Constraint Programming
Doctoral class

Hadrien Cambazard

(ROSP, G-SCOP)

Constraint Programming

- A framework to model and solve combinatorial optimization problems

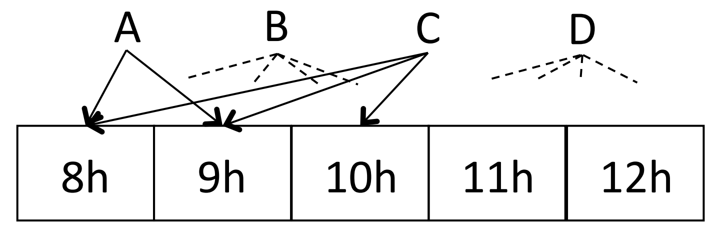
Constraint Programming

- A framework to model and solve combinatorial optimization problems

- A toy timetabling problem :
  a. **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  b. **A must end at 10h** and **C must end at 11h at the latest**
  c. **C must take place before B**
  d. **B must take place before D** with **one or two hours of break** inbetween

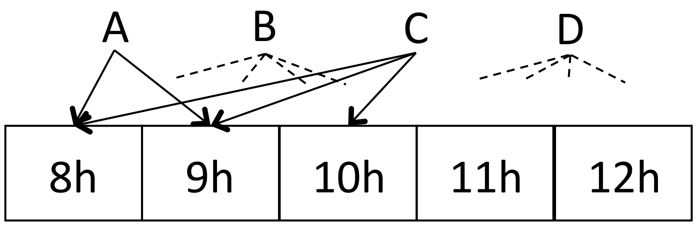
Constraint Programming

- A toy timetabling problem :
  - a. **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  - b. **A must end at 10h** and **C must end at 11h at the latest**
  - c. **C must take place before B**
  - d. **B must take place before D** with **one or two hours of break** inbetween



Constraint Programming

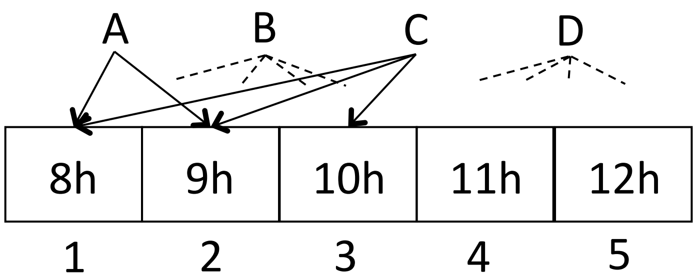
- A toy timetabling problem :
  a. **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  b. **A must end at 10h** and **C must end at 11h at the latest**
  c. **C must take place before B**
  d. **B must take place before D** with **one or two hours of break** inbetween



- A problem is modelled with **variables**, **domains**, and **constraints**

Constraint Programming

- A toy timetabling problem :
  a. **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  b. **A must end at 10h** and **C must end at 11h at the latest**
  c. **C must take place before B**
  d. **B must take place before D** with **one or two hours of break** inbetween

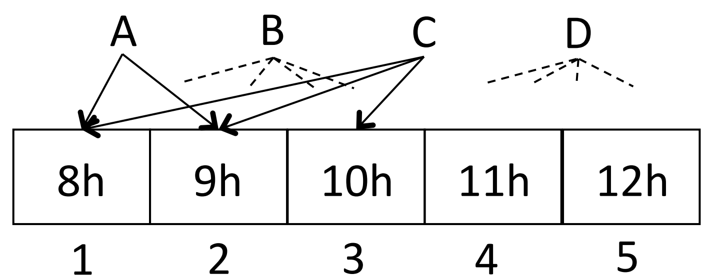


- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1, 2\}$$
$$x_C \in \{1, 2, 3\}$$
$$x_B, x_D \in \{1, 2, 3, 4, 5\}$$

Constraint Programming

- A toy timetabling problem :
  a. **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  b. **A must end at 10h** and **C must end at 11h at the latest**
  c. **C must take place before B**
  d. **B must take place before D** with **one or two hours of break** inbetween

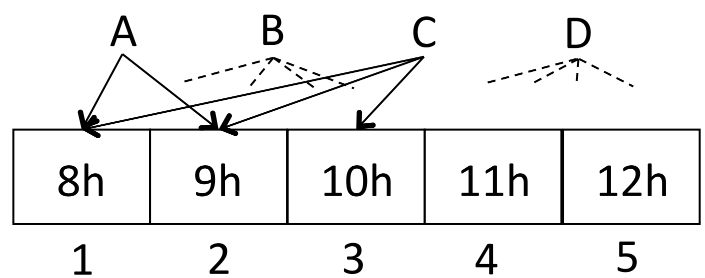


- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\} \quad x_C < x_B$$

Constraint Programming

- A toy timetabling problem :
  - a. **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  - b. **A must end at 10h** and **C must end at 11h at the latest**
  - c. **C must take place before B**
  - d. **B must take place before D** with **one or two hours of break** inbetween

