Escuela AI HUB de verano
Sobre IA y Educación

Symbolic AI

Felip Manyà, IIIA-CSIC
Jordi Coll, UdG

JULY 2ND, 2024

Outline

Outline

AI and Problem Solving

- One of the first objectives of AI was to **simulate human intelligence**.
- **Reasoning** and **problem solving** were one of the first challenges faced.
- Reasoning: **logic and automated theorem proving.**
- Problem solving as **searching in a space of states** (puzzles, chess, . . .)
- **Combinatorial explosion**: First winter (toy examples).

Symbolic problem solving operates on the principle of representing knowledge and reasoning through symbols and logical rules. By manipulating abstract symbols, machines gain the ability to analyze relationships, infer new information, and arrive at logical conclusions.

Two Problem Solving Approaches

Specialized

Combinatorial Problem
↓
Algorithm
↓
Solution

Generic

Combinatorial Problem
↓
Reduction to another Problem P
↓
Solver for P
↓
Solution

P is, for instance, MIP, SAT, CSP, SMT, MaxSAT, . . .

Model & Solve Paradigm

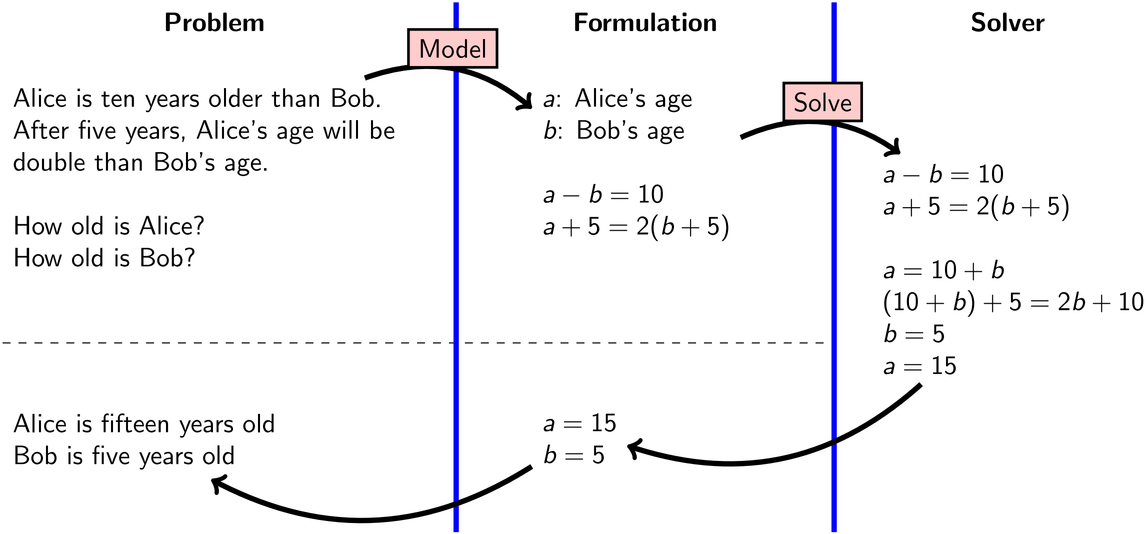
- Input problem P
- Encoding (MODEL)
- Feed encoding to solver (SOLVE)
- Generate solution for P

Specialized versus Generic

- In the generic approach, emphasis is on encoding.
- In the specialized approach, the human cost is bigger.
- In the generic approach, encodings can compensate for much of the loss due to going to a uniform representation formalism like SAT, CSP or MIP.
- In the specialized approach, it is difficult to duplicate efforts put in developing fast solvers for SAT, CSP or MIP.

Outline

Model and solve



Problem

Alice is ten years older than Bob.
After five years, Alice's age will be double than Bob's age.

How old is Alice?
How old is Bob?

Model

Formulation

a: Alice's age
b: Bob's age

$a - b = 10$
$a + 5 = 2(b + 5)$

Solve

Solver

$a - b = 10$
$a + 5 = 2(b + 5)$

$a = 10 + b$
$(10 + b) + 5 = 2b + 10$
$b = 5$
$a = 15$

$a = 15$
$b = 5$

Alice is fifteen years old
Bob is five years old

Constraint Satisfaction Problem (CSP)

- There exist many symbolic formalism (modelling languages) to formulate problems.

- Every symbolic formalism has **its own nomenclature**.

- Constraint Satisfaction Problems (CSP) capture most of them.

- A CSP is a tuple (X, D, C):
    - $X = \{x_1, \ldots, x_n\}$ is a set of variables.
    - $D = \{D_1, \ldots, D_n\}$ is the domain of each variable.
    - $C = \{c_1, \ldots, c_m\}$ is a set of constraints.

- The **solver** must find an assignment $I(x_i)$ to every variable x_i such that:
    - The value given to a variable is in its domain: $I(x_i) \in D_i$.
    - All constraints must be satisfied by I.

Constraint Satisfaction Problem (CSP)

- There exist many symbolic formalism (modelling languages) to formulate problems.

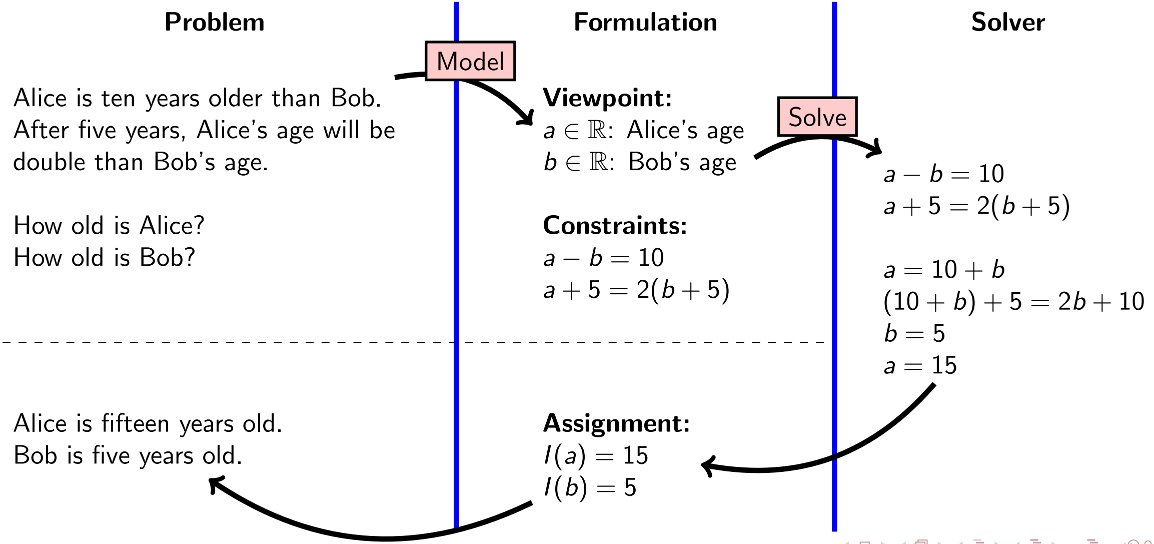
- Every symbolic formalism has **its own nomenclature**.

- Constraint Satisfaction Problems (CSP) capture most of them.

