Algebraic cryptanalysis and multivariate cryptography

Monika Trimoska

PQSCA summer school
June 17, Albena, Bulgaria

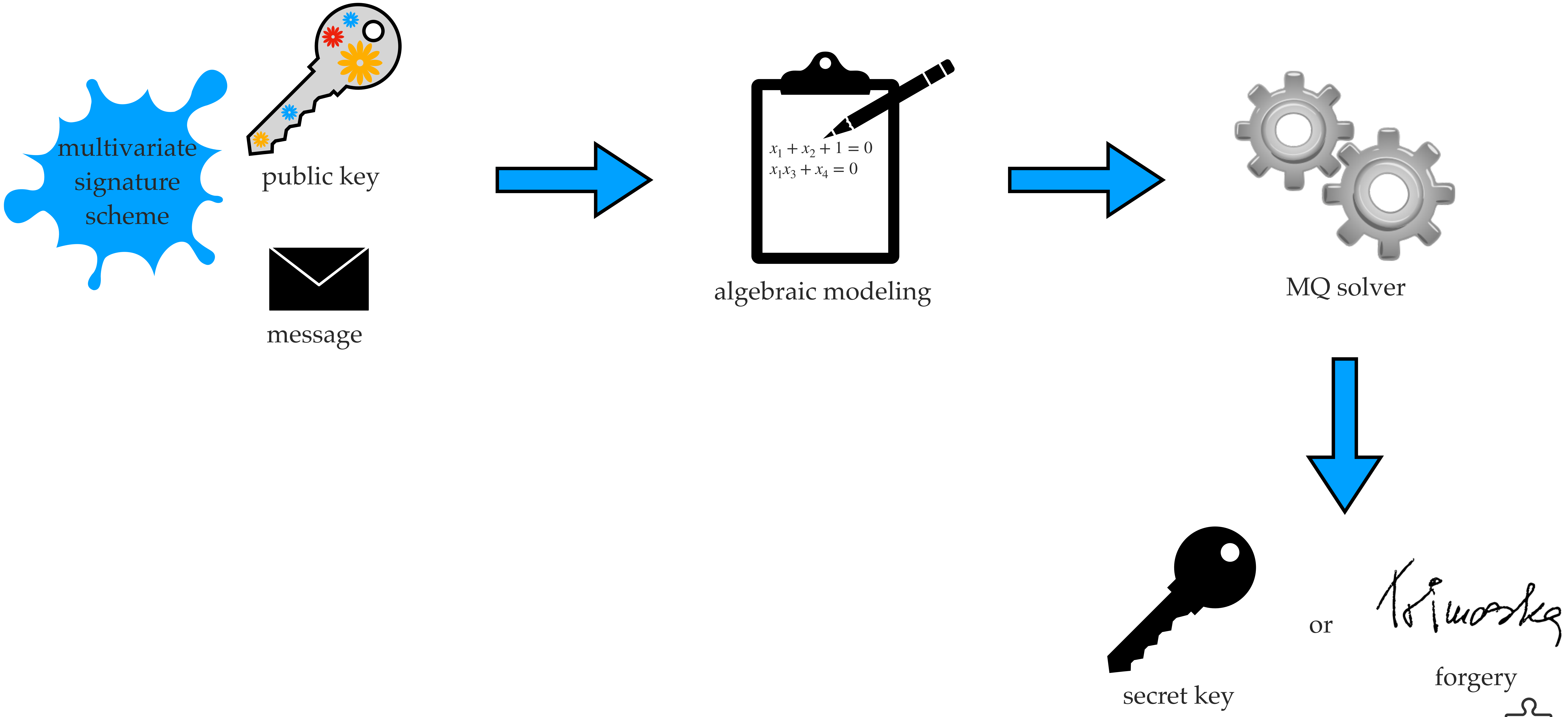
TU/e

*Animated version at: https://mtrimoska.com/slides/PQCSA_cryptanalysis/

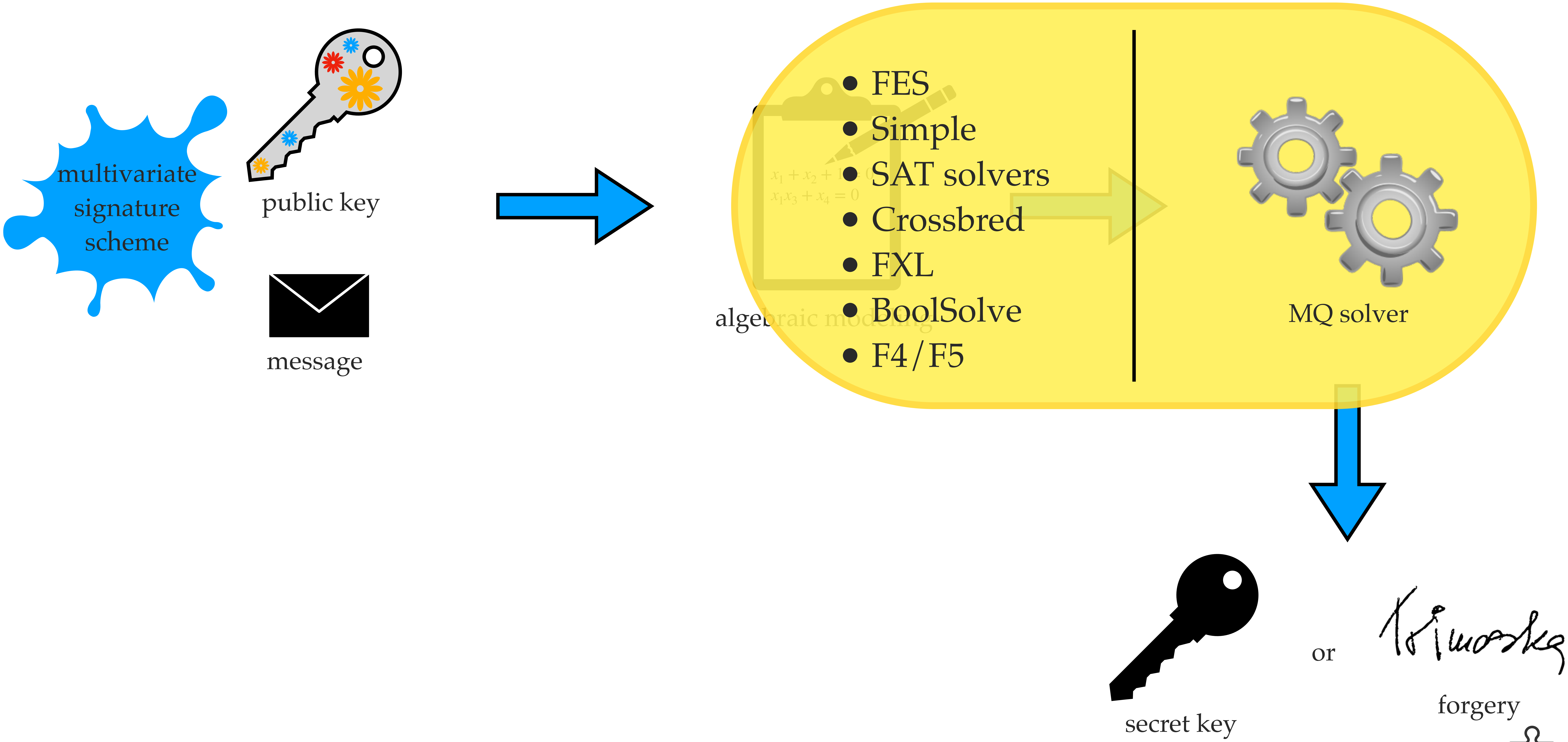
Algebraic cryptanalysis

A type of cryptanalytic methods where the problem of finding the secret key (or any attack goal) is reduced to the problem of finding a solution to a nonlinear multivariate polynomial system of equations.