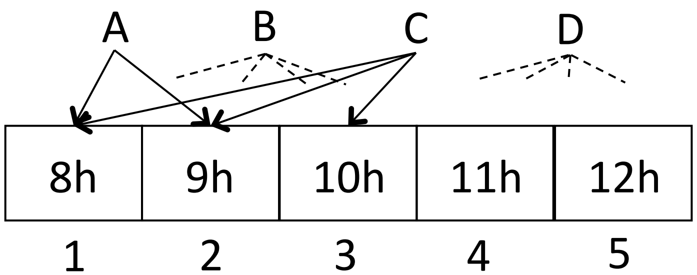


- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\} \qquad allDifferent(x_A, x_B, x_C, x_D)$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\} \quad x_C < x_B$$

Constraint Programming

- A toy timetabling problem :
  - **a.** **4** meetings A, B, C, D **of one hour** take place **in the same room** between 8h and 13h
  - **b.** **A must end at 10h** and **C must end at 11h at the latest**
  - **c.** **C must take place before B**
  - **d.** **B must take place before D** with **one or two hours of break** inbetween



- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

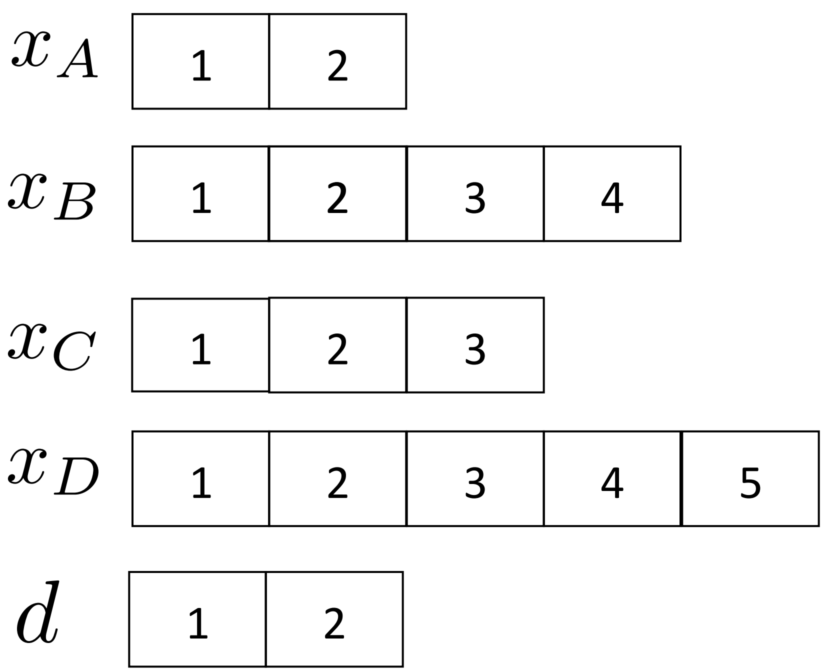
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



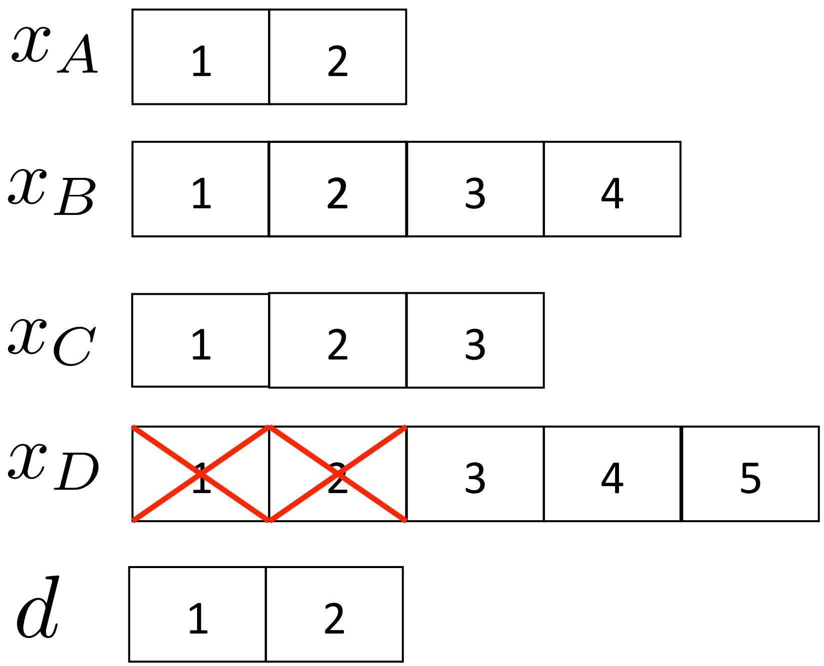
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