- A CSP is a tuple (X, D, C):
  - $X = \{x_1, \ldots, x_n\}$ is a set of variables.
  - $D = \{D_1, \ldots, D_n\}$ is the domain of each variable.
  - $C = \{c_1, \ldots, c_m\}$ is a set of constraints.

- The **modeller** must define:
  - **Viewpoint**: what **variables** there are, what are their **domains**, and what are their **meaning**.
  - **Constraints**: what properties must satisfy an assignment.

Model and solve



Problem

Alice is ten years older than Bob. After five years, Alice's age will be double than Bob's age.

How old is Alice?
How old is Bob?

Alice is fifteen years old.
Bob is five years old.

Formulation

Model

Viewpoint:
$a \in \mathbb{R}$: Alice's age
$b \in \mathbb{R}$: Bob's age

Constraints:
$a - b = 10$
$a + 5 = 2(b + 5)$

Assignment:
$I(a) = 15$
$I(b) = 5$

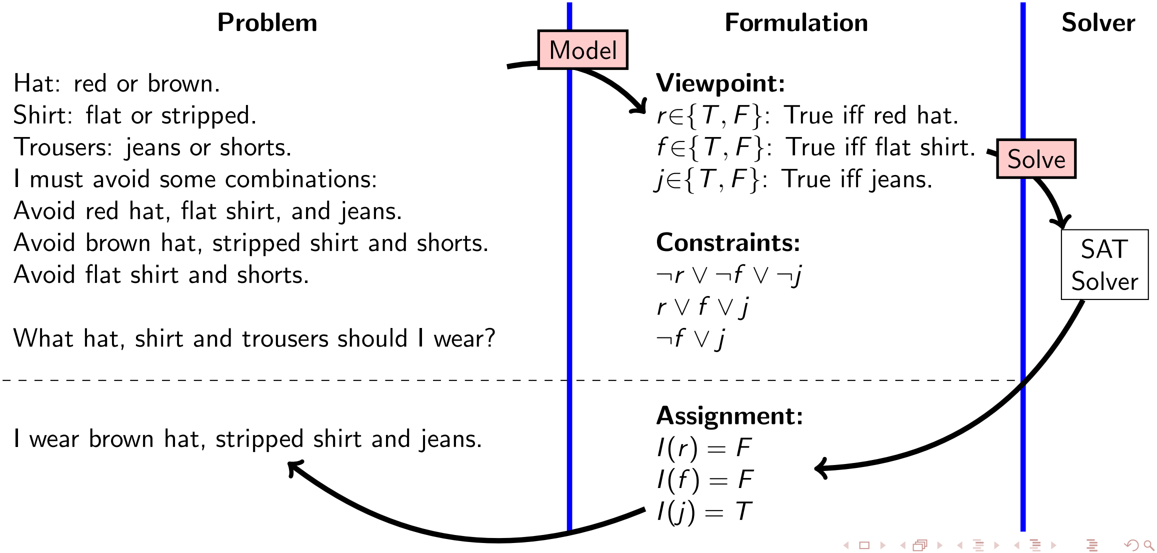
Solver

Solve

$a - b = 10$
$a + 5 = 2(b + 5)$

$a = 10 + b$
$(10 + b) + 5 = 2b + 10$
$b = 5$
$a = 15$

Model and solve: Boolean Satisfiability (SAT)



Problem

Formulation

Solver

Model

Hat: red or brown.
Shirt: flat or stripped.
Trousers: jeans or shorts.
I must avoid some combinations:
Avoid red hat, flat shirt, and jeans.
Avoid brown hat, stripped shirt and shorts.
Avoid flat shirt and shorts.

What hat, shirt and trousers should I wear?

Viewpoint:

$r \in \{T, F\}$: True iff red hat.
$f \in \{T, F\}$: True iff flat shirt.
$j \in \{T, F\}$: True iff jeans.

Solve

SAT Solver

Constraints:

$\neg r \vee \neg f \vee \neg j$
$r \vee f \vee j$
$\neg f \vee j$

I wear brown hat, stripped shirt and jeans.

Assignment:

$I(r) = F$
$I(f) = F$
$I(j) = T$

Outline

Existing symbolic formalisms

Generic:

- Boolean Satisfiability (SAT) and MaxSAT
- Satisfiabilit Modulo Theories (SMT)
- Constraint Programming (CP)
- Mixed Integer Linear Programming (MILP)
- Pseudo-Boolean Constraints (PB)
- ...

Domain-specific:

- Planning Domain Definition Language (PDDL)
- ...

Boolean Satisfiability (SAT) and MaxSAT

Variables:

- Boolean: value *True* or *False*

Constraints:

- Propositional logic, usually in Conjunctive Normal Form.

Example:

$$(x_1 \vee x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3 \vee x_4)$$

MaxSAT: maximize the number of satisfied clauses. Used for optimization problems.

Satisfiability Modulo Theories (SMT)

Variables:

- Different domains: Boolean, integer, bitvectors, ...

Constraints:

- Propositional logic modulo formal theories.

Example:

$$(a > b \lor x) \land (f(c) < f(d)) \land (i - j \leq 3 \lor i - k \leq 3)$$

Constraint Programming (CP)

Variables:

- Finite domain, e.g. $\{0, 1, \ldots, 10\}$.

Constraints:

- There are many languages: Minizinc, Essence', . . .
- Logic, arithmetic, global constraints

Example:

$$allDifferent(a, b, c, d, e)$$

Mixed Integer Linear Programming (MILP)

Variables:

- Real and integer. If all variables are $\{0, 1\}$, we have Pseudo-Boolean (PB).

Constraints:

- System of linear inequalities.

Objective function.

Example:

Maximize:

$$2x + 3y + 4z$$

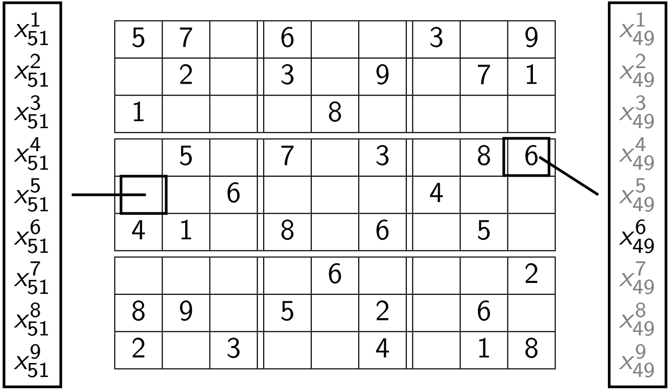
Subject to:

$$x + y \leq 2$$

$$2y + 3z \leq 20$$

Outline

Encoding Sudoku into SAT: viewpoint



x_{51}^{1}
x_{51}^{2}
x_{51}^{3}
x_{51}^{4}
x_{51}^{5}
x_{51}^{6}
x_{51}^{7}
x_{51}^{8}
x_{51}^{9}

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

x_{49}^{1}
x_{49}^{2}
x_{49}^{3}
x_{49}^{4}
x_{49}^{5}
x_{49}^{6}
x_{49}^{7}
x_{49}^{8}
x_{49}^{9}

Viewpoint: $9^3 = 729$ Boolean variables x_{ij}^{k}, with $i, j, k \in 1..9$:

$x_{ij}^{k} = \mathit{True}$ *means "at row* i*, column* j*, the value is* k*"*

Encoding Sudoku into SAT: constraints

1. Some number must be assigned to each cell.

For each cell located in row i and column j, we add the clause:

$$x_{ij}^1 \lor x_{ij}^2 \lor \cdots \lor x_{ij}^9$$

Encoding Sudoku into SAT: constraints

1. Some number must be assigned to each cell.
2. No number is repeated in the same row.

For each number k and for each row i, we add the clauses:

$$\begin{array}{lllll}
\neg x_{i1}^{k} \lor \neg x_{i2}^{k} & & & & \\
\neg x_{i1}^{k} \lor \neg x_{i3}^{k} & \neg x_{i2}^{k} \lor \neg x_{i3}^{k} & & & \\
\neg x_{i1}^{k} \lor \neg x_{i4}^{k} & \neg x_{i2}^{k} \lor \neg x_{i4}^{k} & \neg x_{i3}^{k} \lor \neg x_{i4}^{k} & & \\
\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots & & \\
\neg x_{i1}^{k} \lor \neg x_{i9}^{k} & \neg x_{i2}^{k} \lor \neg x_{i9}^{k} & \neg x_{i3}^{k} \lor \neg x_{i9}^{k} & \ldots\ldots & \neg x_{i8}^{k} \lor \neg x_{i9}^{k}
\end{array}$$

Encoding Sudoku into SAT: constraints

1. Some number must be assigned to each cell.
2. No number is repeated in the same row.
3. No number is repeated in the same column.

For each number k and for each column j, we add the clauses:

$$
\begin{array}{lllll}
\neg x_{1j}^{k} \vee \neg x_{2j}^{k} & & & & \\
\neg x_{1j}^{k} \vee \neg x_{3j}^{k} & \neg x_{2j}^{k} \vee \neg x_{3j}^{k} & & & \\
\neg x_{1j}^{k} \vee \neg x_{4j}^{k} & \neg x_{2j}^{k} \vee \neg x_{4j}^{k} & \neg x_{3j}^{k} \vee \neg x_{4j}^{k} & & \\
\ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots & & \\
\neg x_{1j}^{k} \vee \neg x_{9j}^{k} & \neg x_{2j}^{k} \vee \neg x_{9j}^{k} & \neg x_{3j}^{k} \vee \neg x_{9j}^{k} & \ldots\ldots & \neg x_{8j}^{k} \vee \neg x_{9j}^{k}
\end{array}
$$

Encoding Sudoku into SAT: constraints

1. Some number must be assigned to each cell.
2. No number is repeated in the same row.
3. No number is repeated in the same column.
4. No number is repeated in the 3x3 square.

Encoding Sudoku into SAT: constraints

1. Some number must be assigned to each cell.
2. No number is repeated in the same row.
3. No number is repeated in the same column.
4. No number is repeated in the 3x3 square.
5. There are predefined values.

DIMACS Format for SAT

```
p   cnf  3   3
-1  -2   -3  0
1   2    3   0
-2  3    0
```

$$(\neg r \lor \neg f \lor \neg j) \land (r \lor f \lor j) \land (\neg f \lor j)$$

Sudoku example:

`https://iiia.csic.es/~jjene/sat/`

Pseudo-Boolean Encoding of the Knapsack Problem

Problem: Given a set of items, select a subset to fill a knapsack. We want to reach profit K.

Viewpoint: $p_i \in \{0,1\}$ is 1 if we put item i inside the knapsack.

$$Var = \{p_i \mid 1 \leq i \leq n\}$$

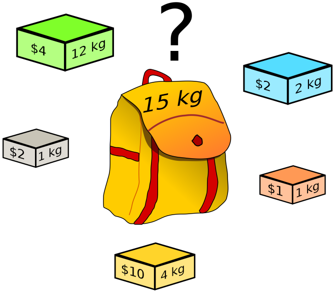
Constraints:

- Respect maximum load:

$$w_1 p_1 + w_2 p_2 + \cdots + w_n p_n \leq L$$

- Obtain required profit:

$$b_1 p_1 + b_2 p_2 + \cdots + b_n p_n \geq K$$



Constraint Programming for SEND+MORE=MONEY

Problem:

$$\begin{array}{ccccc} & \mathsf{S} & \mathsf{E} & \mathsf{N} & \mathsf{D} \\ + & \mathsf{M} & \mathsf{O} & \mathsf{R} & \mathsf{E} \\ \hline \mathsf{M} & \mathsf{O} & \mathsf{N} & \mathsf{E} & \mathsf{Y} \end{array}$$

Viewpoint: All variables in $\{S, E, N, D, M, O, R, Y\}$ take a value $\{0, 1, \ldots, 9\}$.

Constraints:

$$\texttt{allDifferent}(\{S, E, N, D, M, O, R, Y\})$$

$$\begin{aligned} 1000 * S + 100 * E + 10 * N + D \\ + 1000 * M + 100 * O + 10 * R + E \\ = 10000 * M + 1000 * O + 100 * N + 10 * E + Y \end{aligned}$$

Escuela AI HUB de verano
Sobre IA y Educación

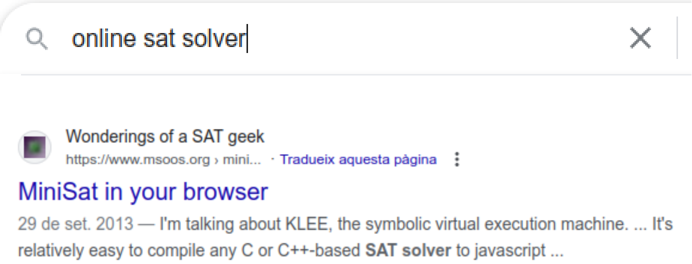
Experiment with Symbolic AI

Felip Manyà, IIIA-CSIC
Jordi Coll, UdG

JULY 2ND, 2024

CSIC Consejo Superior de Investigaciones Científicas
CSIC HUB Inteligencia Artificial
Fundación "la Caixa"
IFIC Institut de Física Corpuscular
Universitat de València

Try it yourselves!

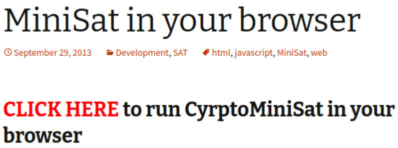


online sat solver

Wonderings of a SAT geek
https://www.msoos.org › mini... · Tradueix aquesta pàgina

MiniSat in your browser

29 de set. 2013 — I'm talking about KLEE, the symbolic virtual execution machine. ... It's relatively easy to compile any C or C++-based **SAT solver** to javascript ...