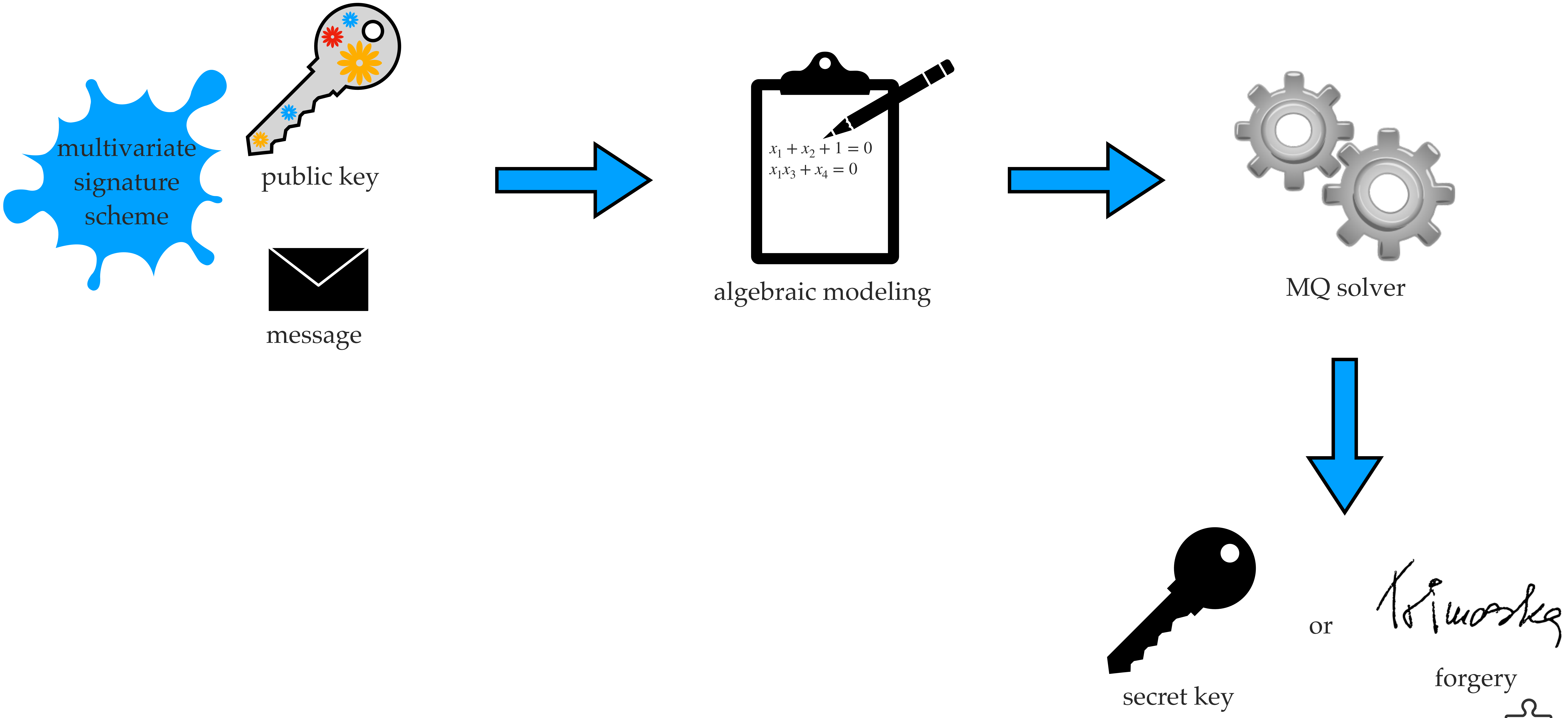
Algebraic cryptanalysis



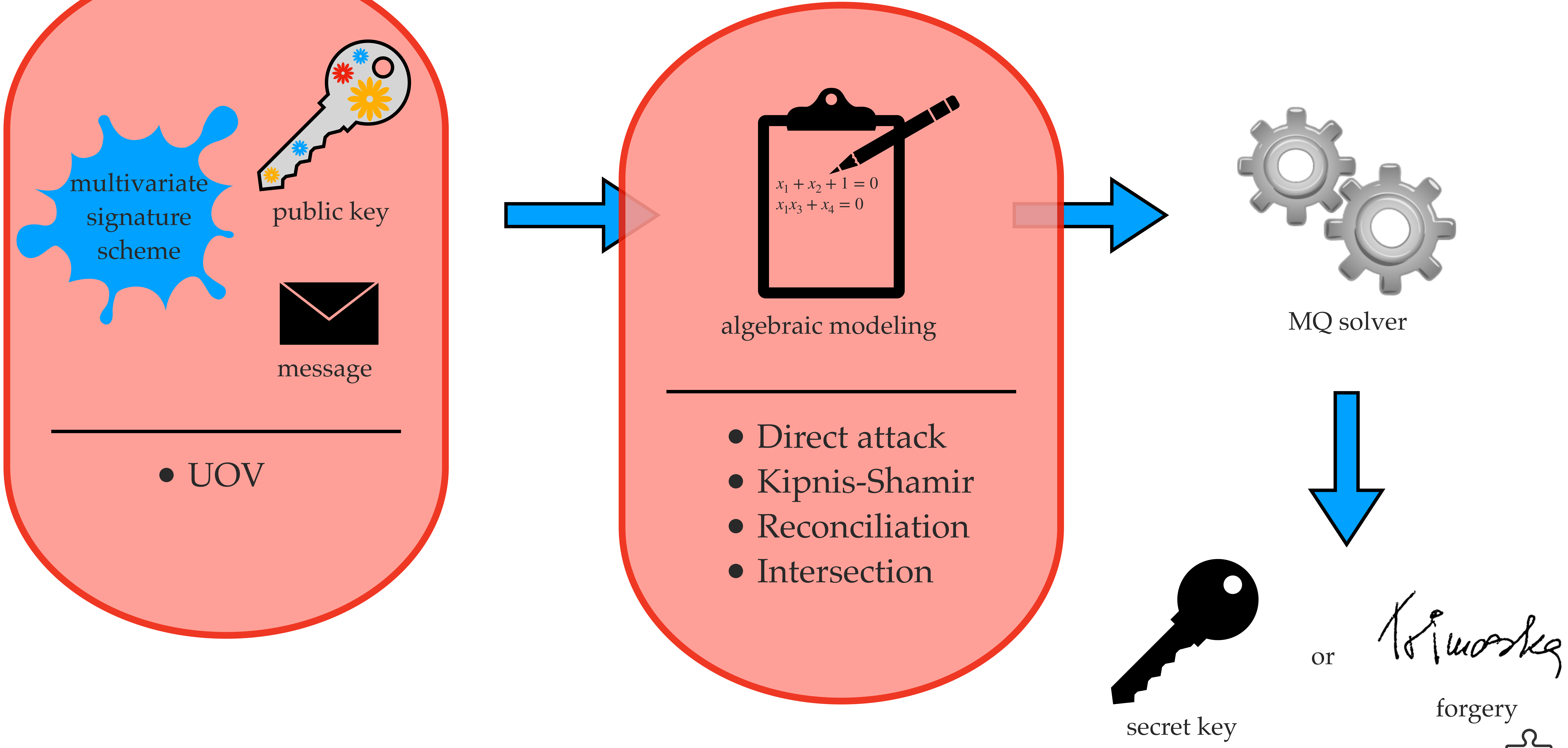
Algebraic cryptanalysis



Algebraic cryptanalysis



Algebraic cryptanalysis



The MQ problem (recall)

The MQ problem

Given m multivariate quadratic polynomials $f_1, \ldots, f_m$ of n variables over a finite field $\mathbb{F}_q$, find a tuple $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{F}_q^n$, such that $f_1(\mathbf{x}) = \ldots = f_m(\mathbf{x}) = 0$.

Example.

$$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$$

$$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$$

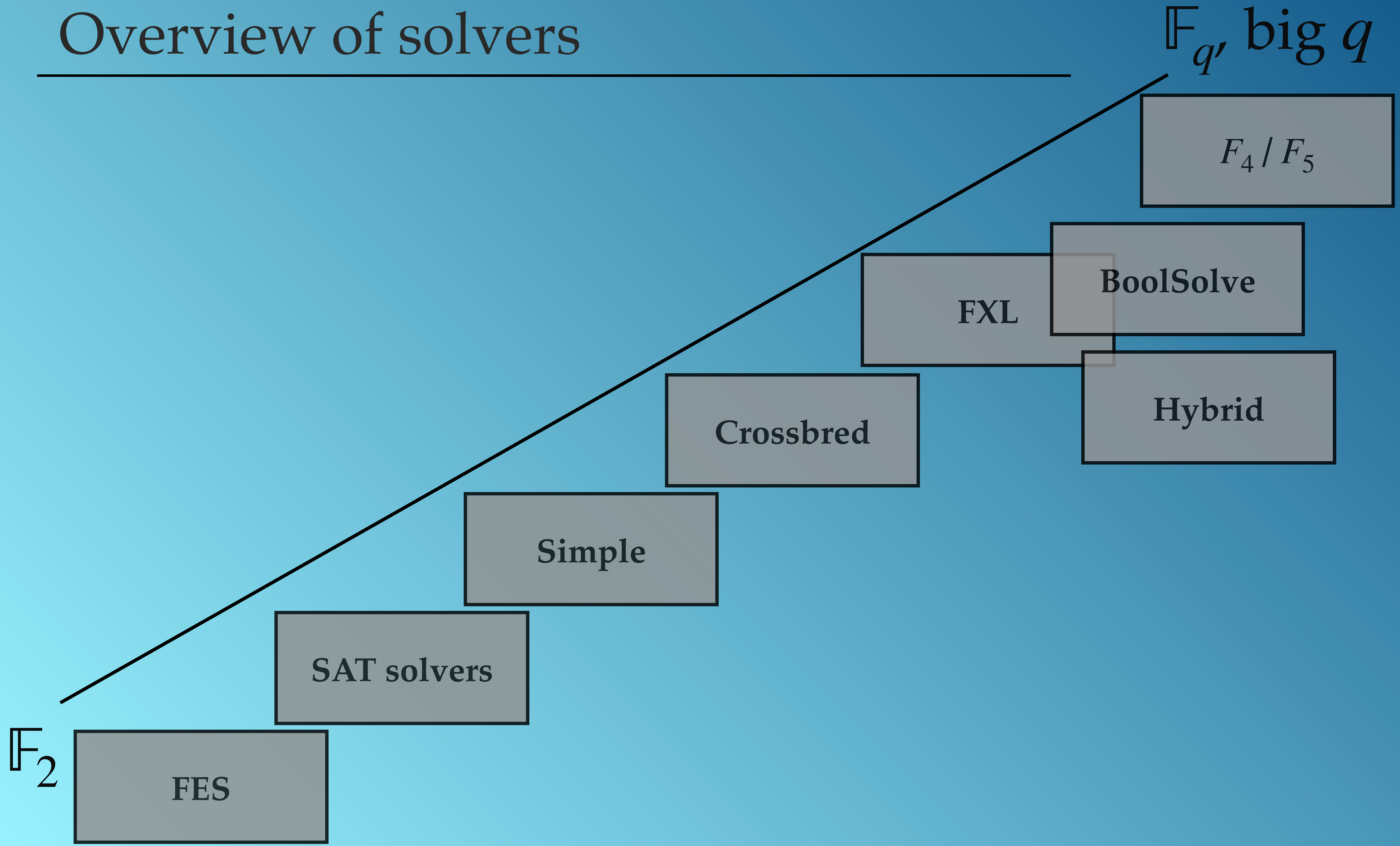
$$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$$

$$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$$

$$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$$

$$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$$

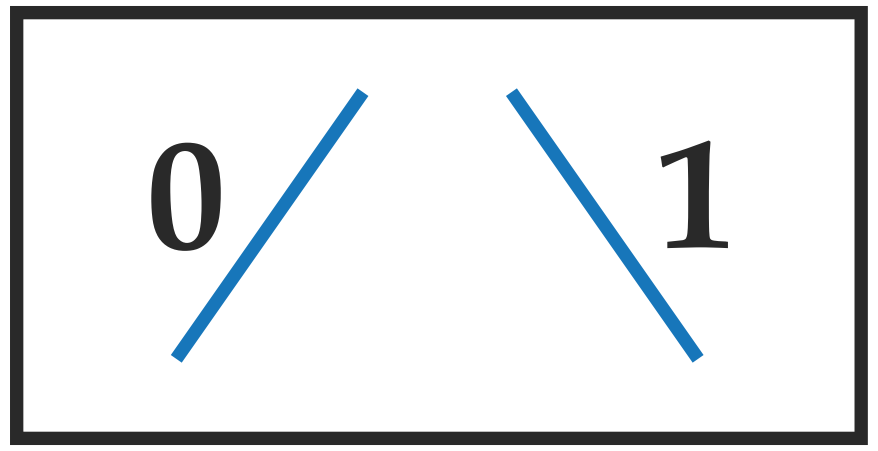
Overview of solvers

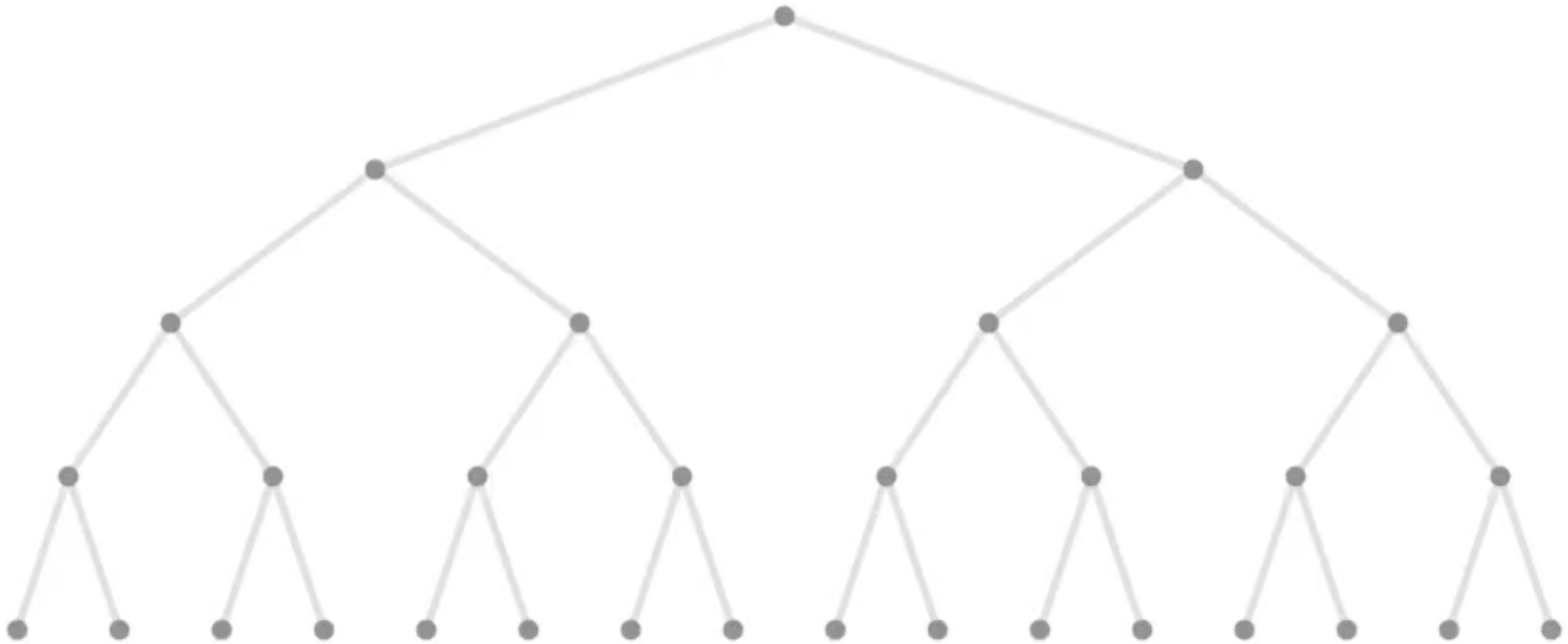


(Fast) Exhaustive Search

[Bouillaguet, Chen, Cheng, Chou, Niederhagen, Shamir, Yang, 2010]

Exhaustive Search





Binary search tree

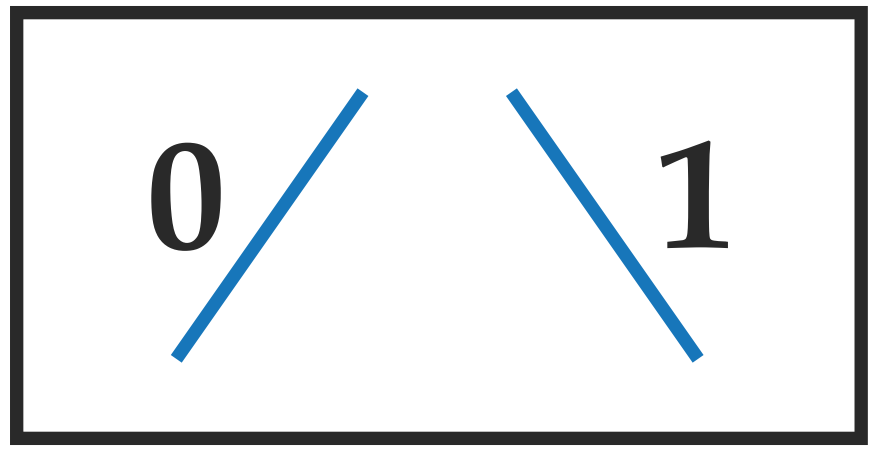
$$x_1 \cdot x_2 + x_1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$