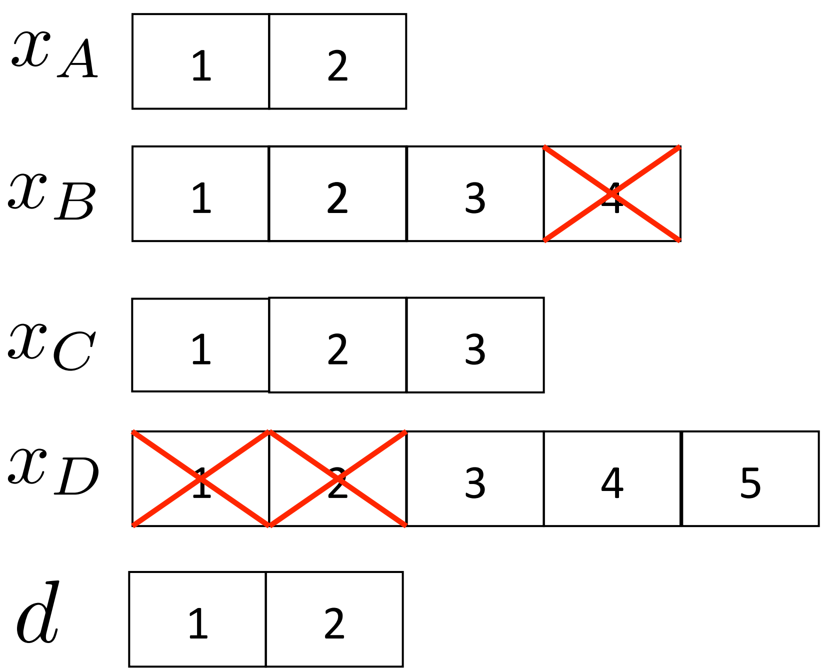
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



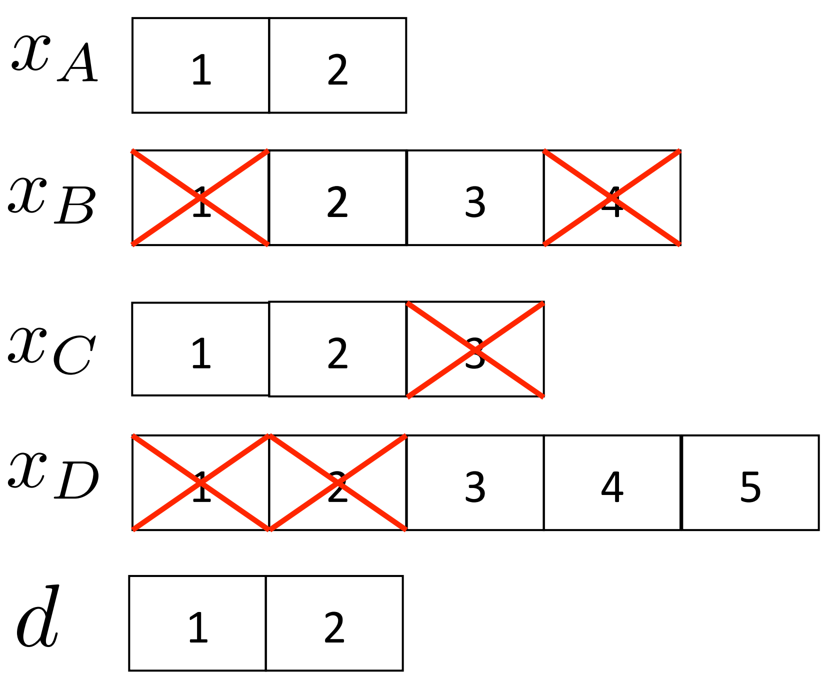
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



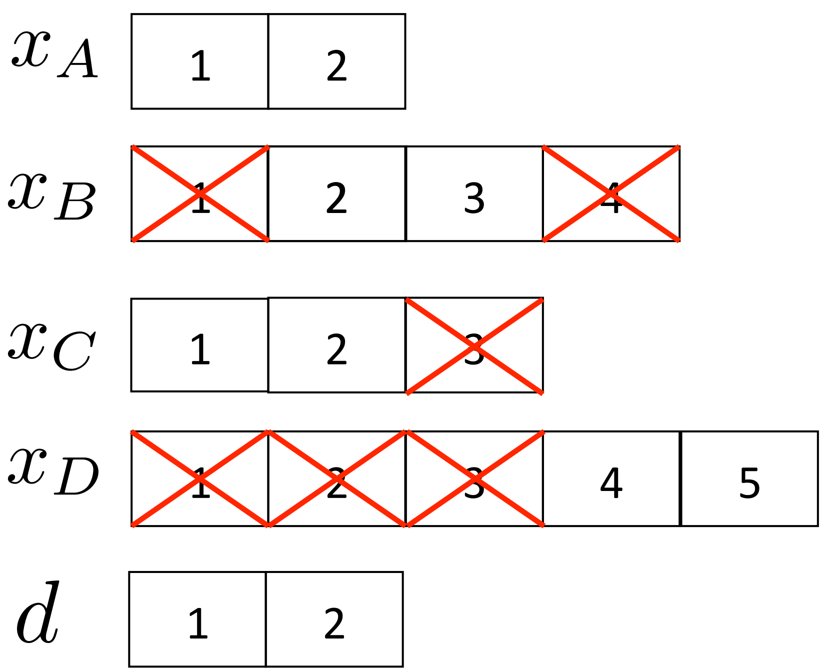
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