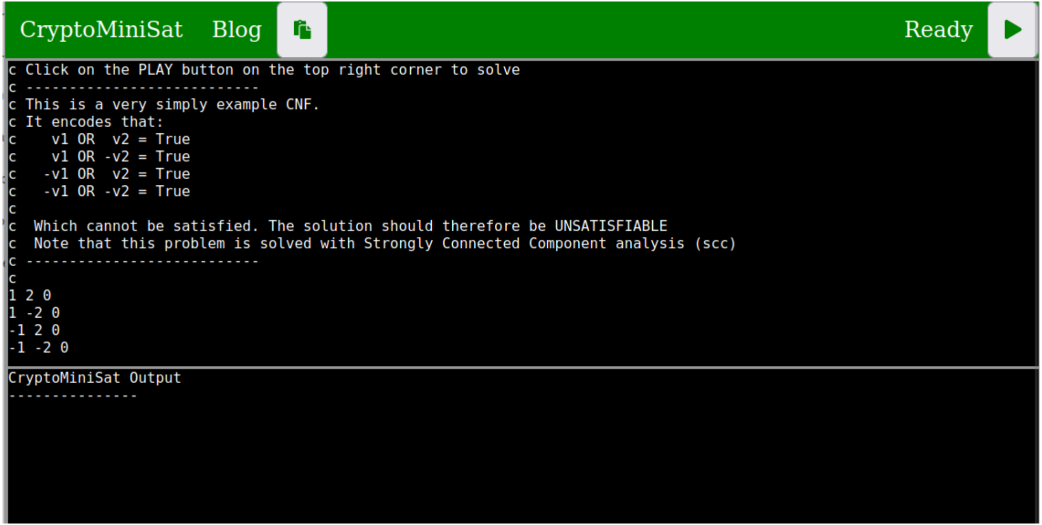


MiniSat in your browser

September 29, 2013 · Development, SAT · html, javascript, MiniSat, web

CLICK HERE to run CyrptoMiniSat in your browser

Try it yourselves!

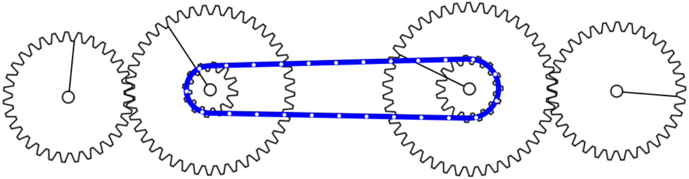


```
c Click on the PLAY button on the top right corner to solve
c ---------------------------
c This is a very simply example CNF.
c It encodes that:
c    v1 OR  v2 = True
c    v1 OR -v2 = True
c   -v1 OR  v2 = True
c   -v1 OR -v2 = True
c
c  Which cannot be satisfied. The solution should therefore be UNSATISFIABLE
c  Note that this problem is solved with Strongly Connected Component analysis (scc)
c ---------------------------
c
1 2 0
1 -2 0
-1 2 0
-1 -2 0
```

```
CryptoMiniSat Output
---------------
```

Gears: how do they rotate?

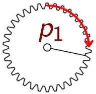
Problem: Given a set of gears joined either directly or by a chain, how will they rotate?



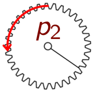
Gears: how do they rotate?

Problem: Given a set of gears joined either directly or by a chain, how will they rotate?

Viewpoint: p_i is *True* if and only if gear i rotates clockwise.



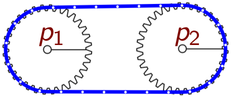
$p_1 = True$



$p_2 = False$

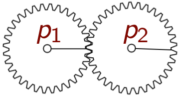
Constraints:

- Two gears connected by a chain rotate in the same direction.



$p_1 \leftrightarrow p_2$

- Two gears connected directly rotate in opposite directions.

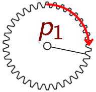


$\neg(p_1 \leftrightarrow p_2)$

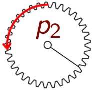
Gears: how do they rotate?

Problem: Given a set of gears joined either directly or by a chain, how will they rotate?

Viewpoint: p_i is *True* if and only if gear i rotates clockwise.



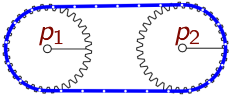
$p_1 = True$



$p_2 = False$

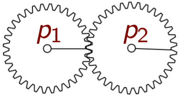
Constraints:

- Two gears connected by a chain rotate in the same direction.



$$p_1 \leftrightarrow p_2 \quad \equiv \quad (p_1 \vee \neg p_2) \wedge (\neg p_1 \vee p_2)$$

- Two gears connected directly rotate in opposite directions.

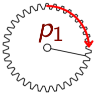


$$\neg(p_1 \leftrightarrow p_2) \quad \equiv \quad (p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2)$$

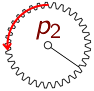
Gears: how do they rotate?

Problem: Given a set of gears joined either directly or by a chain, how will they rotate?

Viewpoint: p_i is *True* if and only if gear i rotates clockwise.



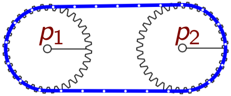
$p_1 = \mathit{True}$



$p_2 = \mathit{False}$

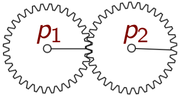
Constraints:

- Two gears connected by a chain rotate in the same direction.



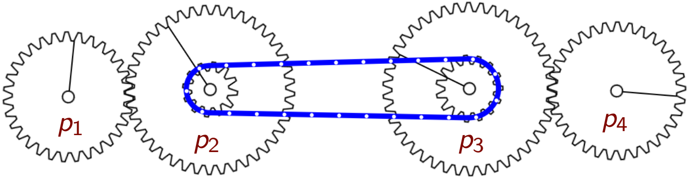
$$p_1 \leftrightarrow p_2 \quad \equiv \quad (p_1 \vee \neg p_2) \wedge (\neg p_1 \vee p_2) \quad \overset{\text{DIMACS}}{\equiv} \quad \begin{array}{rrr} 1 & -2 & 0 \\ -1 & 2 & 0 \end{array}$$

- Two gears connected directly rotate in opposite directions.

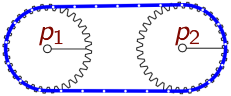


$$\neg(p_1 \leftrightarrow p_2) \quad \equiv \quad (p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2) \quad \overset{\text{DIMACS}}{\equiv} \quad \begin{array}{rrr} 1 & 2 & 0 \\ -1 & -2 & 0 \end{array}$$

Gears: how do they rotate?

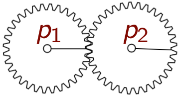


- Two gears connected by a chain rotate in the same direction.



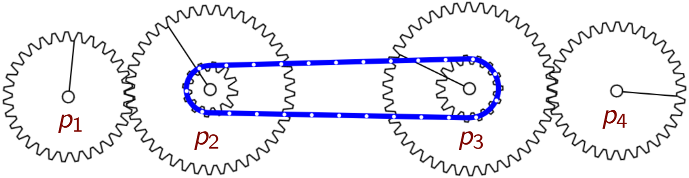
$$p_1 \leftrightarrow p_2 \quad \equiv \quad (p_1 \lor \neg p_2) \land (\neg p_1 \lor p_2) \quad \overset{\text{DIMACS}}{\equiv} \quad \begin{array}{rrr} 1 & -2 & 0 \\ -1 & 2 & 0 \end{array}$$

- Two gears connected directly rotate in opposite directions.

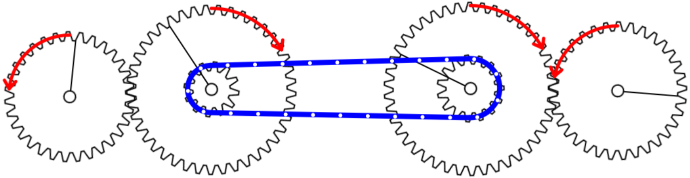


$$\neg(p_1 \leftrightarrow p_2) \quad \equiv \quad (p_1 \lor p_2) \land (\neg p_1 \lor \neg p_2) \quad \overset{\text{DIMACS}}{\equiv} \quad \begin{array}{rrr} 1 & 2 & 0 \\ -1 & -2 & 0 \end{array}$$

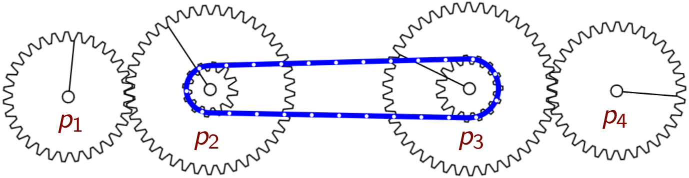
Gears: how do they rotate?