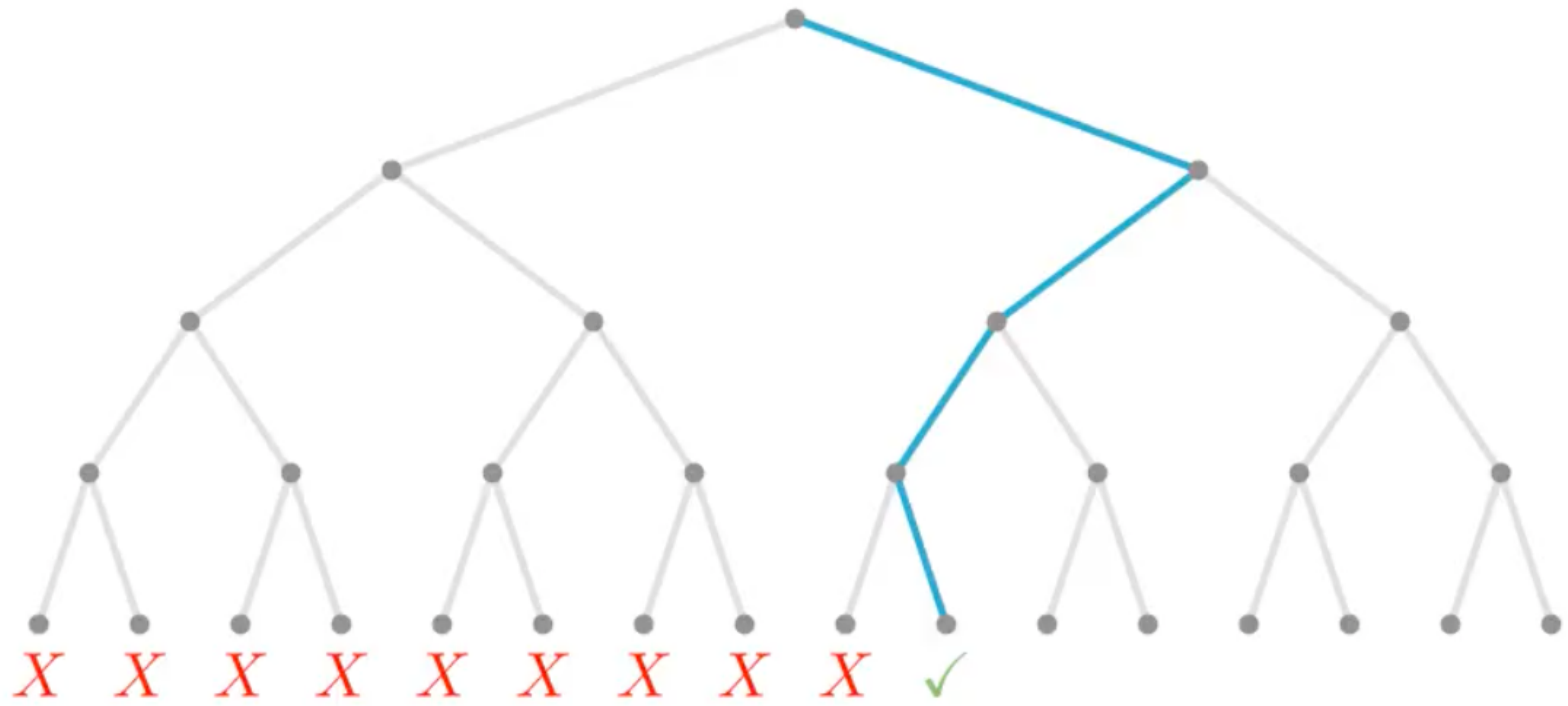
$$x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_2 + 1 = 0$$

$$x_1 \cdot x_2 + x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_4 = 0$$

$$x_1 \cdot x_4 + x_2 \cdot x_3 + x_2 + x_3 + x_4 = 0$$

Exhaustive Search





Binary search tree

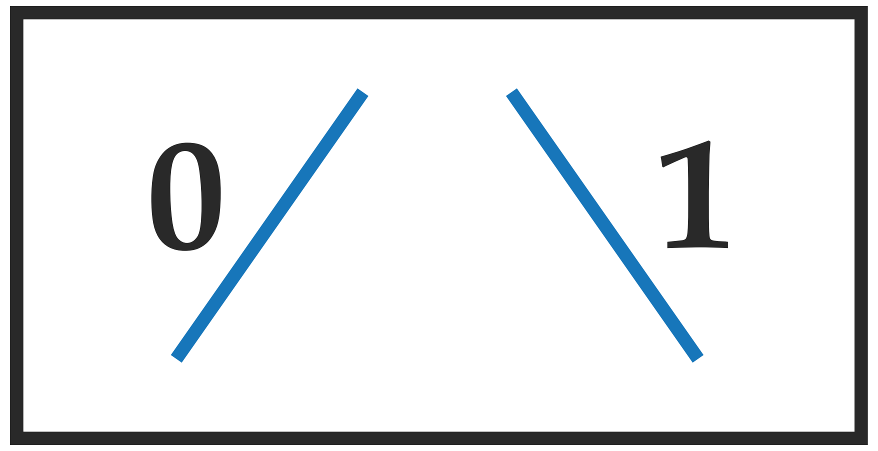
$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

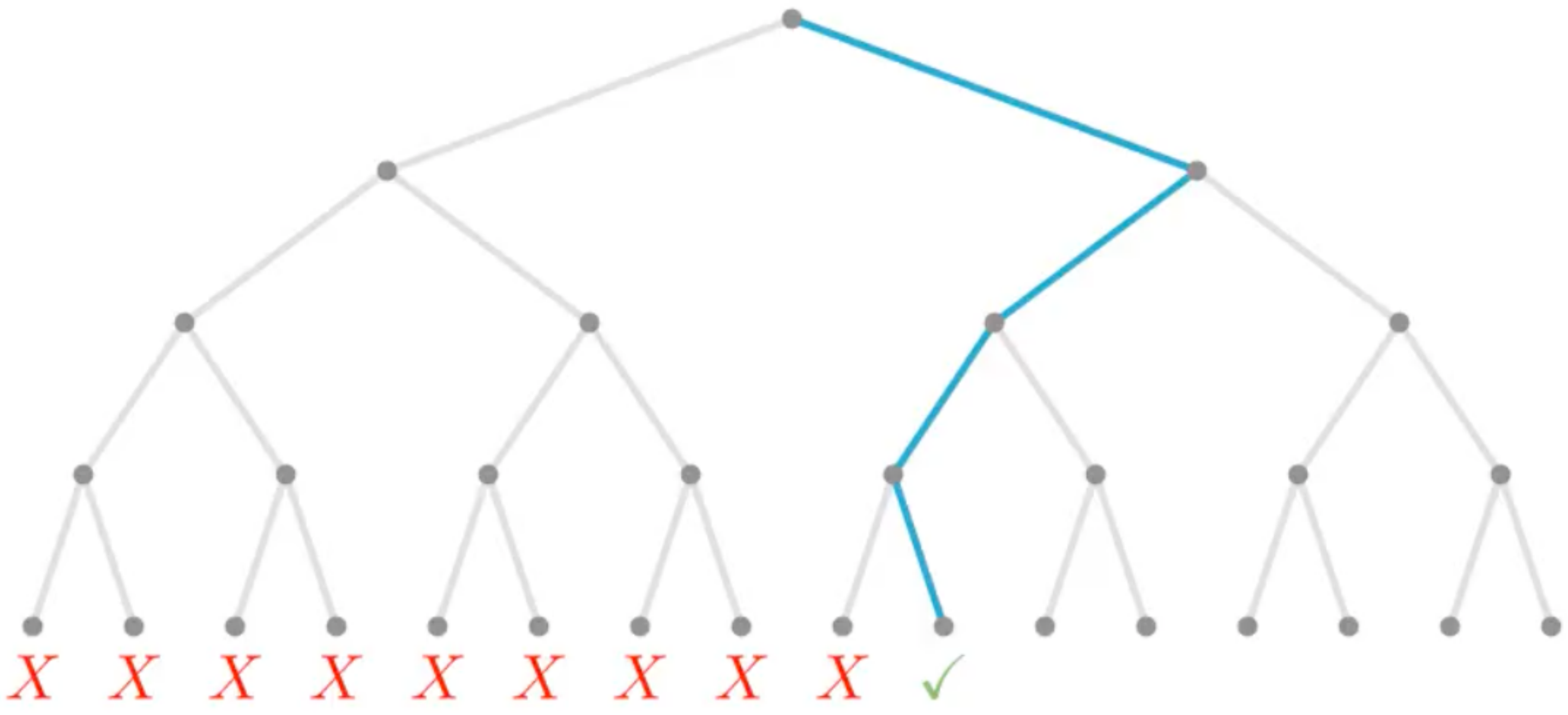
$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

Exhaustive Search



Worst-case complexity: $\mathcal{O}(2^n)$



Binary search tree

$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

Fast Exhaustive Search

* The libFES solver

Gray code

- An ordering of the binary system where two successive values differ in only one bit.