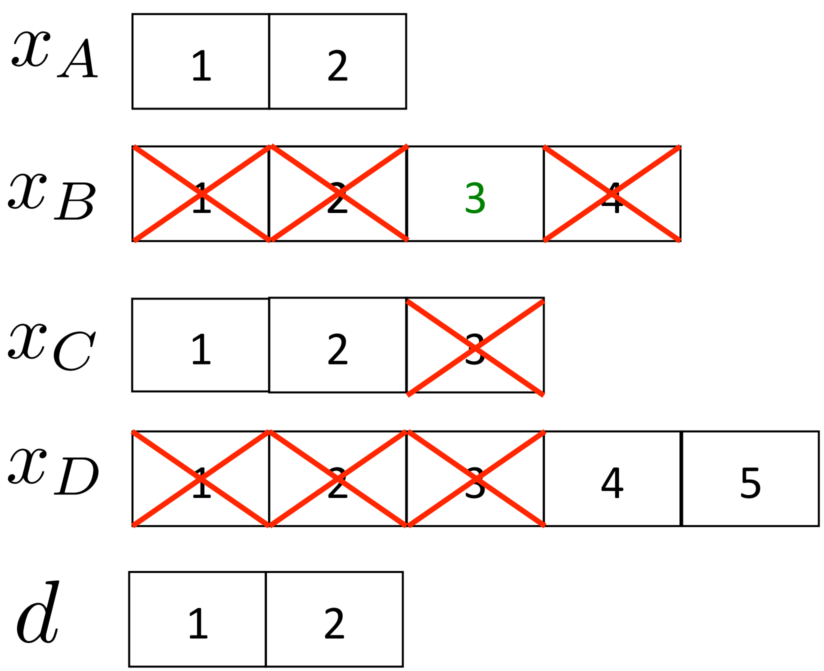
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1, 2\}$$
$$x_C \in \{1, 2, 3\}$$
$$x_B, x_D \in \{1, 2, 3, 4, 5\}$$
$$d \in \{1, 2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



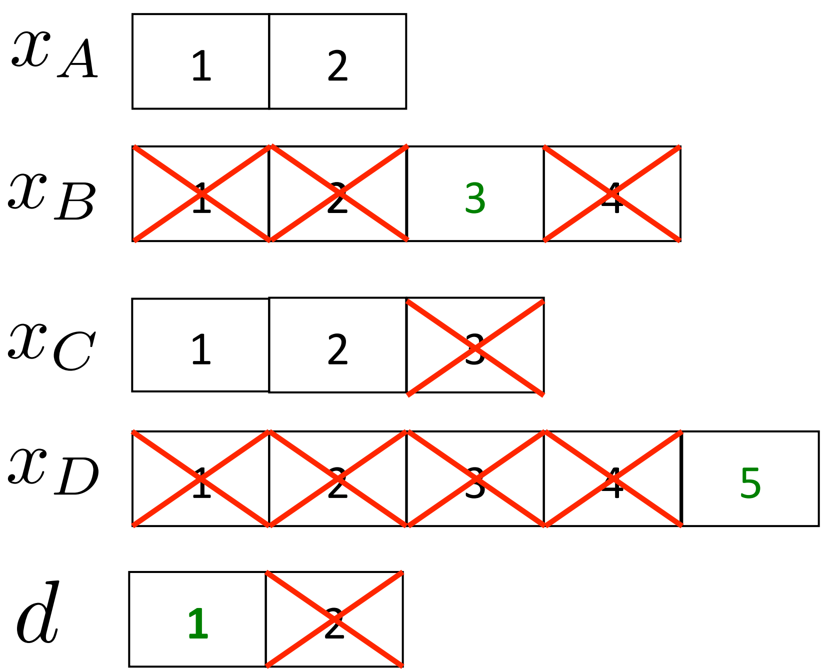
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

$$x_A \in \{1, 2\}$$
$$x_C \in \{1, 2, 3\}$$
$$x_B, x_D \in \{1, 2, 3, 4, 5\}$$
$$d \in \{1, 2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