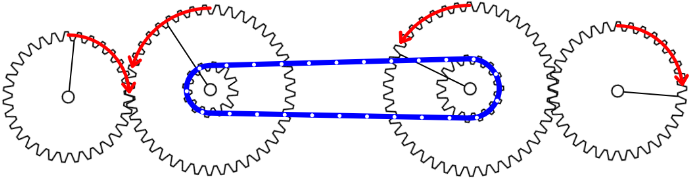
Solution: -1 2 3 -4 $\overset{\text{DIMACS}}{\equiv}$ $p_1 = \mathit{False},\ p_2 = \mathit{True},\ p_3 = \mathit{True},\ p_4 = \mathit{False}$



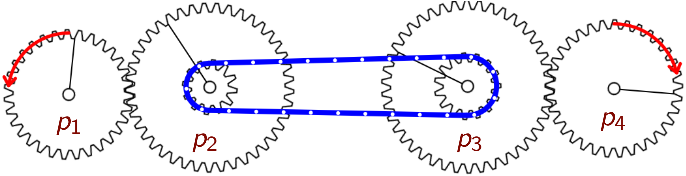
Gears: how do they rotate?



Solution 2: 1 -2 -3 4 $\overset{\text{DIMACS}}{\equiv}$ $p_1 = \mathit{True},\ p_2 = \mathit{False},\ p_3 = \mathit{False},\ p_4 = \mathit{True}$

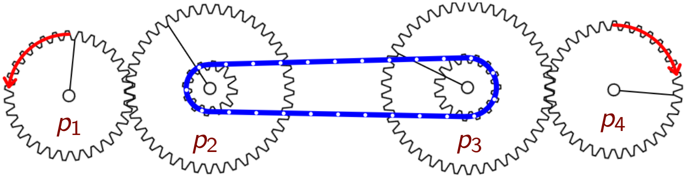


Gears: how do they rotate?



```
-1 0
4 0
```

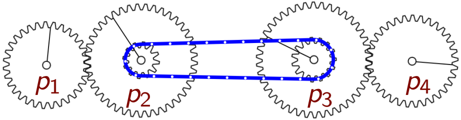
Gears: how do they rotate?



```
-1 0
4 0
```

Solution: **UNSATISFIABLE**

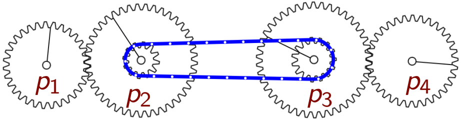
How do we automate this reasoning?



Truth table:

p_1	p_2	p_3	p_4	$\neg(p_1 \leftrightarrow p_2)$	$p_2 \leftrightarrow p_3$	$\neg(p_3 \leftrightarrow p_4)$	SAT?
0	0	0	0	0	1	0	NO
0	0	0	1	0	1	1	NO
0	0	1	0	0	0	0	NO
0	0	1	1	0	0	1	NO
0	1	0	0	1	0	1	NO
0	1	0	1	1	0	0	NO
0	1	1	0	1	1	1	YES
0	1	1	1	1	1	0	NO
...	...	...	...	...	...	...	...

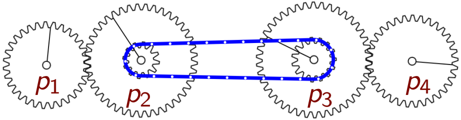
How do we automate this reasoning?



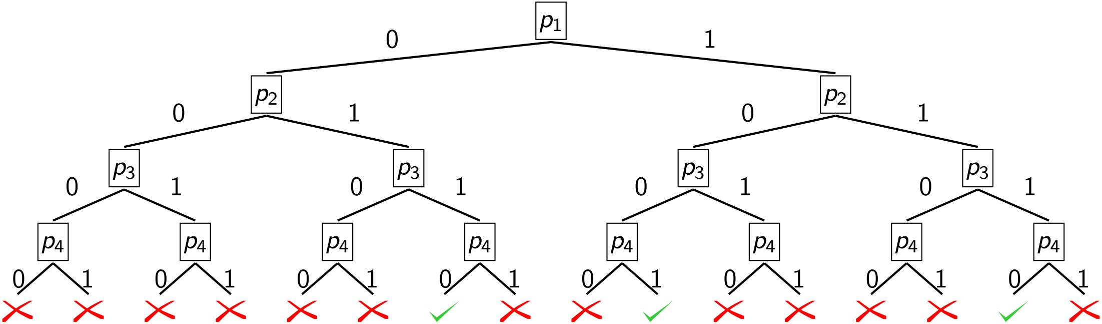
Truth table: Exponential size, 2^n, prohibitive!

p_1	p_2	p_3	p_4	$\neg(p_1 \leftrightarrow p_2)$	$p_2 \leftrightarrow p_3$	$\neg(p_3 \leftrightarrow p_4)$	SAT?
0	0	0	0	0	1	0	NO
0	0	0	1	0	1	1	NO
0	0	1	0	0	0	0	NO
0	0	1	1	0	0	1	NO
0	1	0	0	1	0	1	NO
0	1	0	1	1	0	0	NO
0	1	1	0	1	1	1	YES
0	1	1	1	1	1	0	NO
...	...	...	...	...	...	...	...

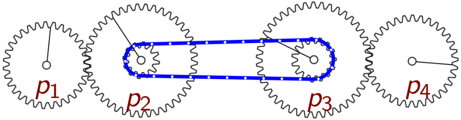
How do we automate this reasoning?



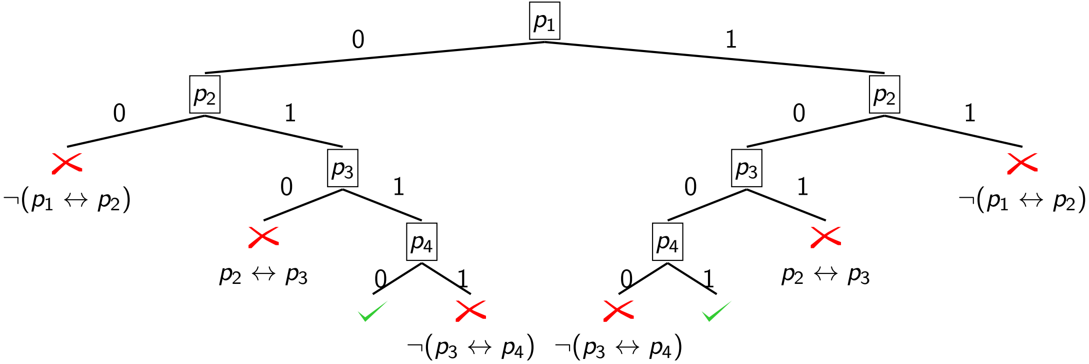
Search tree:



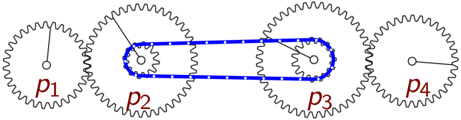
How do we automate this reasoning?