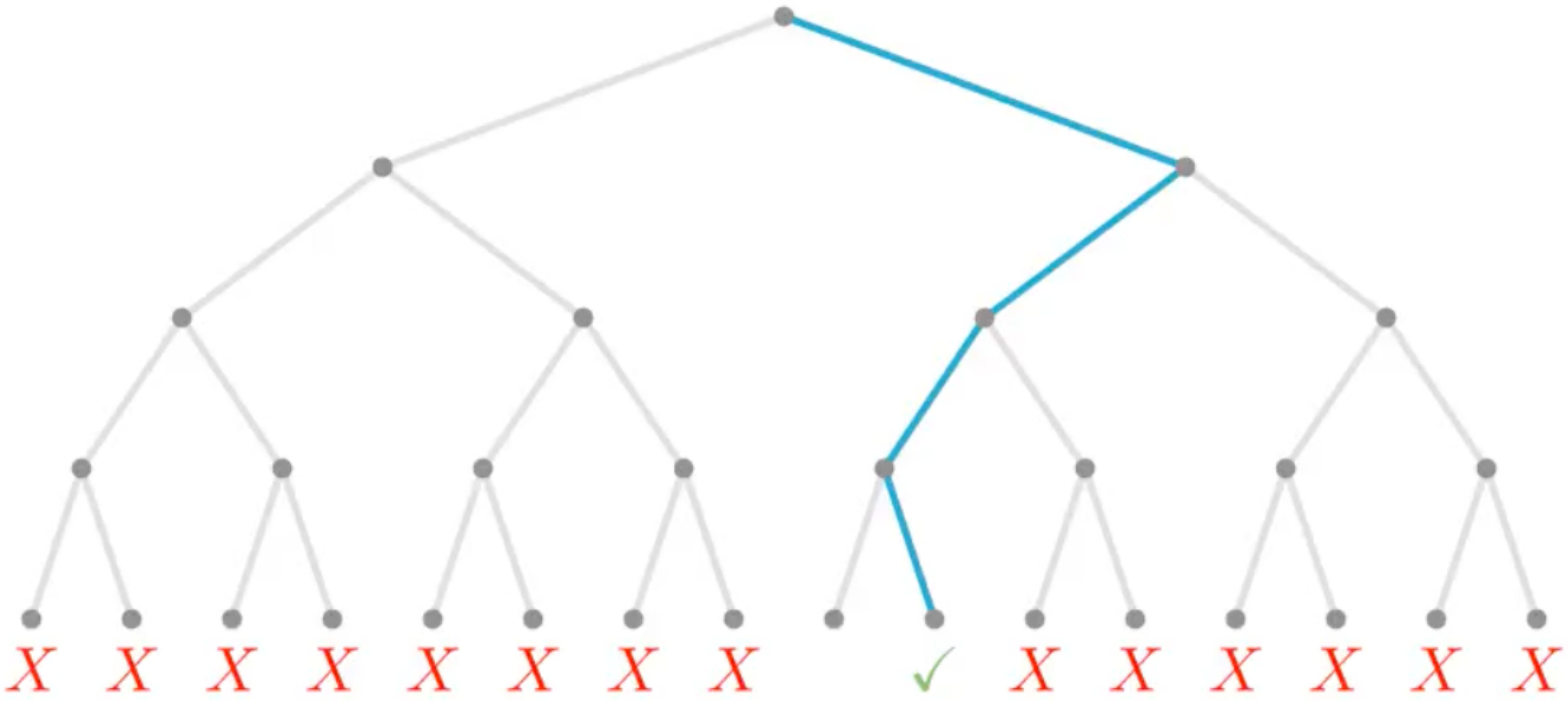
Example. $n = 4$

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000

Fast Exhaustive Search

Gray code

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000



$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

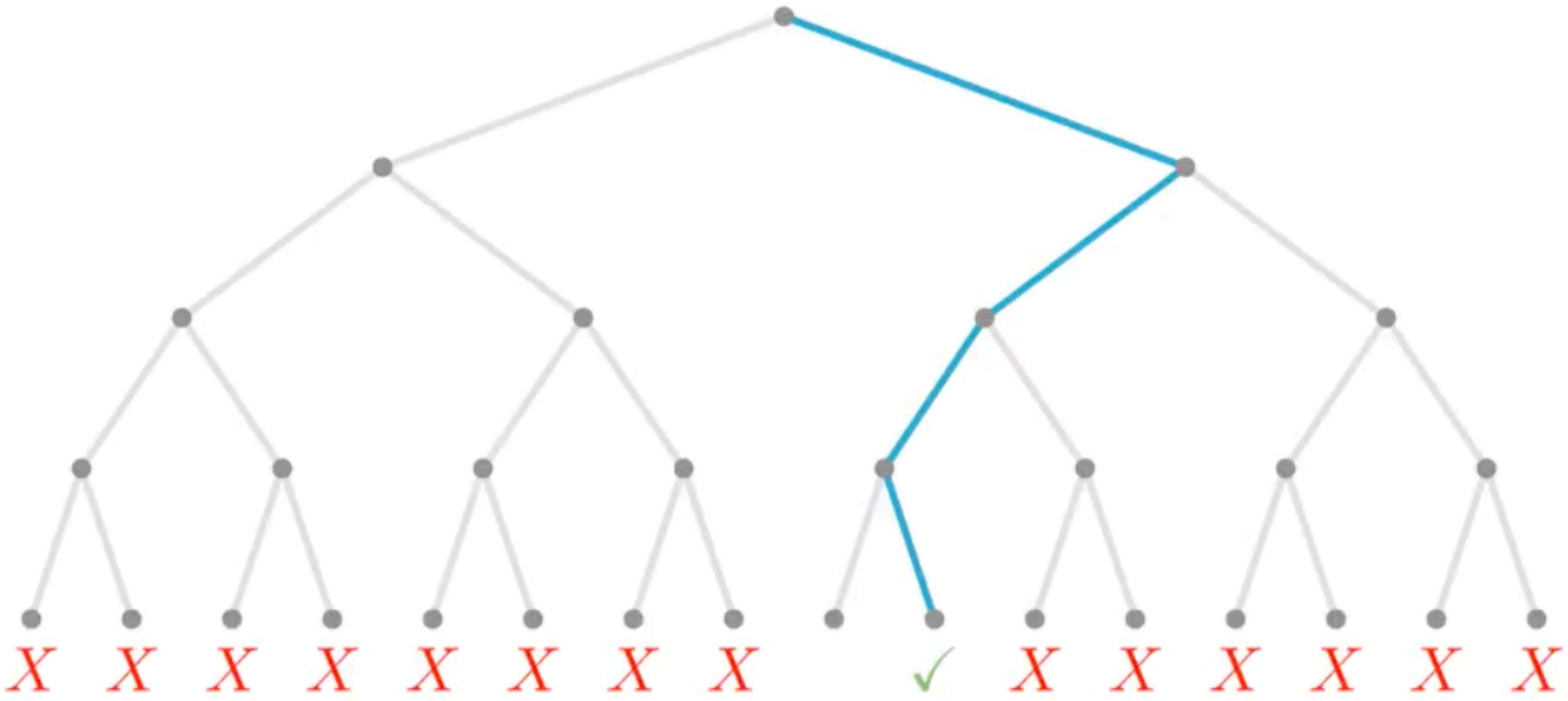
Fast Exhaustive Search

Worst-case complexity: $\mathcal{O}(2^n)$

! But, it differs from the depth-first traversal in the polynomial factors

Gray code

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000



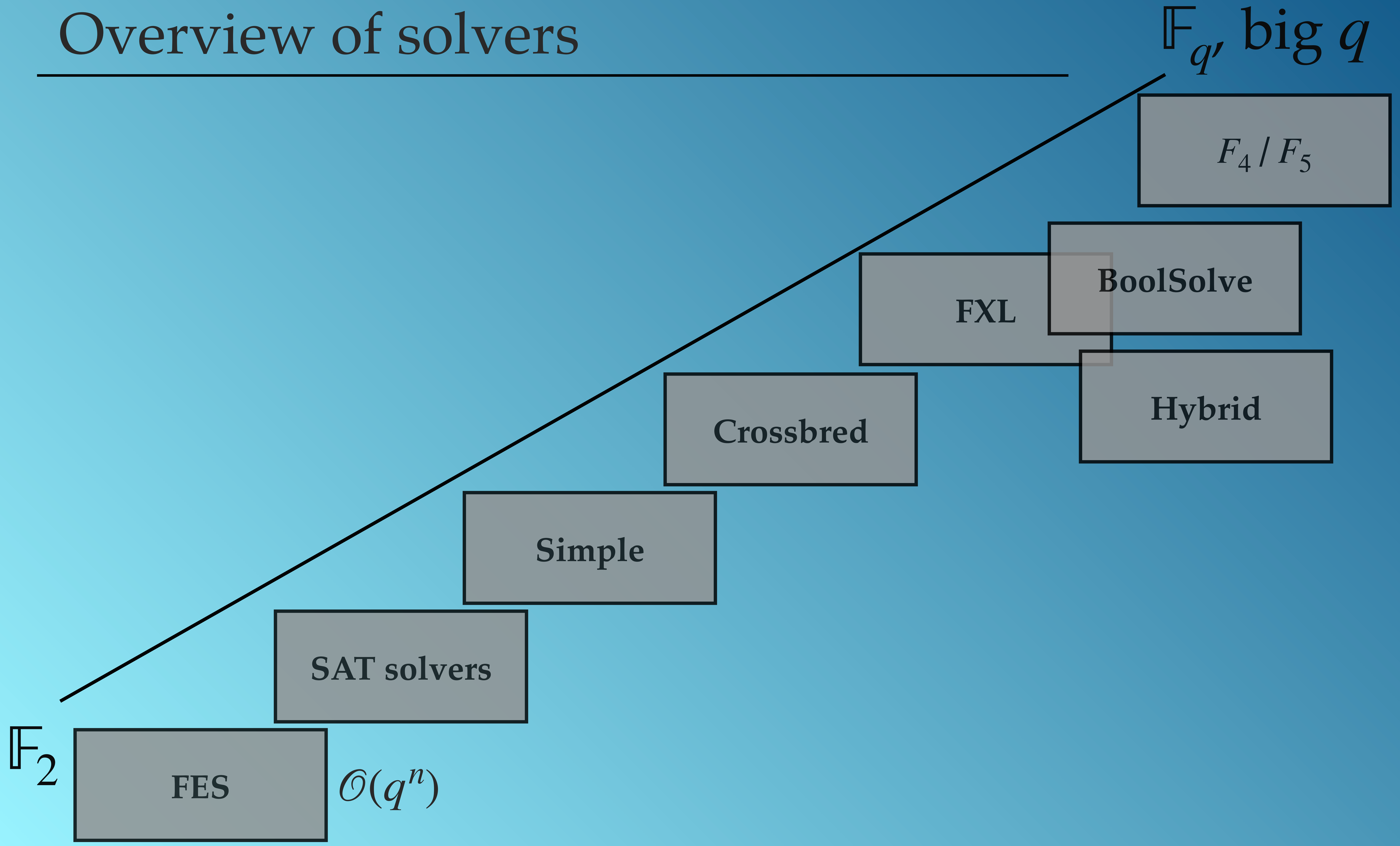
$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

Overview of solvers



SAT solvers

CryptoMiniSat [Soos, Nohl, Castelluccia, 2009], WDSat [T., Dequen, Ionica, 2020]

Simple algorithm

[Bouillaguet, Delaplace, T., 2021]

(SAT solvers)

- Propositional formula in Conjunctive Normal Form (CNF): a conjunction of clauses where each clause is a disjunction of literals and where each literal is a variable or a negated variable.

Example.

$$(x_1 \lor \neg x_2) \land$$
$$(x_2 \lor x_3 \lor x_4) \land$$
$$(\neg x_1 \lor x_4)$$

(SAT solvers)

- Propositional formula in Conjunctive Normal Form (CNF): a conjunction of clauses where each clause is a disjunction of literals and where each literal is a variable or a negated variable.

Example.
$$(x_1 \lor \neg x_2) \land$$
$$(x_2 \lor x_3 \lor x_4) \land$$
$$(\neg x_1 \lor x_4)$$

The SATisfiability problem

Given a propositional formula, determine whether there exists an interpretation (assignment of all variables) such that the formula is satisfied (evaluates to TRUE).

(SAT solvers)

- Propositional formula in Conjunctive Normal Form (CNF): a conjunction of clauses where each clause is a disjunction of literals and where each literal is a variable or a negated variable.

Example.

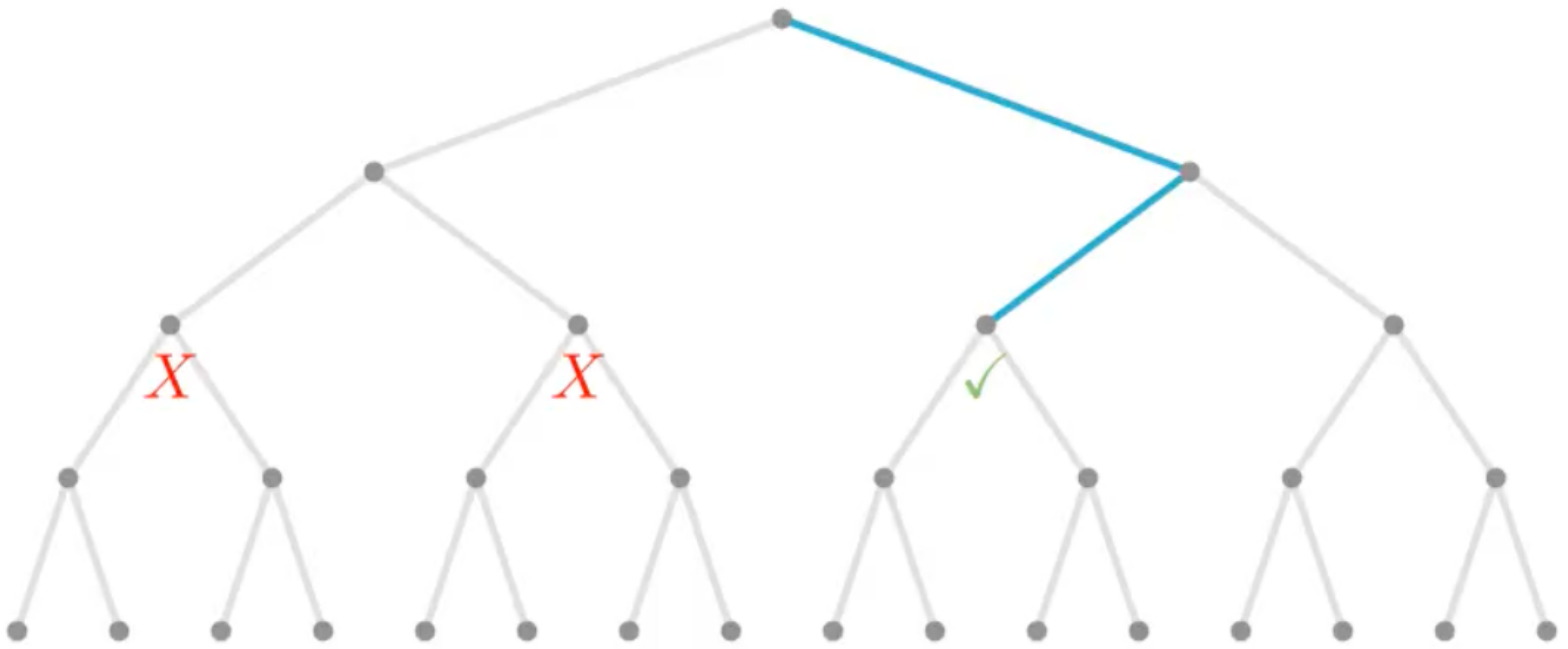
$$
\begin{aligned}
&(x_1 \vee \neg x_2) \wedge \\
&(x_2 \vee x_3 \vee x_4) \wedge \\
&(\neg x_1 \vee x_4)
\end{aligned}
$$

The SATisfiability problem

Given a propositional formula, determine whether there exists an interpretation (assignment of all variables) such that the formula is satisfied (evaluates to TRUE).

SAT solver: a tool for solving the SAT problem.