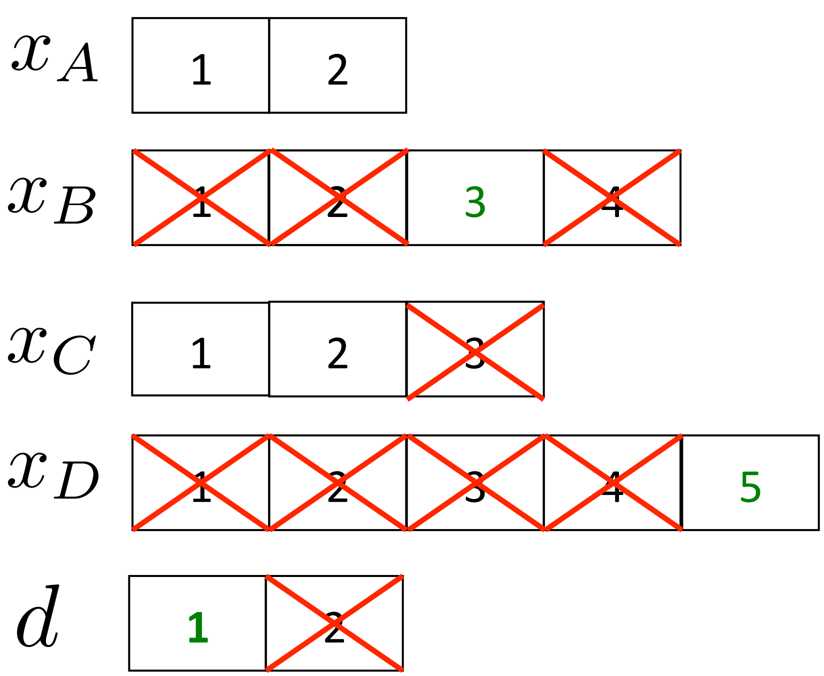
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

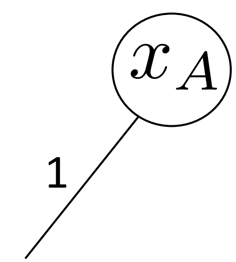
$$x_A \in \{1,2\}$$
$$x_C \in \{1,2,3\}$$
$$x_B, x_D \in \{1,2,3,4,5\}$$
$$d \in \{1,2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

Propagation



Branching (Tree Search)



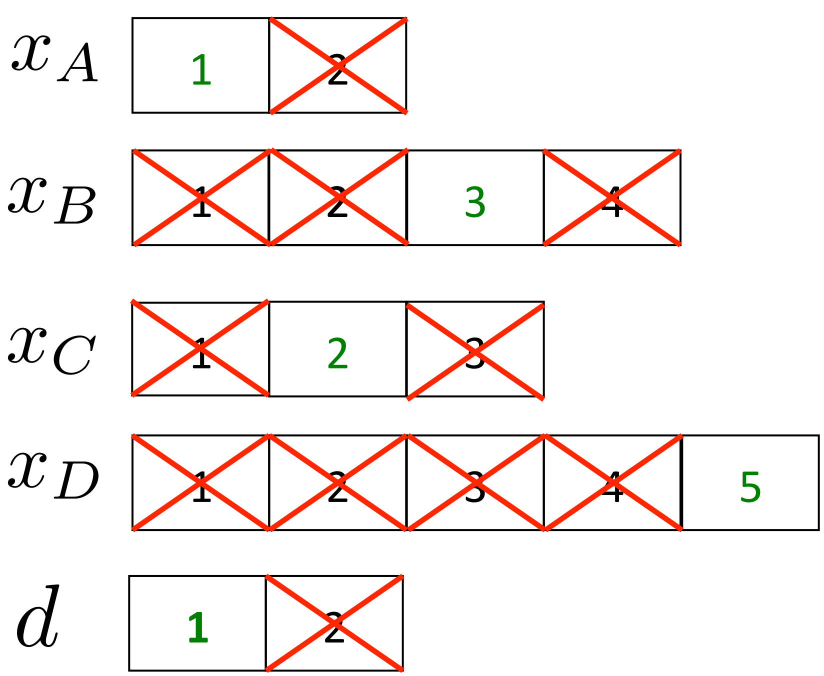
Constraint Programming

- A problem is modelled with **variables**, **domains**, and **constraints**

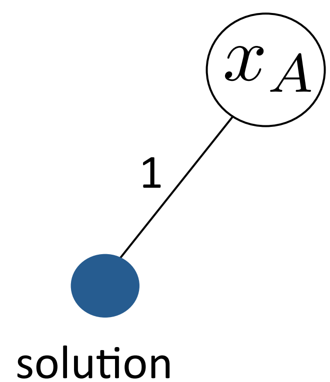
$$x_A \in \{1, 2\}$$
$$x_C \in \{1, 2, 3\}$$
$$x_B, x_D \in \{1, 2, 3, 4, 5\}$$
$$d \in \{1, 2\}$$