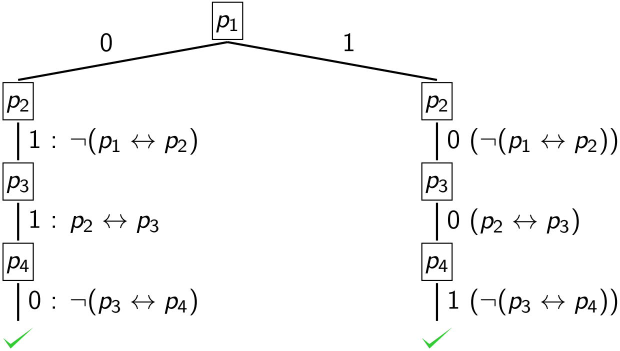
Search tree + pruning:



How do we automate this reasoning?



Search tree + pruning + propagation:



p_1

0

p_2

1 : $\neg(p_1 \leftrightarrow p_2)$

p_3

1 : $p_2 \leftrightarrow p_3$

p_4

0 : $\neg(p_3 \leftrightarrow p_4)$

✓

1

p_2

0 $(\neg(p_1 \leftrightarrow p_2))$

p_3

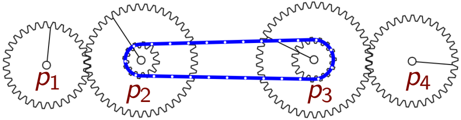
0 $(p_2 \leftrightarrow p_3)$

p_4

1 $(\neg(p_3 \leftrightarrow p_4))$

✓

How do we automate this reasoning?

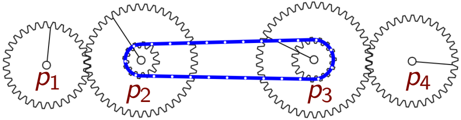


Inference with resolution rule:

$$\neg(p_1 \leftrightarrow p_2) \quad \equiv \quad (p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2)$$

$$p_2 \leftrightarrow p_3 \quad \equiv \quad (\neg p_2 \vee p_3) \wedge (p_2 \vee \neg p_3)$$

How do we automate this reasoning?



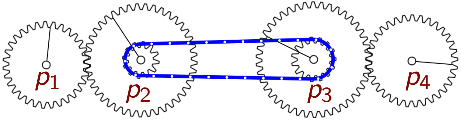
Inference with resolution rule:

$$\neg(p_1 \leftrightarrow p_2) \quad \equiv \quad (p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2)$$

$$p_2 \leftrightarrow p_3 \quad \equiv \quad (\neg p_2 \vee p_3) \wedge (p_2 \vee \neg p_3)$$

$$\frac{p_1 \vee p_2 \qquad \neg p_2 \vee p_3}{p_1 \vee p_3} \qquad\qquad \frac{\neg p_1 \vee \neg p_2 \qquad p_2 \vee \neg p_3}{\neg p_1 \vee \neg p_3}$$

How do we automate this reasoning?



Inference with resolution rule:

$$\neg(p_1 \leftrightarrow p_2) \quad \equiv \quad (p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2)$$

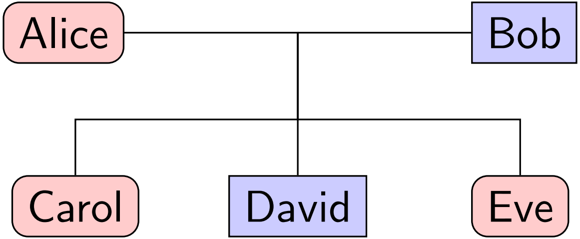
$$p_2 \leftrightarrow p_3 \quad \equiv \quad (\neg p_2 \vee p_3) \wedge (p_2 \vee \neg p_3)$$

$$\frac{p_1 \vee p_2 \qquad \neg p_2 \vee p_3}{p_1 \vee p_3} \qquad\qquad \frac{\neg p_1 \vee \neg p_2 \qquad p_2 \vee \neg p_3}{\neg p_1 \vee \neg p_3}$$

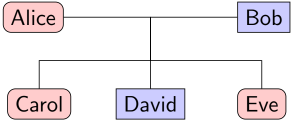
$$\neg(p_1 \leftrightarrow p_3) \quad \equiv \quad (p_1 \vee p_3) \wedge (\neg p_1 \vee \neg p_3)$$

Who is the impostor?

Who is the impostor?



Who is the impostor?



Viewpoint

a_1/b_2/c_3/d_4/e_5: Alice/Bob/Carol/David/Eve is the impostor

Constraints

At least one impostor: $a_1 \vee b_2 \vee c_3 \vee d_4 \vee e_5$

At most one impostor:

$\neg a_1 \vee \neg b_2$
$\neg a_1 \vee \neg c_3$
$\neg a_1 \vee \neg d_4$
$\neg a_1 \vee \neg e_5$

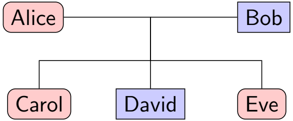
$\neg b_2 \vee \neg c_3$
$\neg b_2 \vee \neg d_4$
$\neg b_2 \vee \neg e_5$

$\neg c_3 \vee \neg d_4$
$\neg c_3 \vee \neg e_5$

$\neg d_4 \vee \neg e_5$

```
1 2 3 4 5 0
-1 -2 0
-1 -3 0
-1 -4 0
-1 -5 0
-2 -3 0
-2 -4 0
-2 -5 0
-3 -4 0
-3 -5 0
-4 -5 0
```

Who is the impostor?



Alice

Bob

Carol

David

Eve

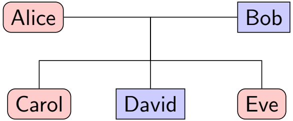
Viewpoint

$\mathbf{a_1}/\mathbf{b_2}/\mathbf{c_3}/\mathbf{d_4}/\mathbf{e_5}$: Alice/Bob/Carol/David/Eve is the impostor

Constraints

Exactly one impostor

Who is the impostor?



Viewpoint

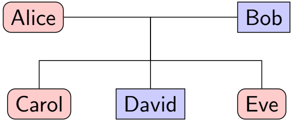
$\mathbf{a_1/b_2/c_3/d_4/e_5}$: Alice/Bob/Carol/David/Eve is the impostor

Constraints

Exactly one impostor

What if we add $a_1 \overset{\text{DIMACS}}{\equiv} 1\ \ 0$?

Who is the impostor?



Viewpoint

$\mathbf{a_1}$/$\mathbf{b_2}$/$\mathbf{c_3}$/$\mathbf{d_4}$/$\mathbf{e_5}$: Alice/Bob/Carol/David/Eve is the impostor

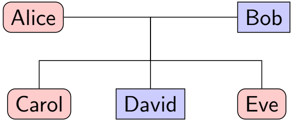
Constraints

Exactly one impostor

What if we add $a_1 \overset{\text{DIMACS}}{\equiv}$ `1 0`? Yes, Alice **can be** the impostor.

$$a_1 = \mathit{True}, b_2 = \mathit{False}, c_3 = \mathit{False}, d_4 = \mathit{False}, e_5 = \mathit{False}$$

Who is the impostor?