Partial assignment and conflicts



$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$

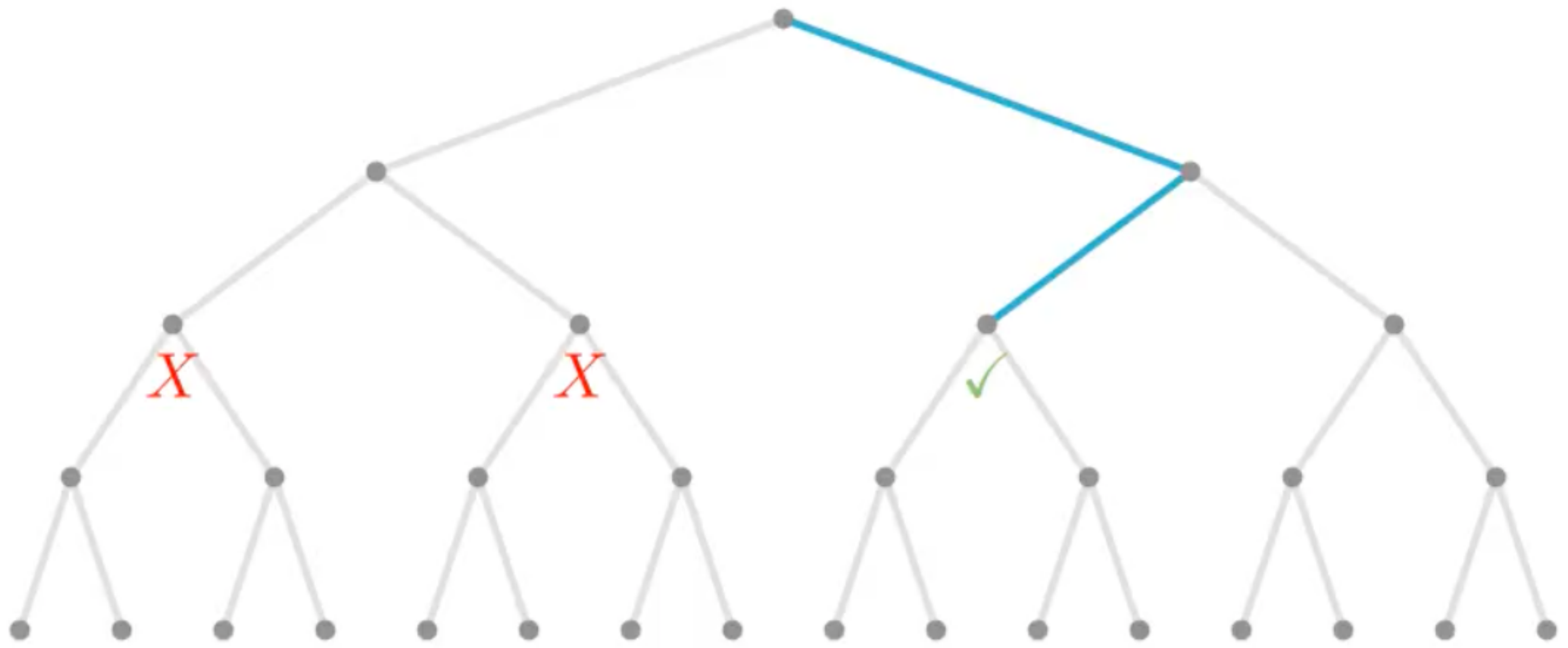
$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$

$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

Partial assignment and conflicts

Which (portion of) branches are missing ??



$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$

$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$

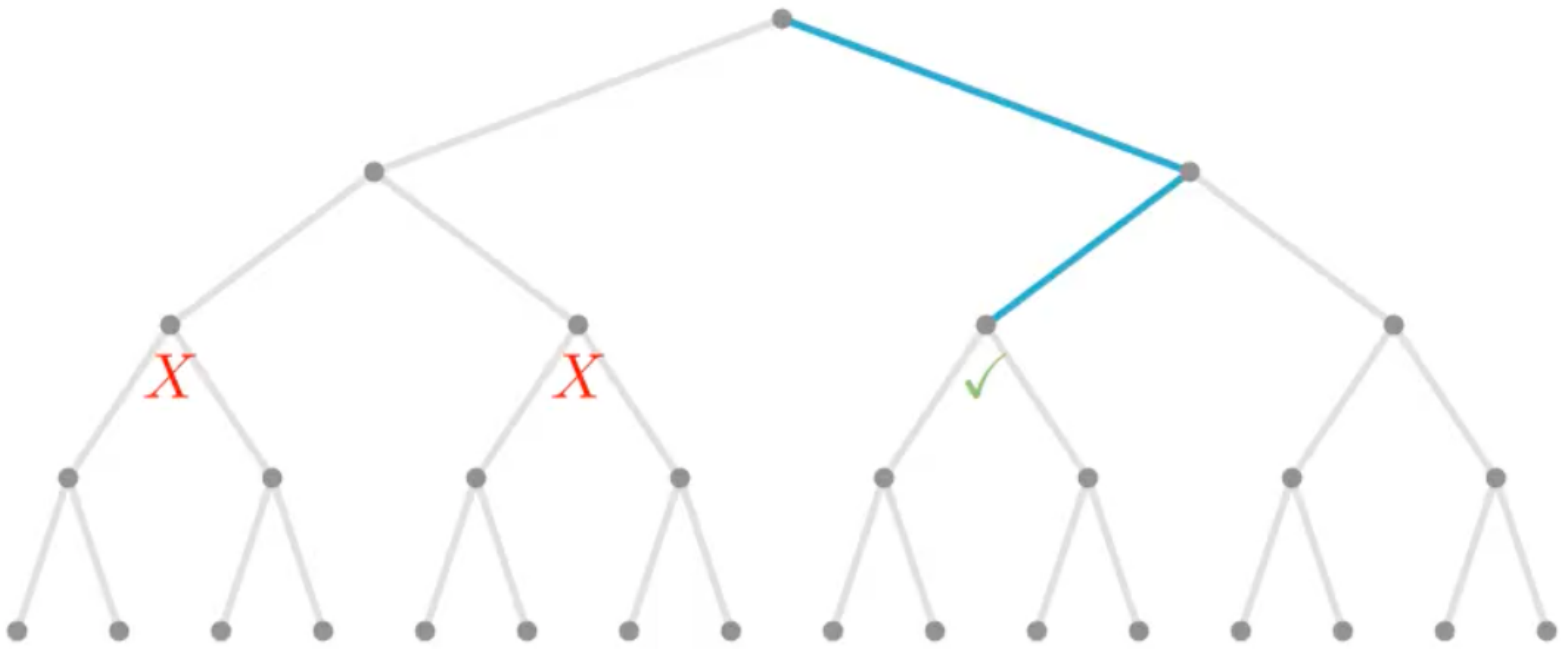
$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$

$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

Partial assignment and conflicts

Which (portion of) branches are missing ??

→ Worst-case complexity: $\mathcal{O}(2^n)$

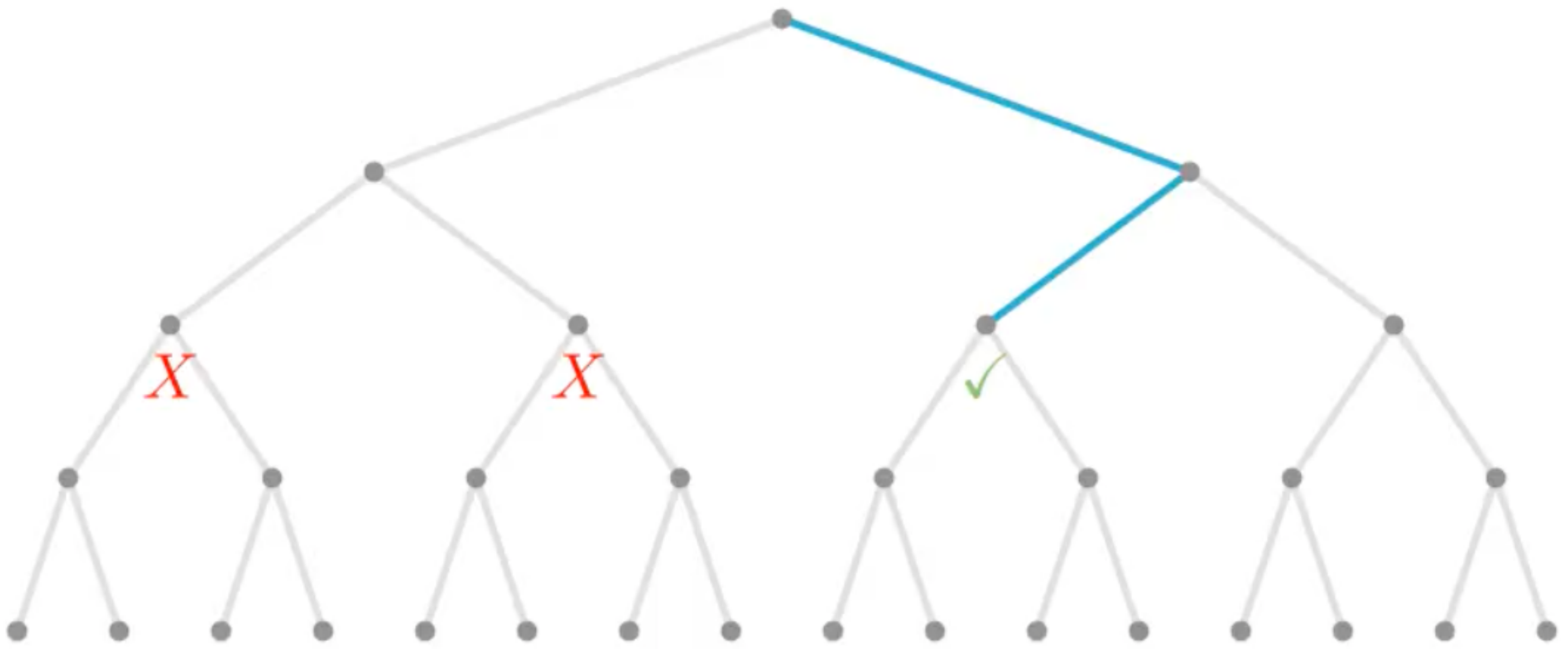


$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$
$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$
$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$
$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

Partial assignment and conflicts

Which (portion of) branches are missing ??

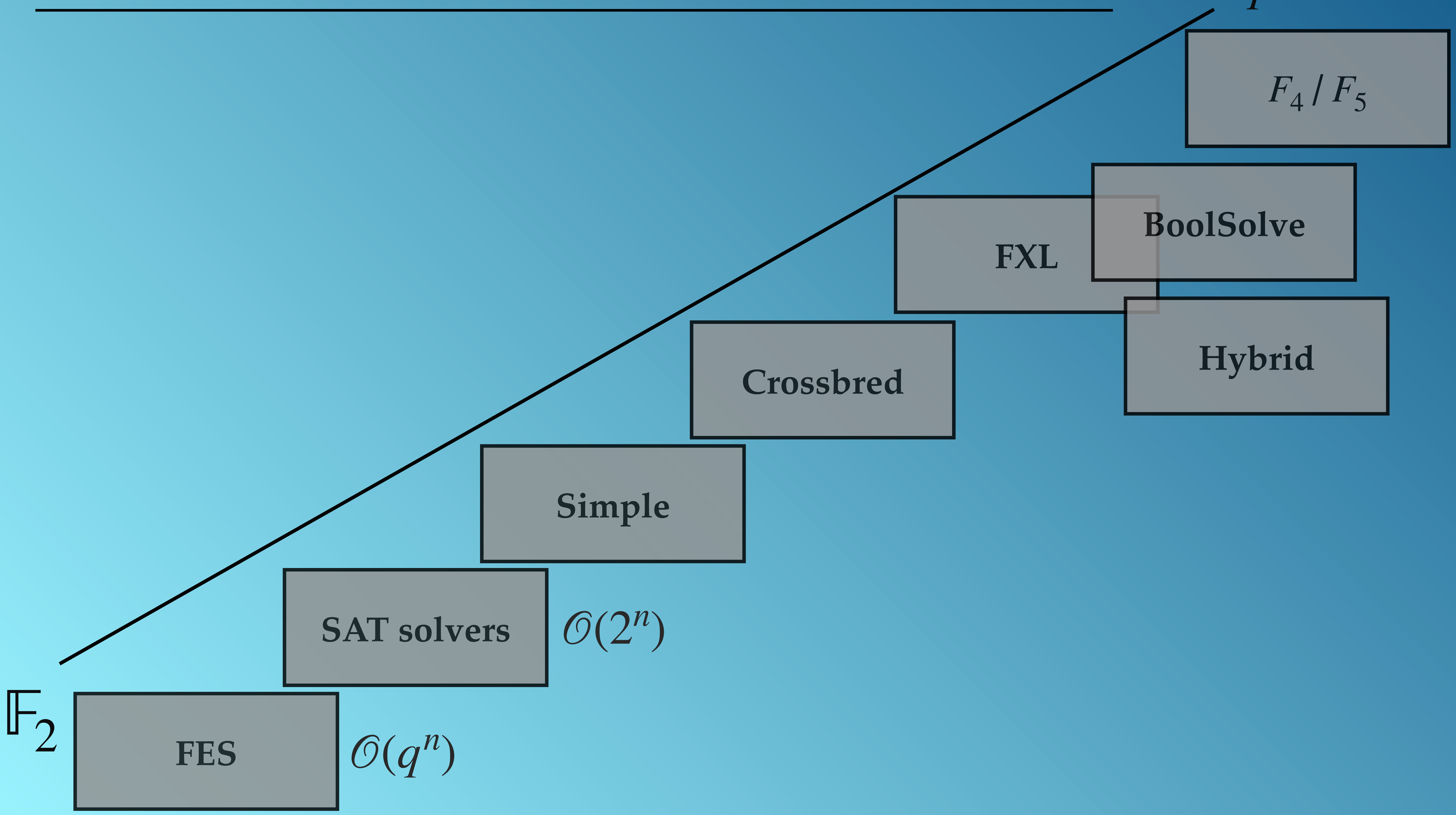
Worst-case complexity: $\mathcal{O}(2^n)$



$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$
$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$
$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$
$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

XOR-enabled SAT solvers: take as input XOR constraints as well; perform Gaussian elimination;
*CryptoMiniSat, WDSat

Overview of solvers



Macaulay matrix

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

↪ Linearisation: for each nonlinear monomial, replace all of its occurrences by a new variable.