$$allDifferent(x_A, x_B, x_C, x_D)$$
$$x_B + d + 1 = x_D$$
$$x_C < x_B$$

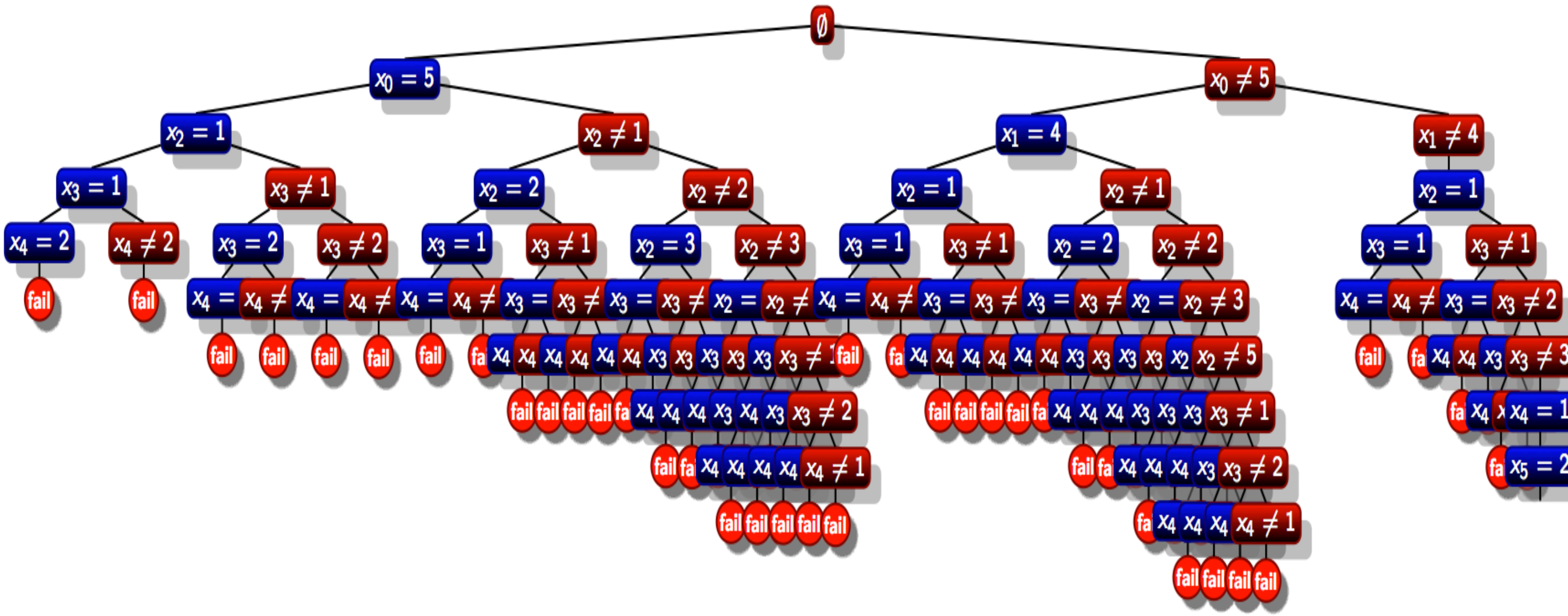
Propagation



Branching (Tree Search)



Branching is usually much bigger



Constraint Programming

- *A constraint* :
  1. A **logical relationship** that values of the variables must satisfy in a solution
  2. A **reasoning block** for a **frequent** sub-problem that we can solve efficiently (filtering algorithm)
  3. A reusable **modelling block**
  4. (Almost) no restriction on the constraints :
     - **Checkable in polynomial time**
     - Often a known relation for which an efficient algorithm exists

- *Filtering* : Detect inconsistent variable/value pairs and remove them

- *Constraint Propagation*:
  - **Exchange of information** (trough domains) between constraints (➡ fix point)
  - The inference method to solve the **composite** problem.
  - Used **before each decision** of the branching algorithm

- *Tree search*:
  - Take **decisions** (reductions of the problem) until a solution is found or the proof that none exists is completed (**backtrack** search).

Restrictions of the problem formulation

MILP (Mixed Integer Linear Programming)

- linear inequalities
- Real/integer variables

$$\text{Maximize} \quad P = p_1 x_1 + p_2 x_2 + \cdots + p_k x_k$$

$$\begin{aligned} \text{Subject to:} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1k}x_k \leq q_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2k}x_k \leq q_2 \\ & \vdots \\ & a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nk}x_k \leq q_n \\ & x_1, x_2, \cdots x_k \geq 0 \end{aligned}$$

SAT (Boolean satisfiability)

- Clauses (conjonctive normal form)
- Boolean variables

$$\Phi = C_1 \wedge \ldots \wedge C_8$$

$$\begin{aligned} C_1 &= (x_1 \vee x_2 \vee \neg x_3) \\ C_2 &= (\neg x_1 \vee \neg x_2 \vee \neg x_4) \\ C_3 &= (x_1 \vee \neg x_2 \vee \neg x_5) \\ C_4 &= (\neg x_1 \vee x_3 \vee \neg x_4) \\ C_5 &= (x_1 \vee \neg x_3 \vee x_5) \\ C_6 &= (x_1 \vee \neg x_4 \vee x_5) \\ C_7 &= (x_2 \vee x_4 \vee x_5) \\ C_8 &= (\neg x_3 \vee x_4 \vee \neg x_5) \end{aligned}$$

CP (Constraint Programming)