Viewpoint

$\mathbf{a_1}/\mathbf{b_2}/\mathbf{c_3}/\mathbf{d_4}/\mathbf{e_5}$: Alice/Bob/Carol/David/Eve is the impostor

Constraints

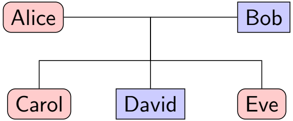
Exactly one impostor

What if we add $a_1 \overset{\text{DIMACS}}{\equiv}$ `1 0`? Yes, Alice **can be** the impostor.

$$a_1 = \mathit{True}, b_2 = \mathit{False}, c_3 = \mathit{False}, d_4 = \mathit{False}, e_5 = \mathit{False}$$

Is Alice the impostor?

Who is the impostor?



Viewpoint

$\mathbf{a_1}/\mathbf{b_2}/\mathbf{c_3}/\mathbf{d_4}/\mathbf{e_5}$: Alice/Bob/Carol/David/Eve is the impostor

Constraints

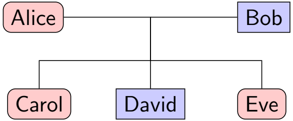
Exactly one impostor

What if we add $a_1 \overset{\text{DIMACS}}{\equiv}$ `1 0`? Yes, Alice **can be** the impostor.

$$a_1 = True, b_2 = False, c_3 = False, d_4 = False, e_5 = False$$

Is Alice the impostor? Now add $\neg a_1 \overset{\text{DIMACS}}{\equiv}$ `-1 0` instead.

Who is the impostor?


Alice
Bob
Carol
David
Eve

Viewpoint

a_1/b_2/c_3/d_4/e_5: Alice/Bob/Carol/David/Eve is the impostor

Constraints

Exactly one impostor

What if we add $a_1 \overset{\text{DIMACS}}{\equiv}$ 1 0? Yes, Alice **can be** the impostor.

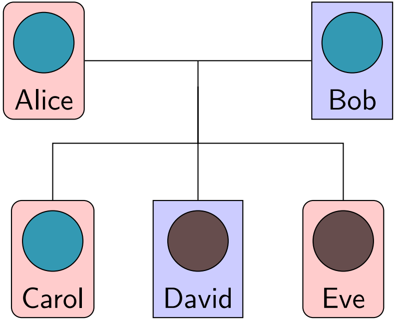
$$a_1 = \mathit{True}, b_2 = \mathit{False}, c_3 = \mathit{False}, d_4 = \mathit{False}, e_5 = \mathit{False}$$

Is Alice the impostor? Now add $\neg a_1 \overset{\text{DIMACS}}{\equiv}$ -1 0 instead. Alice **can also be** the real mother.

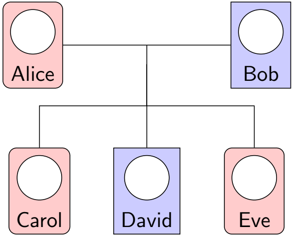
$$a_1 = \mathit{False}, b_2 = \mathit{False}, c_3 = \mathit{False}, d_4 = \mathit{True}, e_5 = \mathit{False}$$

Who is the impostor?

Let's add more clues: recessive treats! E.g.: blue eyes are recessive.



Who is the impostor?



Viewpoint

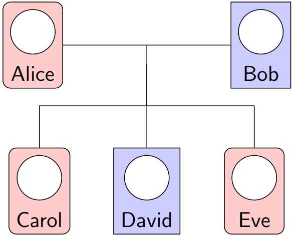
A_6/B_7/C_8/D_9/E_{10}: Alice/Bob/Carol/David/Eve has blue eyes.

Constraints

If neither the parent nor the child are impostor, and the parents have blue eyes, then the child has blue eyes.

$$\neg a_1 \wedge \neg b_2 \wedge \neg c_3 \wedge A_6 \wedge B_7 \rightarrow C_8$$

Who is the impostor?



Viewpoint

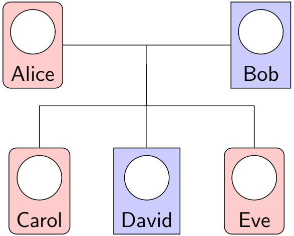
$\mathbf{A_6}/\mathbf{B_7}/\mathbf{C_8}/\mathbf{D_9}/\mathbf{E_{10}}$: Alice/Bob/Carol/David/Eve has blue eyes.

Constraints

If neither the parent nor the child are impostor, and the parents have blue eyes, then the child has blue eyes.

$$a_1 \lor b_2 \lor c_3 \lor \neg A_6 \lor \neg B_7 \lor C_8$$

Who is the impostor?



Viewpoint

$\mathbf{A_6/B_7/C_8/D_9/E_{10}}$: Alice/Bob/Carol/David/Eve has blue eyes.

Constraints

If neither the parent nor the child are impostor, and the parents have blue eyes, then the child has blue eyes.

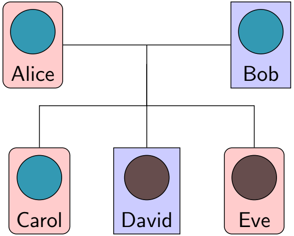
$$a_1 \lor b_2 \lor c_3 \lor \neg A_6 \lor \neg B_7 \lor C_8 \quad \overset{\text{DIMACS}}{\equiv} \quad 1\ 2\ 3\ {-6}\ {-7}\ 8\ 0$$

$$a_1 \lor b_2 \lor d_4 \lor \neg A_6 \lor \neg B_7 \lor D_9 \quad \equiv \quad 1\ 2\ 4\ {-6}\ {-7}\ 9\ 0$$

$$a_1 \lor b_2 \lor e_5 \lor \neg A_6 \lor \neg B_7 \lor E_{10} \quad \equiv \quad 1\ 2\ 5\ {-6}\ {-7}\ 10\ 0$$

Who is the impostor?

```
c Exactly one impostor
1 2 3 4 5 0
-1 -2 0
-1 -3 0
-1 -4 0
-1 -5 0
-2 -3 0
-2 -4 0
-2 -5 0
-3 -4 0
-3 -5 0
-4 -5 0
```



```
c Inheritance of blue eyes rules
1 2 3 -6 -7 8 0
1 2 4 -6 -7 9 0
1 2 5 -6 -7 10 0
c Eye color of the family members
6 0
7 0
8 0
-9 0
-10 0
```

Who is the impostor?

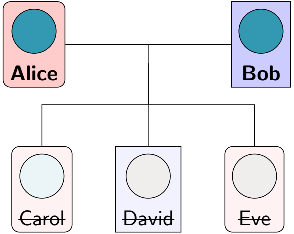
- Can Alice be the impostor? `1 0`
- Can Alice be a real family member? `-1 0`
- Can Bob be the impostor? `2 0`
- Can Bob be a real family member? `-2 0`
- Can Carol be the impostor? `3 0`
- Can Carol be a real family member? `-3 0`
- Can David be the impostor? `4 0`
- Can David be a real family member? `-4 0`
- Can Eve be the impostor? `5 0`
- Can Eve be a real family member? `-5 0`

Who is the impostor?