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

Linearisation: for each nonlinear monomial, replace all of its occurrences by a new variable.

Example.

$$
\begin{aligned}
f_1 &: x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0\\
f_2 &: x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0\\
f_3 &: x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0\\
f_4 &: x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0\\
f_5 &: x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0\\
f_6 &: x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
\quad\longrightarrow\quad
\begin{aligned}
f_1 &: y_2 + y_5 + x_1 + x_3 + x_4 = 0\\
f_2 &: y_4 + y_3 + y_6 + x_1 + x_2 + x_4 = 0\\
f_3 &: y_5 + y_6 + x_1 + x_3 + 1 = 0\\
f_4 &: y_1 + y_2 + y_4 + x_3 + x_4 + 1 = 0\\
f_5 &: y_1 + y_4 + y_3 + x_3 = 0\\
f_6 &: y_2 + y_3 + y_6 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
$$

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

Linearisation: for each nonlinear monomial, replace all of its occurrences by a new variable.

Example.

$$
\begin{aligned}
f_1 &: x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0 \\
f_2 &: x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0 \\
f_3 &: x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0 \\
f_4 &: x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0 \\
f_5 &: x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0 \\
f_6 &: x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
\longrightarrow
\begin{aligned}
f_1 &: y_2 + y_5 + x_1 + x_3 + x_4 = 0 \\
f_2 &: y_4 + y_3 + y_6 + x_1 + x_2 + x_4 = 0 \\
f_3 &: y_5 + y_6 + x_1 + x_3 + 1 = 0 \\
f_4 &: y_1 + y_2 + y_4 + x_3 + x_4 + 1 = 0 \\
f_5 &: y_1 + y_4 + y_3 + x_3 = 0 \\
f_6 &: y_2 + y_3 + y_6 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
$$

Linearisation

Linearisation adds solutions: a *random* quadratic system of m equations in n variables, when $n = m$, is expected to have one solution (probability is $\sim \frac{1}{q}$ for systems over $\mathbb{F}_q$). The corresponding linearised system has a solution space of dimension $\binom{n+1}{2} - m$.

$\binom{n}{2}$ quadratic plus n linear monomials

Linearisation

Linearisation adds solutions: a *random* quadratic system of m equations in n variables, when $n = m$, is expected to have one solution (probability is $\sim \frac{1}{q}$ for systems over $\mathbb{F}_q$). The corresponding linearised system has a solution space of dimension $\binom{n+1}{2} - m$.

$\binom{n}{2}$ quadratic plus n linear monomials

Loss of information: e.g. assignment $x_1 = 1; x_2 = 0; y_1 = 1;$ is part of a valid solution to the linearised system, but $x_1 x_2 \neq y_1$.

Macaulay matrix

Monomials →

Equations ↓

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1											
f_2											
f_3											
f_4											
f_5											
f_6											

$$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$$
$$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$$
$$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$$
$$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$$
$$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$$
$$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$$

Macaulay matrix

Monomials →

Equations ↓

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$

$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$

$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$

$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$

$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$

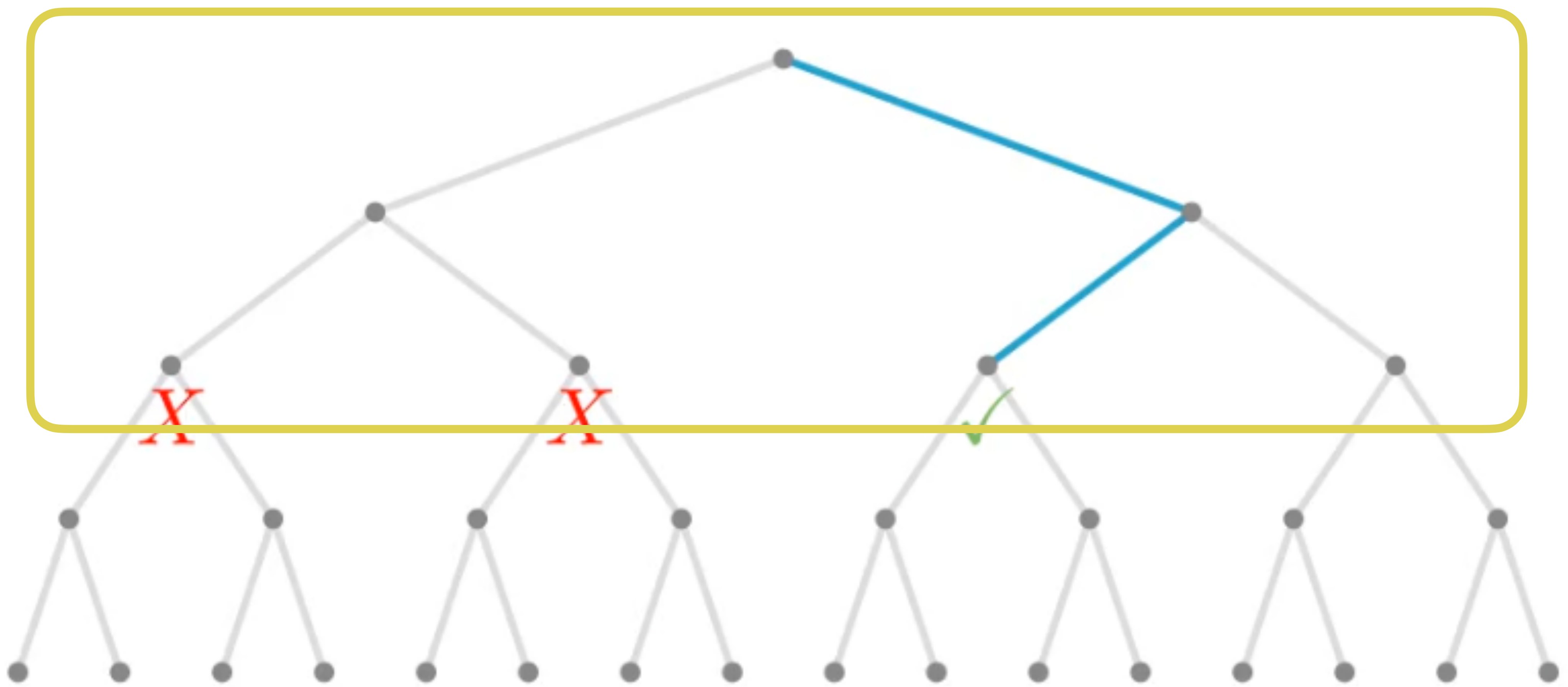
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

Simple algorithm

[Bouillaguet, Delaplace, T., 2021]

Simple algorithm

- Partial assignment
  - Gaussian elimination



$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$
$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$
$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$
$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

Simple algorithm

Guess sufficiently many variables so that the remaining polynomial system can be solved by linearization.

Simple algorithm: complexity

Simple algorithm: complexity

- n - number of variables
- m - number of equations

Simple algorithm: complexity

- n - number of variables
- m - number of equations

↳ Enumeration ends when:

number of monomials ≤ number of equations

Simple algorithm: complexity

- n - number of variables
- m - number of equations

Enumeration ends when:

number of monomials $\leq$ number of equations

$$\binom{n-?}{2} \leq m$$

Simple algorithm: complexity

- n - number of variables
- m - number of equations

Enumeration ends when:

number of monomials ≤ number of equations

$$\binom{n-?}{2} \leq m$$

$$\mathcal{O}(2^{n-\sqrt{2m}})$$

Simple algorithm: complexity

- n - number of variables
- m - number of equations

↳ Enumeration ends when:

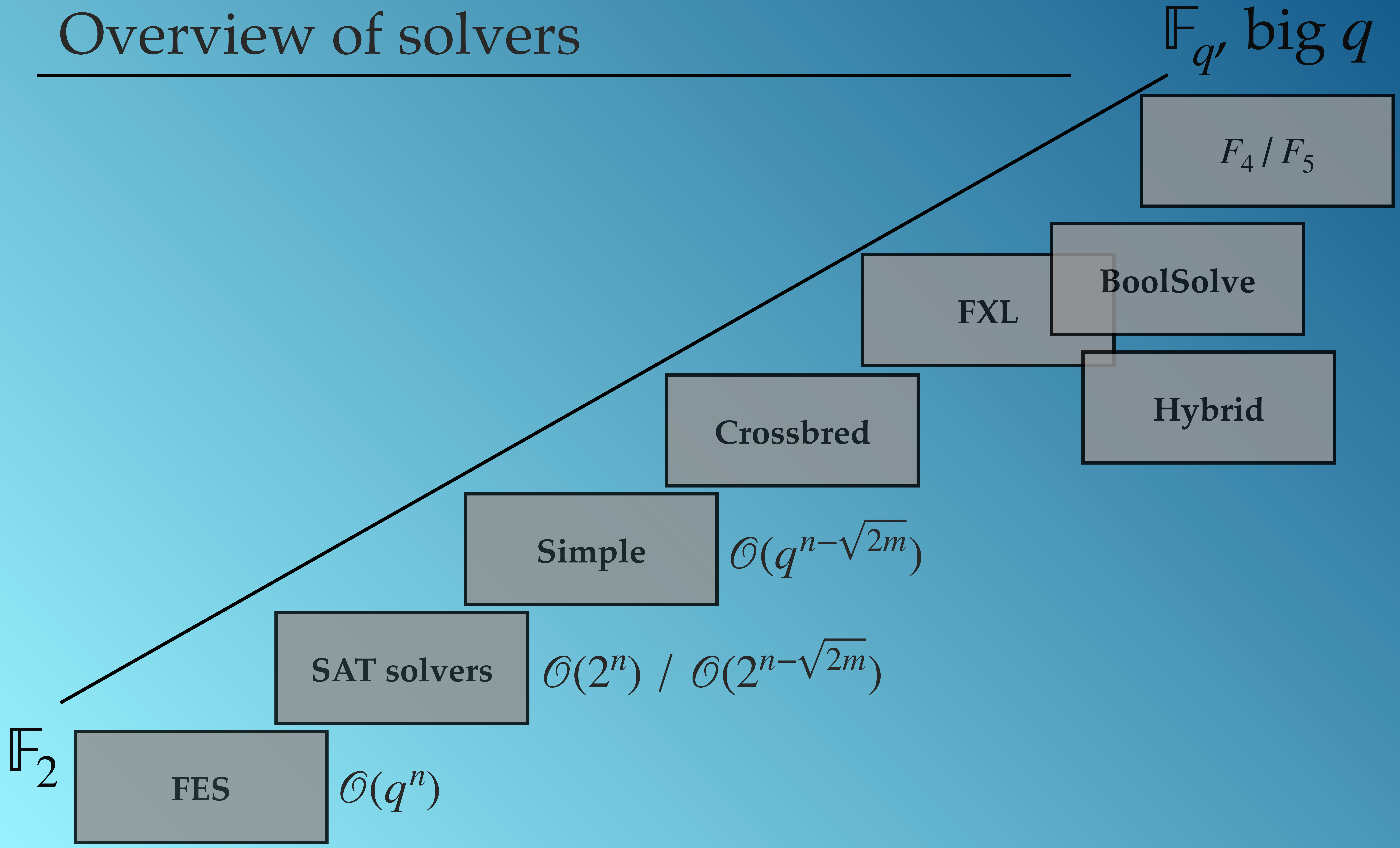
number of monomials ≤ number of equations

$$\binom{n-?}{2} \leq m$$

↳ $\mathcal{O}(2^{n-\sqrt{2m}})$

→ See also: Quantum BDT [Edme, Fouque, Schrottenloher]

Overview of solvers



Gröbner basis algorithms

[Buchberger, 1965]
[Lazard, 1983]
F_4/F_5 [Faugère, 1999/2002]
(XL [Courtois, Klimov, Patarin, Shamir, 2000])

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

$D = 3$

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$
f_1	0	1	0	1	0	1	0	0	1	1	0				
f_2	0	0	1	1	1	0	1	1	0	1	0				
f_3	0	0	0	1	0	1	0	1	1	0	1				
f_4	1	1	0	1	1	0	0	0	1	1	1				
f_5	1	0	1	1	1	0	0	0	1	0	0				
f_6	0	1	1	1	0	0	1	1	1	1	0				
x_1f_1															
x_2f_1															
$\dots$															

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

$$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$$
$$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$$
$$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$$
$$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$$
$$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$$
$$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$$

$D = 4$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$
f_1	0	1	0	1	0	1	0	0	1	1	0					
f_2	0	0	1	1	1	0	1	1	0	1	0					
f_3	0	0	0	1	0	1	0	1	1	0	1					
f_4	1	1	0	1	1	0	0	0	1	1	1					
f_5	1	0	1	1	1	0	0	0	1	0	0					
f_6	0	1	1	1	0	0	1	1	1	1	0					
x_1f_1																
x_2f_1																
… $x_1x_2f_1$																

$x_1x_3f_1$

Gröbner basis

Gröbner basis

- Let $R = \mathbb{F}_q[x_1, \ldots, x_n]$ be the polynomial ring in n variables.

