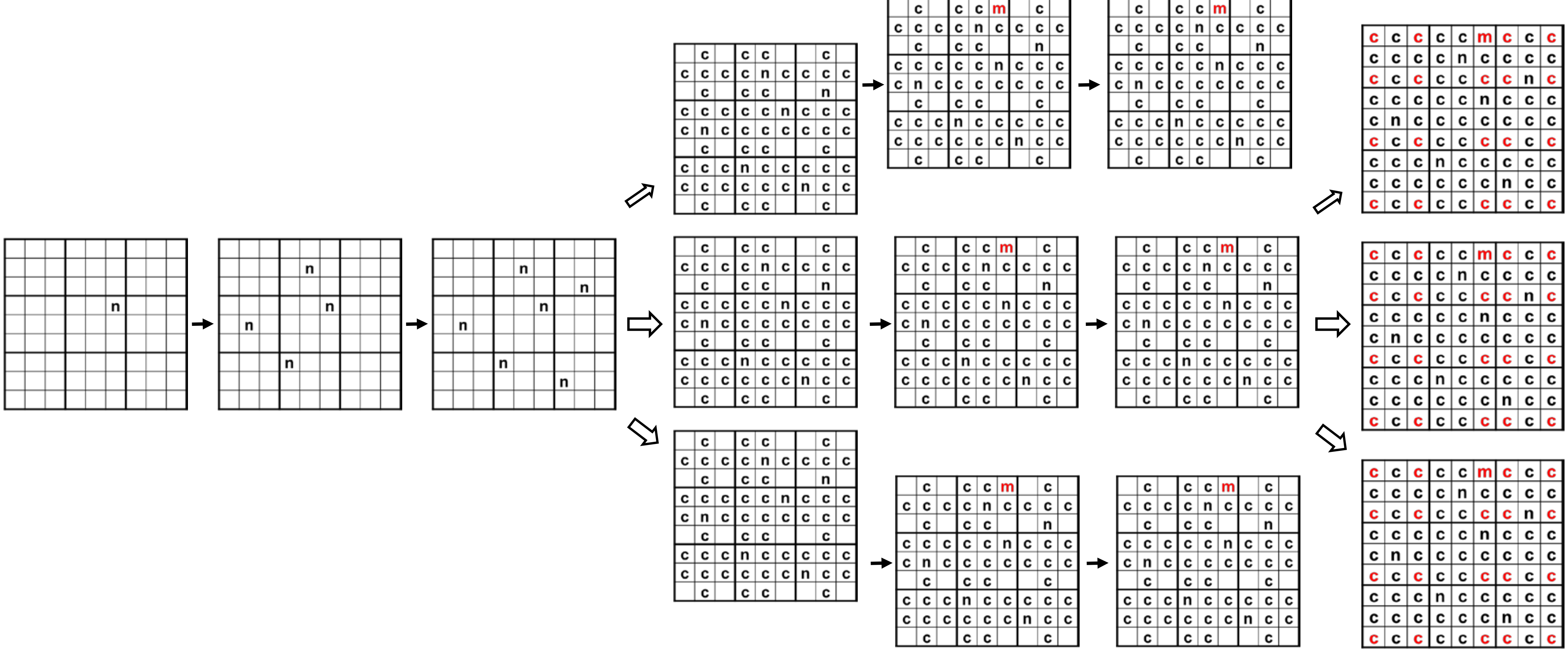
One of 72 solutions

0 solutions

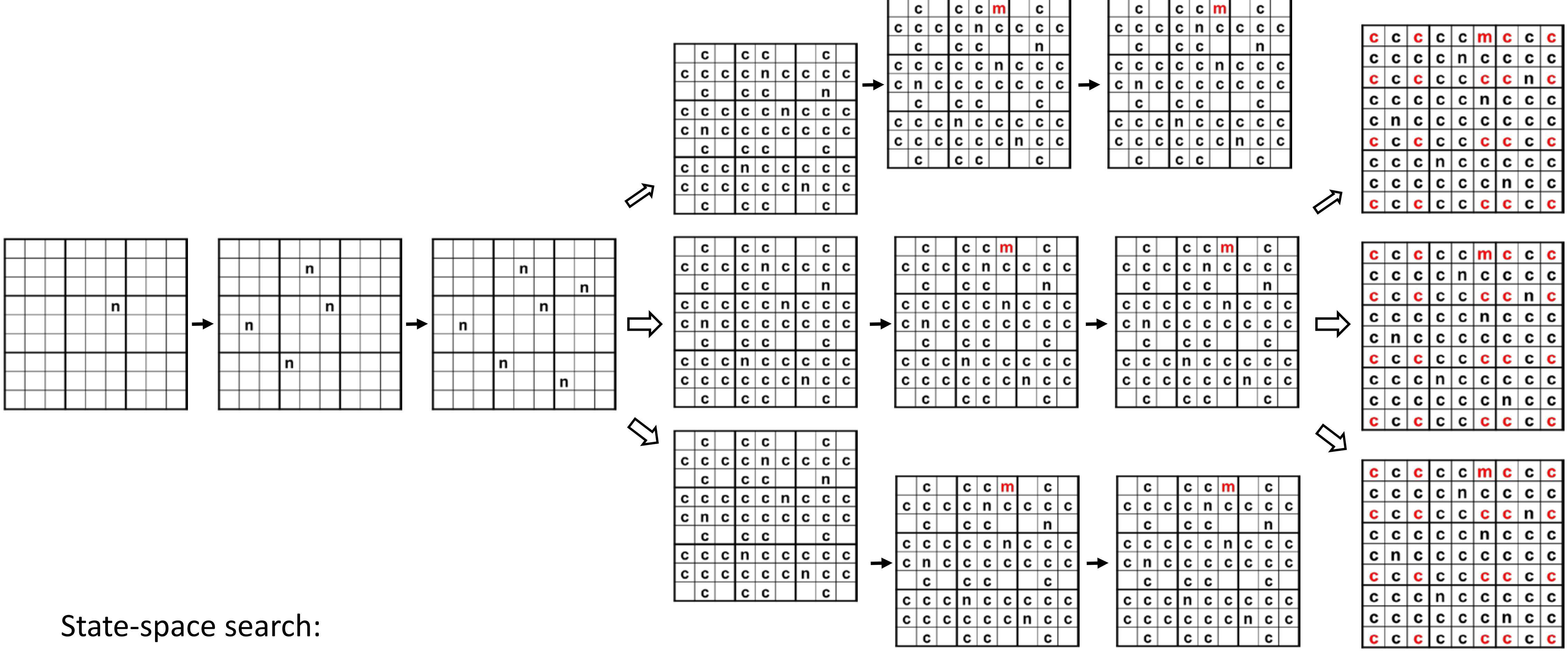




- Given values create
  variables and constraints.
- E.g., related to rows or columns or
  in a region around the given number.
- The constraint model is generated,
- solved and
- each solution triggers new requirements(!)



- Give futher requirements (m)
- Create further variables and constraints



State-space search:

- Time span for collecting data/detections/numbers
- Define partial (!) constraint problems with collected data
- Solving provides number of solutions
- Each solution is a further state
- Include new data/detections into the successor states, i.e., into the new constraint problems, as requirements
- Evaluate states through confidence values of detections
- Select number of successors according to resources (e.g. with Beam Search)

- Give futher requirements (m)
- Create further variables and constraints

Characteristics of Iterative Constraint Reasoning

- A (configuration or constraint) problem is **too large** for solving in one step.
- The requirements are **unknown in the beginning**.
- A „Task giver“ provides requirements in **discrete time steps**.
- A time span with a certain size defines a **partial constraint problem**.
- Globally solving it delivers **multiple solutions**.
- Each solution is a **start of a next problem** (“let’s take the assumption that the solution holds”).
- A **state space search** organizes partial constraint solving.
- Constraints **between time steps** can be included (variables might not change, make hypotheses about the future).
- **Evaluation of states** through confidence values.

Parameters of Iterative Sudoku

- **Size** of field
- Time span: How long should one **wait** before solving a problem?
- Which constraints shall be **generated**?
- How **many** solutions and which should be followed?
- What are **time dependent** variables?
- Handling **resources** (e.g., computation time)

Summary

- Iterative Sudoku as example for combining iterative constraint reasoning in a state space search
- Useable for hybrid AI architectures
- Notion of iterative constraint reasoning
- Template for a configuration process
- In the paper you find:
  - Definitions for iterative constraint reasoning
  - Comparision to other constraint approaches
- Next steps: experiments

Thank you for listening!

Paper: https://dl.acm.org/doi/abs/10.1145/3503229.3547051

Contact: lothar.hotz@uni-hamburg.de