- Can Alice be the impostor? `1 0` YES
- Can Alice be a real family member? `-1 0` YES
- Can Bob be the impostor? `2 0` YES
- Can Bob be a real family member? `-2 0` YES
- Can Carol be the impostor? `3 0` NO
- Can Carol be a real family member? `-3 0` YES
- Can David be the impostor? `4 0` NO
- Can David be a real family member? `-4 0` YES
- Can Eve be the impostor? `5 0` NO
- Can Eve be a real family member? `-5 0` YES

Who is the impostor?

```
c Exactly one impostor
1 2 3 4 5 0
-1 -2 0
-1 -3 0
-1 -4 0
-1 -5 0
-2 -3 0
-2 -4 0
-2 -5 0
-3 -4 0
-3 -5 0
-4 -5 0
```



```
c Inheritance of blue eyes rules
1 2 3 -6 -7 8 0
1 2 4 -6 -7 9 0
1 2 5 -6 -7 10 0
c Eye color of the family members
6 0
7 0
8 0
-9 0
-10 0
c Extra: either Alice or Carol
c        are impostor
1 3 0
```

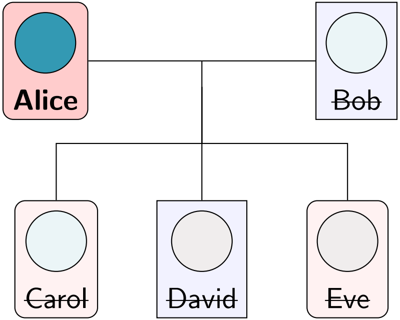
Who is the impostor?

- Can Alice be the impostor? `1 0`
- Can Alice be a real family member? `-1 0`
- Can Bob be the impostor? `2 0`
- Can Bob be a real family member? `-2 0`
- Can Carol be the impostor? `3 0`
- Can Carol be a real family member? `-3 0`
- Can David be the impostor? `4 0`
- Can David be a real family member? `-4 0`
- Can Eve be the impostor? `5 0`
- Can Eve be a real family member? `-5 0`

Who is the impostor?

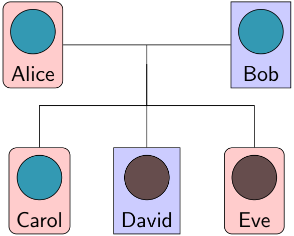
- Can Alice be the impostor? `1 0` YES
- Can Alice be a real family member? `-1 0` NO
- Can Bob be the impostor? `2 0` NO
- Can Bob be a real family member? `-2 0` YES
- Can Carol be the impostor? `3 0` NO
- Can Carol be a real family member? `-3 0` YES
- Can David be the impostor? `4 0` NO
- Can David be a real family member? `-4 0` YES
- Can Eve be the impostor? `5 0` NO
- Can Eve be a real family member? `-5 0` YES

Who is the impostor?



ALICE IS THE IMPOSTOR

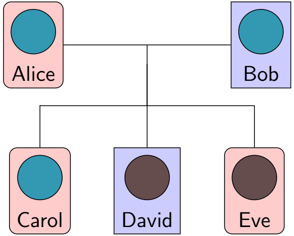
Who is the impostor?



$$\dfrac{\dfrac{\dfrac{a_1 \lor b_2 \lor d_4 \lor \neg A_6 \lor \neg B_7 \lor D_9 \qquad A_6}{a_1 \lor b_2 \lor d_4 \lor \neg B_7 \lor D_9} \qquad B_7}{a_1 \lor b_2 \lor d_4 \lor D_9} \qquad \neg D_9}{a_1 \lor b_2 \lor d_4}$$

$$\dfrac{\begin{array}{c}\text{If Alice,Bob,David are not impostor, and Alice,Bob have blue eyes} \rightarrow \text{David has blue eyes}\\ \text{Alice has blue eyes}\\ \text{Bob has blue eyes}\\ \text{David has not blue eyes}\end{array}}{\text{Either Alice, Bob or David is impostor}}$$

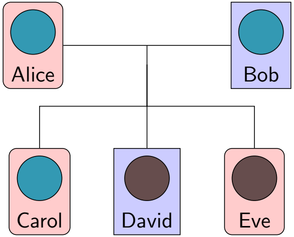
Who is the impostor?



$$\dfrac{\dfrac{\dfrac{a_1 \vee b_2 \vee e_5 \vee \neg A_6 \vee \neg B_7 \vee E_{10} \qquad A_6}{a_1 \vee b_2 \vee e_5 \vee \neg B_7 \vee E_{10}} \qquad B_7}{a_1 \vee b_2 \vee e_5 \vee E_{10}} \qquad \neg E_{10}}{a_1 \vee b_2 \vee e_5}$$

$$\frac{\begin{array}{c}\text{If Alice,Bob,Eve are not impostor, and Alice,Bob have blue eyes} \rightarrow \text{Eve has blue eyes}\\ \text{Alice has blue eyes}\\ \text{Bob has blue eyes}\\ \text{Eve has not blue eyes}\end{array}}{\text{Either Alice, Bob or Eve is impostor}}$$

Who is the impostor?



Either Alice, Bob or David is impostor
Either Alice, Bob or Eve is impostor
Either Alice or Carol impostor
There is at most one impostor

Alice is impostor

Outline

Conclusions and remarks

- There exist **many symbolic formalisms** to solve challenging problems.
    - They are usually interreducible.
    - **Different underlying solving techniques**.
    - No approach is the best in all fields.
    - They are used to solve computationally hard problems.
    - They can handle huge numbers of variables and constraints.

- Having a **good encoding** is as much important as having a good solver.

Conclusions and remarks

- **Puzzle problems are suitable** for learning on symbolic AI.

- Solving CSPs is an increasing (and never ending?) challenge:
  - Solve the problem **correctly**.
  - Solve the problem **more efficiently**.

- Model and solve is a good exercise to **train abstraction and reasoning capacities**.

Some Challenges of Problem Solving

- Complete versus incomplete solvers.
- Hybrid approaches to increase scalability.
- Combining machine learning and symbolic reasoning.
- Improved modeling tools.
- Providing human-understandable explanations for the derived solutions.
- Publicly available solvers and benchmarks. Competitions of solvers.
- Green algorithms.

Resources

CSPLib: A problem library for constraints

- CSP library: `https://www.csplib.org/`

SAT and MaxSAT

- SAT solver online:
  - `https://msoos.github.io/cryptominisat_web/`
- Python library OptiLog:
  - `https://ulog.udl.cat/static/doc/optilog/html/index.html`
- Competitions (Dimacs format, instances, solvers, . . .):
  - `https://satcompetition.github.io/2022/`
  - `https://maxsat-evaluations.github.io/2022/`

SMT

- Competition (SMTLIB format, benchmarks, solvers, . . .):
  - `https://smt-comp.github.io/2022/`

Resources

Mixed Integer Linear Programming

- Solvers:
  - GNU Linear Programming Kit: `https://www.gnu.org/software/glpk/`
  - IBM ILOG CPLEX: `https://www.ibm.com/es-es/products/ilog-cplex-optimization-studio`
  - SCIP: `https://www.scipopt.org/`

Constraint Programming

- CP languages and solvers:
  - Essence' and SavileRow: `https://savilerow.cs.st-andrews.ac.uk/`
  - Minizinc: `https://www.minizinc.org/`
  - IBM ILOG CP Optimizer: `https://www.ibm.com/es-es/analytics/cplex-cp-optimizer`

Escuela AI HUB de verano
Sobre IA y Educación

Symbolic AI

Felip Manyà, IIIA-CSIC
Jordi Coll, UdG

JULY 2ND, 2024