Gröbner basis

- Let $R = \mathbb{F}_q[x_1, \ldots, x_n]$ be the polynomial ring in n variables.
- An ideal in R is an additive subgroup I such that if $g \in R$ and $f \in I$, then $gf \in I$.

Gröbner basis

- Let $R = \mathbb{F}_q[x_1, \ldots, x_n]$ be the polynomial ring in n variables.
- An ideal in R is an additive subgroup I such that if $g \in R$ and $f \in I$, then $gf \in I$.
- The subset $\{f_1, \ldots, f_m\} \subset R$ is a set of generators for an ideal I if every element $t \in I$ can be written in the form $t = \sum_1^n$ with $g_i \in R$.

Gröbner basis

- Let $R = \mathbb{F}_q[x_1, \ldots, x_n]$ be the polynomial ring in n variables.

- An ideal in R is an additive subgroup I such that if $g \in R$ and $f \in I$, then $gf \in I$.

- The subset $\{f_1, \ldots, f_m\} \subset R$ is a set of generators for an ideal I if every element $t \in I$ can be written in the form $t = \sum_{1}^{n}$ with $g_i \in R$.

- By the Hilbert basis theorem: every ideal in R has a finite set of generators.

Gröbner basis

- Let $R = \mathbb{F}_q[x_1, \ldots, x_n]$ be the polynomial ring in n variables.

- An ideal in R is an additive subgroup I such that if $g \in R$ and $f \in I$, then $gf \in I$.

- The subset $\{f_1, \ldots, f_m\} \subset R$ is a set of generators for an ideal I if every element $t \in I$ can be written in the form $t = \sum_{1}^{n}$ with $g_i \in R$.

- By the Hilbert basis theorem: every ideal in R has a finite set of generators.

- The subset of R defined as $V(I) = \{(a_1, \ldots, a_n) \in \mathbb{F}_q^n \,|\, f(a_1, \ldots, a_n) = 0 \text{ for all } f \in I\}$ is called an algebraic variety. It is the set of all solutions to the system of equations $f_1(x_1, \ldots, x_n) = \ldots = f_1(x_1, \ldots, x_n) = 0.$

Gröbner basis

- Let $R = \mathbb{F}_q[x_1, \ldots, x_n]$ be the polynomial ring in n variables.

- An ideal in R is an additive subgroup I such that if $g \in R$ and $f \in I$, then $gf \in I$.

- The subset $\{f_1, \ldots, f_m\} \subset R$ is a set of generators for an ideal I if every element $t \in I$ can be written in the form $t = \sum_{1}^{n}$ with $g_i \in R$.

- By the Hilbert basis theorem: every ideal in R has a finite set of generators.

- The subset of R defined as $V(I) = \{(a_1, \ldots, a_n) \in \mathbb{F}_q^n \,|\, f(a_1, \ldots, a_n) = 0 \text{ for all } f \in I\}$ is called an algebraic variety. It is the set of all solutions to the system of equations $f_1(x_1, \ldots, x_n) = \ldots = f_1(x_1, \ldots, x_n) = 0$.

- By the Nullstellensatz: $\mathbf{I}(V(I)) = I$, where $\mathbf{I}(V)$ denotes the ideal of V, i.e. $\mathbf{I}(V) = \{f \in R \,|\, f(a) = 0 \text{ for all } a \in V\}$ (Similar to Gauss' fundamental theorem, but for polynomials in many variables).

Gröbner basis

- A Gröbner basis of an ideal I is a set of generators with some nice (useful) property.

Gröbner basis

- A Gröbner basis of an ideal I is a set of generators with some nice (useful) property.

  ↪ For our case, the nice property is that a solution can be extracted easily from the Gröbner basis.

Gröbner basis

- A Gröbner basis of an ideal I is a set of generators with some nice (useful) property.

  ↳ For our case, the nice property is that a solution can be extracted easily from the Gröbner basis.

Example. The shape of a GB with respect to the lexicographic order

$$f_1 : x_1x_3 + x_1 + x_2x_4 + x_5 + x_6 + 1 = 0$$
$$f_2 : x_1x_4 + x_1 + x_2x_3 + x_2 + x_3x_4 + x_3x_6 + x_4 + x_5 = 0$$
$$f_3 : x_1x_5 + x_1 + x_2 + x_3x_4 + x_6 + 1 = 0$$
$$f_4 : x_1x_2 + x_1x_3 + x_2x_5 + x_3 + x_4 + x_6 + 1 = 0$$
$$f_5 : x_1x_4 + x_2x_3 + x_2x_5 + x_5x_6 + 1 = 0$$
$$f_6 : x_1x_3 + x_1x_4 + x_1 + x_2 + x_3x_6 + x_3 + x_5 = 0$$

Gröbner basis

- A Gröbner basis of an ideal I is a set of generators with some nice (useful) property.

  For our case, the nice property is that a solution can be extracted easily from the Gröbner basis.

Example. The shape of a GB with respect to the lexicographic order

$$
\begin{aligned}
f_1 &: x_1x_3 + x_1 + x_2x_4 + x_5 + x_6 + 1 = 0\\
f_2 &: x_1x_4 + x_1 + x_2x_3 + x_2 + x_3x_4 + x_3x_6 + x_4 + x_5 = 0\\
f_3 &: x_1x_5 + x_1 + x_2 + x_3x_4 + x_6 + 1 = 0\\
f_4 &: x_1x_2 + x_1x_3 + x_2x_5 + x_3 + x_4 + x_6 + 1 = 0\\
f_5 &: x_1x_4 + x_2x_3 + x_2x_5 + x_5x_6 + 1 = 0\\
f_6 &: x_1x_3 + x_1x_4 + x_1 + x_2 + x_3x_6 + x_3 + x_5 = 0
\end{aligned}
\longrightarrow
\begin{aligned}
f'_1 &: x_1 + x_6 = 0\\
f'_2 &: x_2 + x_6 = 0\\
f'_3 &: x_3 + x_6 = 0\\
f'_4 &: x_4 + x_6 + 1 = 0\\
f'_5 &: x_5 = 0
\end{aligned}
$$

*****
****
***
**
*

Gröbner basis

- A Gröbner basis of an ideal I is a set of generators with some nice (useful) property.

  ↳ For our case, the nice property is that a solution can be extracted easily from the Gröbner basis.

Example. The shape of a GB with respect to the lexicographic order

$$
\begin{aligned}
f_1 &: x_1x_3 + x_1 + x_2x_4 + x_5 + x_6 + 1 = 0 \\
f_2 &: x_1x_4 + x_1 + x_2x_3 + x_2 + x_3x_4 + x_3x_6 + x_4 + x_5 = 0 \\
f_3 &: x_1x_5 + x_1 + x_2 + x_3x_4 + x_6 + 1 = 0 \\
f_4 &: x_1x_2 + x_1x_3 + x_2x_5 + x_3 + x_4 + x_6 + 1 = 0 \\
f_5 &: x_1x_4 + x_2x_3 + x_2x_5 + x_5x_6 + 1 = 0 \\
f_6 &: x_1x_3 + x_1x_4 + x_1 + x_2 + x_3x_6 + x_3 + x_5 = 0
\end{aligned}
\quad \longrightarrow \quad
\begin{aligned}
f_1' &: x_1 + x_6 = 0 \\
f_2' &: x_2 + x_6 = 0 \\
f_3' &: x_3 + x_6 = 0 \\
f_4' &: x_4 + x_6 + 1 = 0 \\
f_5' &: x_5 = 0
\end{aligned}
$$

*****
****
***
**
*

$$V(< f_1, \ldots, f_6 >) = \{(0,0,0,1,0,0), (1,1,1,0,0,1)\}$$

Gröbner basis algorithms:

Buchberger, Lazard, F4, F5

↪ Follow the core idea that we described, but combine the equations in an organised way, rather than multiplying them by all possible monomials.

Not covered in this talk:

- Monomial orders
- S-polynomials
- Polynomial long division
- Row reduction in parallel
- Reductions to zero
- Syzygy criterion
- …

XL/Gröbner basis algorithms: complexity

XL/Gröbner basis algorithms: complexity

$$\mathcal{O}\left(mD_{reg}\binom{n+D_{reg}-1}{D_{reg}}^{\omega}\right)$$

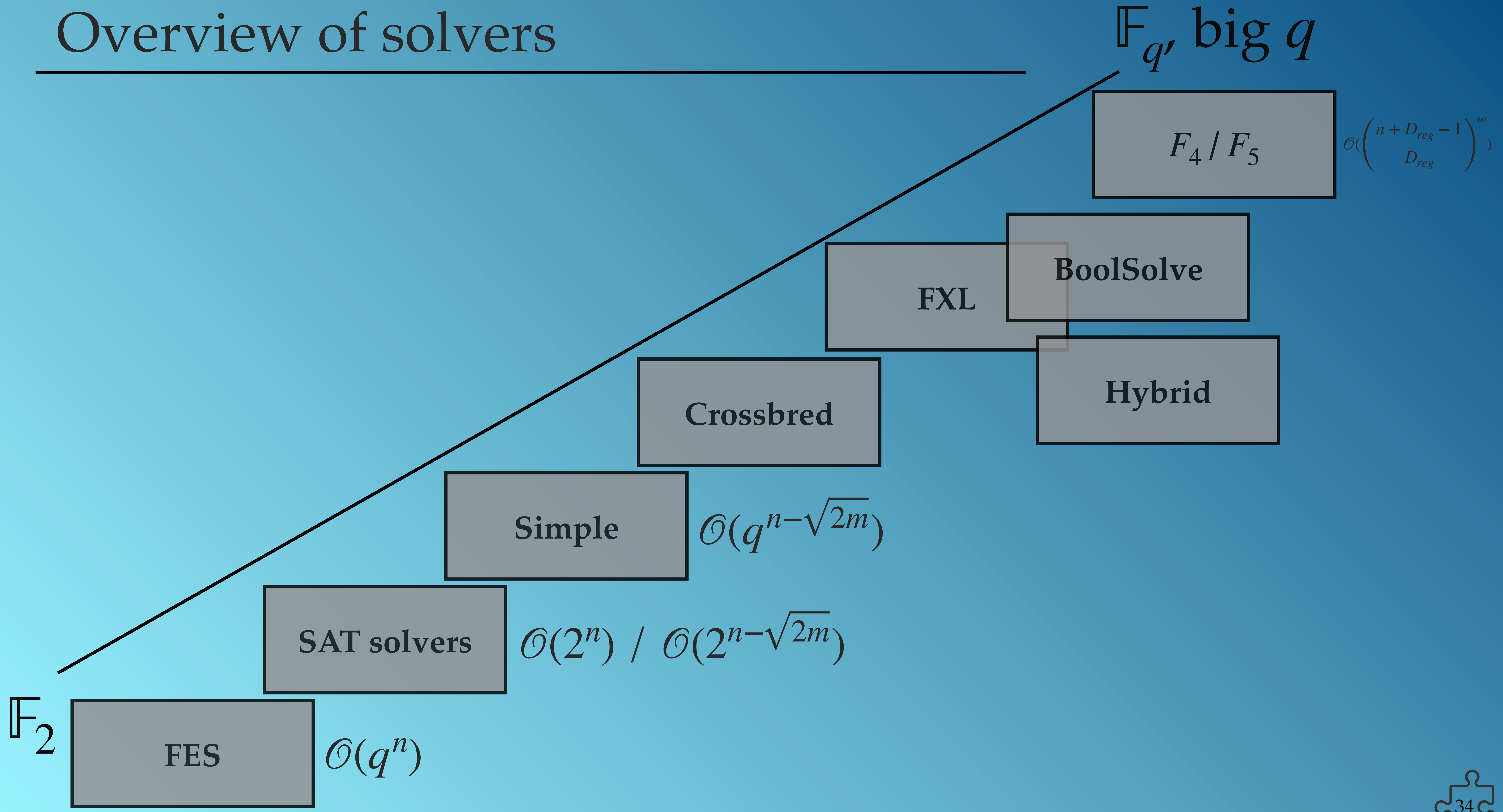
XL/Gröbner basis algorithms: complexity

$$\mathcal{O}\left(mD_{reg} \binom{n + D_{reg} - 1}{D_{reg}}^{\omega} \right)$$

D_{reg}: degree of regularity

↪ the power of the first non-positive coefficient in the expansion of $$\frac{(1-t^2)^m}{(1-t)^n}$$

Overview of solvers



FXL

[Courtois, Klimov, Patarin, Shamir, 2000]

Hybrid

[Bettale, Faugère, Perret, 2009]

BoolSolve

[Bardet, Faugère, Salvy, Spaenlehauer, 2013]

FXL, Hybrid, BoolSolve

Techniques are already covered in the previous section.