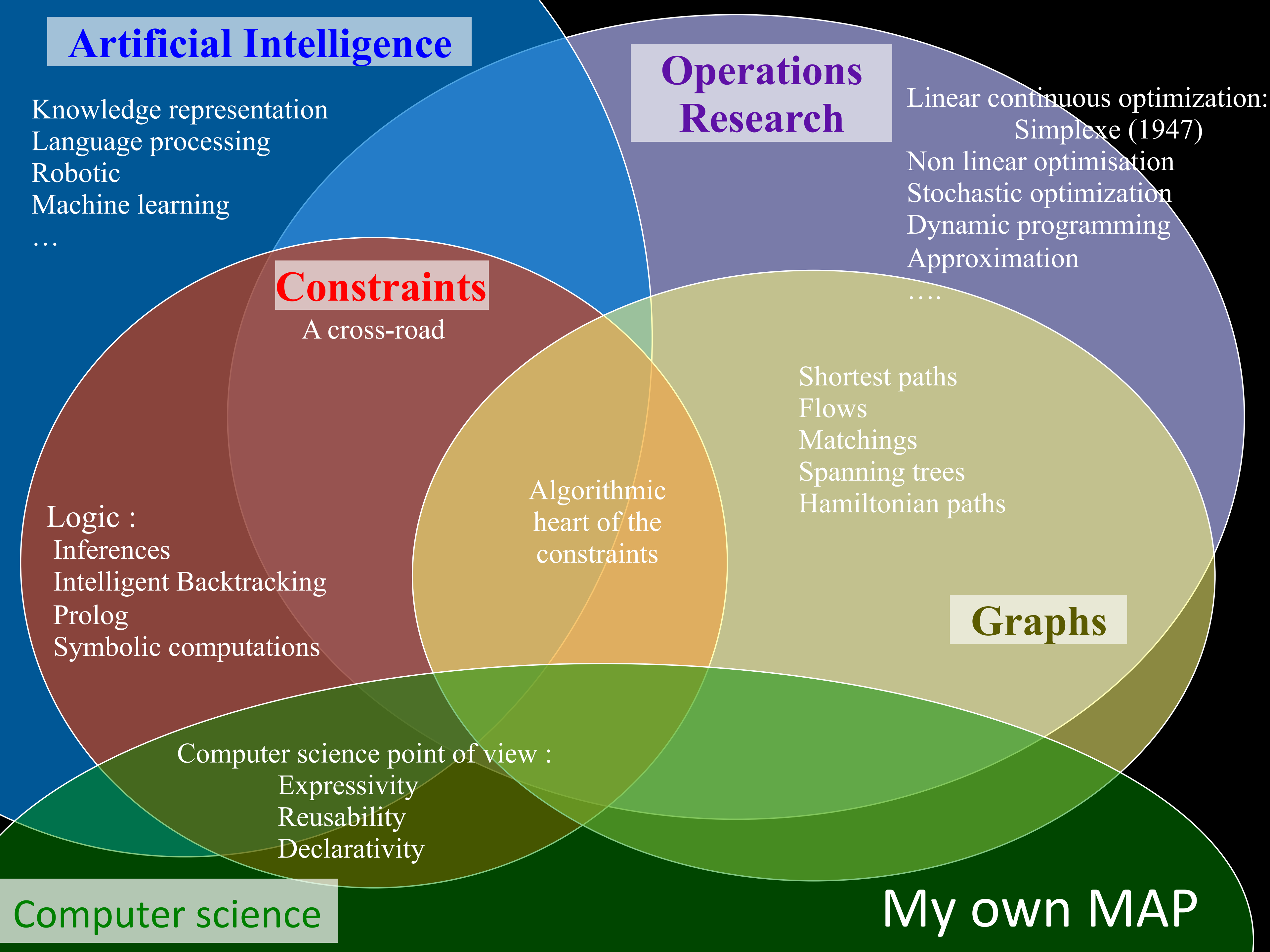
$$\begin{aligned} & \text{AllDifferent}(x_A, x_B, x_C, x_D) \\ & x_B + d + 1 = x_D, \\ & x_C < x_B, \\ & x_A \in \{1,2\}, x_C \in \{1,2,3\}, x_B, x_D \in \{1,2,3,4,5\}, d \in \{1,2\} \end{aligned}$$

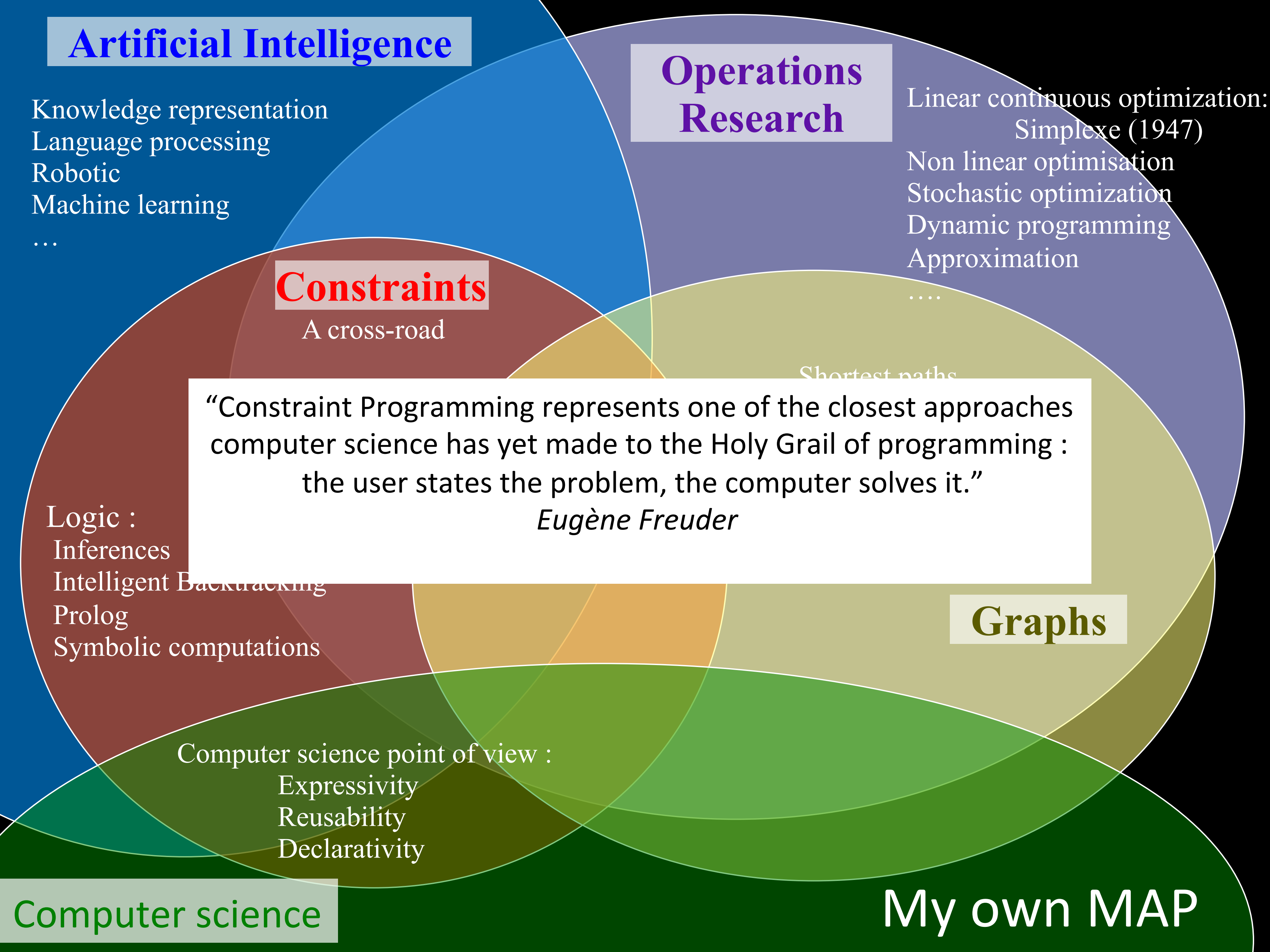
Constraint Programming (short history)

- **Language roots**: logic programming (**declarativity**)
  - Planner (Hewitt, 1965) and later Prolog (Colmerauer, 1972)
  - Developed for *language processing* and then used in *Theorem proving*, …, *Planning*.
- **Algorithmic roots:**
  - Machine vision community (“a man-machine graphical communication systems”, 1963)
  - Image processing (“Waltz Filtering”, 1975) (Application: “scene labeling”)

- Constraint Logic Programming: **CLP**
  - Concerns: **Representation** and **Reasoning**
  - Methodology: **Search** (very AI) but also **Inference**
  - **CHIP** (Van Hentenryck, Constraints Handling In Prolog, 1989)

- Constraint Programming: **CP** (strengthened relationship to OR)
  - 1990: birth of a workshop : “*Principles and Practice of Constraint Programming*” (conf in 1995)
  - Global constraints: (Beldiceanu et al, *Extending chip in order to solve complex scheduling and placement problems*, 1992)
  - 1999: birth of a second workshop: “Constraint Programming, Artificial Intelligence and Operations Research” (conf in 2004)


Artificial Intelligence
Knowledge representation
Language processing
Robotic
Machine learning
…
Operations Research
Linear continuous optimization:
Simplexe (1947)
Non linear optimisation
Stochastic optimization
Dynamic programming
Approximation
….
Constraints
A cross-road
Shortest paths
Flows
Matchings
Spanning trees
Hamiltonian paths
Algorithmic
heart of the
constraints
Logic :
Inferences
Intelligent Backtracking
Prolog
Symbolic computations
Graphs
Computer science point of view :
Expressivity
Reusability
Declarativity
Computer science
My own MAP


Artificial Intelligence
Knowledge representation
Language processing
Robotic
Machine learning
…
Operations Research
Linear continuous optimization:
Simplexe (1947)
Non linear optimisation
Stochastic optimization
Dynamic programming
Approximation
….
Constraints
A cross-road
“Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming : the user states the problem, the computer solves it.”
Eugène Freuder
Logic :
Inferences
Intelligent Backtracking
Prolog
Symbolic computations
Graphs
Computer science point of view :
Expressivity
Reusability
Declarativity
Computer science
My own MAP

Outline