Algorithms will be explained in the summary.

The crossbred algorithm

[Joux, Vitse, 2017]

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

→ Put matrix in reduced row echelon form

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1

…

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

→ Take linear subsystem

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1

…

…if we had another 4 equations

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1

…

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

→ Subsystem is linear in variables $\{x_1, x_2, x_3\}$.

→ Enumerating x_4 will result in a linear subsystem.

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1

…

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1

$\dots$

Crossbred algorithm

→ Subsystem can be linearised

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1

…

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$

$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$

$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$

$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$

$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$

$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

Crossbred algorithm

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

→ Subsystem can be linearised

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1
…											

…if we had another 4 equations, the subsystem would have a unique solution.

Otherwise: check candidate solutions against the other equations.

Crossbred algorithm

Parameters of the algorithm: D, k, d, h

- → Enumerate h variables.
- → Choose k of the remaining variables.
- → Augment system up to degree D (compute degree-D Macaulay matrix).
- → Take the subsystem that is at most degree d in the k chosen variables.
- → Enumerate all but the k chosen variables.
- → Linearise the subsystem and solve it.
- → Check if candidate solutions are consistent with the rest of the system.

Crossbred algorithm

Parameters of the algorithm: D, k, d, h

- Enumerate h variables.
- Choose k of the remaining variables.
- Augment system up to degree D (compute degree-D Macaulay matrix).
- Take the subsystem that is at most degree d in the k chosen variables.
- Enumerate all but the k chosen variables.
- Linearise the subsystem and solve it.
- Check if candidate solutions are consistent with the rest of the system.

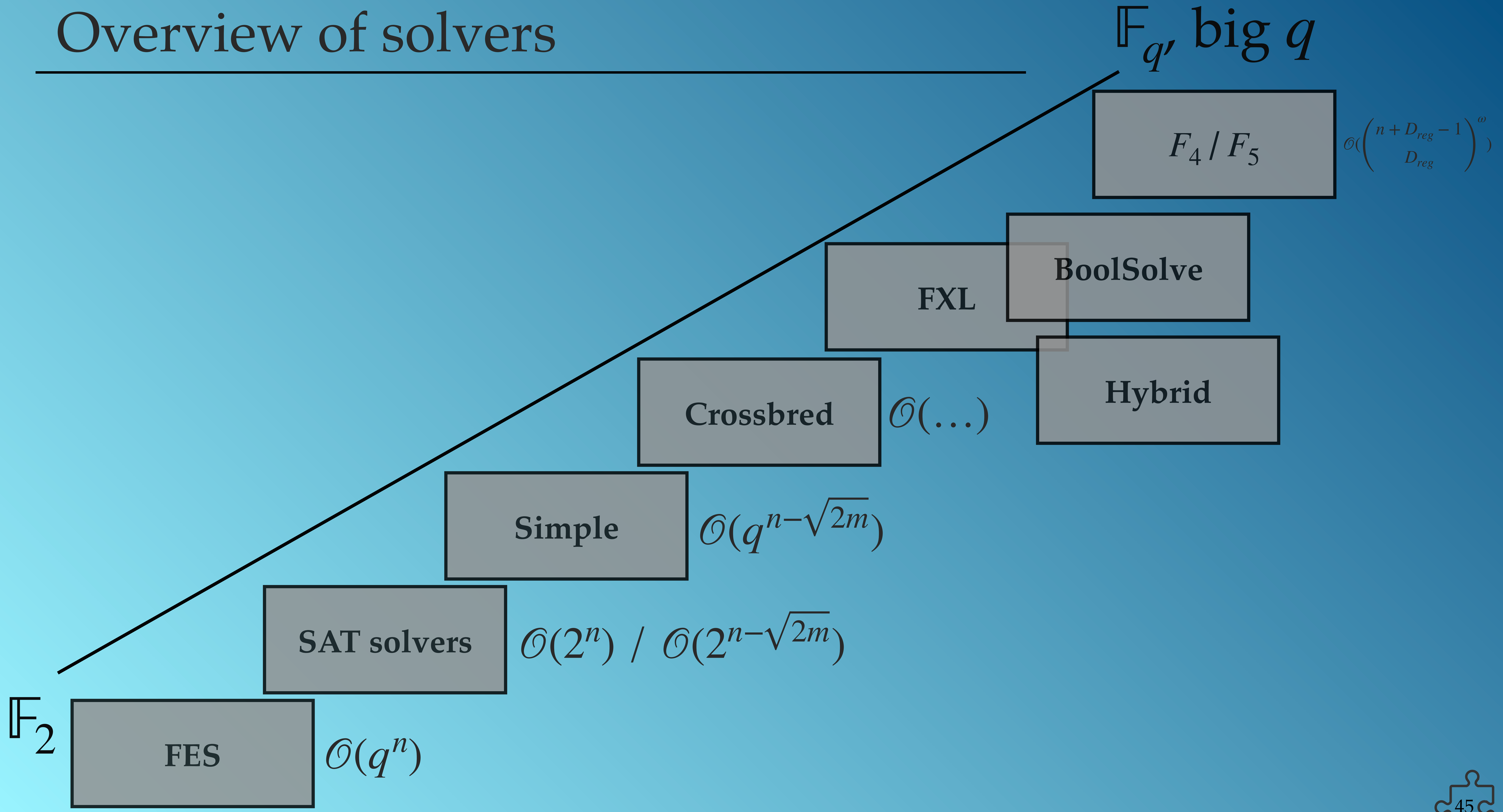
The complexity is calculated as the best trade-off between the four parameters.

Crossbred algorithm

	Number of Variables (n)	Seed (0,1,2,3,4)	Date	Contestants	Computational Resource	Data
1	83	0	2023/09/16	Charles Bouillaguet and Julia Sauvage	https://gitlab.lip6.fr/almasty/hpXbred, 3488 AMD EPYC 7J13 cores on the Oracle public cloud	Details
6	74	0	2016/12/17	Antoine Joux	New hybridized XL related algorithm, Heterogeneous cluster of Intel Xeon @ 2.7-3.5 Ghz	Details
7	74	4	2017/11/15	Kai-Chun Ning, Ruben Niederhagen	Parallel Crossbred, 54 GPUs in the Saber cluster	Details
25	66	0	2016/01/22	Tung Chou, Ruben Niederhagen, Bo-Yin Yang	Gray Code enumeration, Rivyera, 128 Spartan 6 FPGAs	Details

Fukuoka MQ challenge record computations ($m = 2n$)

Overview of solvers



Summary

- (Partial) enumeration
- Candidate solutions (subsystem)
- Conflict search
- Extending to higher degrees
- Computing a Gröbner Basis

- **FES**
- **Simple**
- **FXL**
- F_4 / F_5
- **SAT solvers**
- **Crossbred**
- **BoolSolve**
- **Hybrid**

Summary

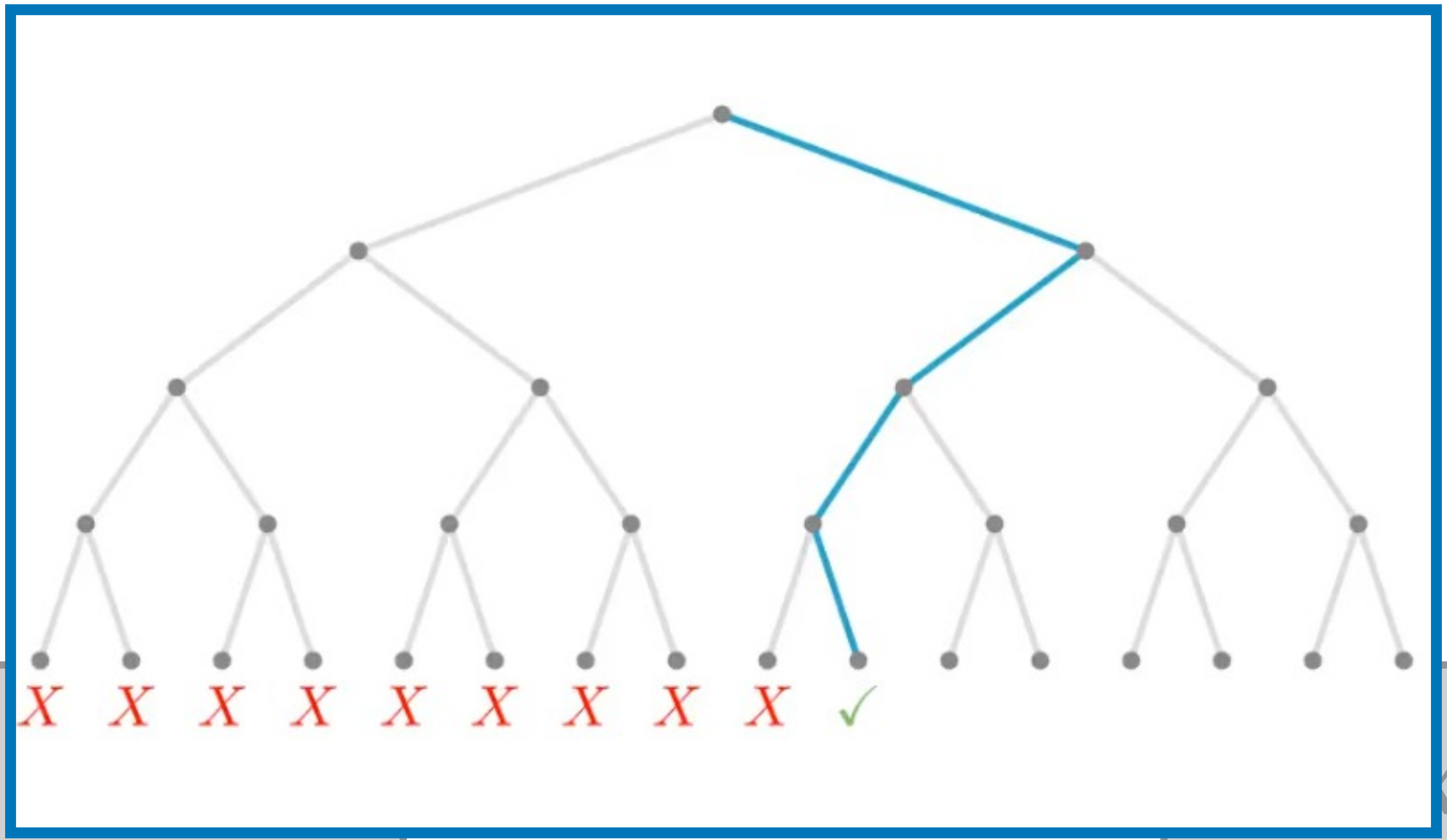
(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis



FES

XL

F_4 / F_5

SAT solvers

Crossbred

BoolSolve

Hybrid

Summary

(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis

FES

Simple

FXL

F_4 / F_5

SAT solvers

Crossbred

BoolSolve

Hybrid

Summary

(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis

	x_1x_2	x_1x_3	x_2x_3	x_1x_4	x_2x_4	x_3x_4	x_1	x_2	x_3	x_4	1
f_1	1	0	0	0	0	0	0	0	0	1	1
f_2	0	1	0	0	0	0	1	1	1	1	0
f_3	0	0	1	0	0	0	1	1	0	1	0
f_4	0	0	0	1	0	0	1	1	1	0	1
f_5	0	0	0	0	1	0	0	1	0	0	0
f_6	0	0	0	0	0	1	1	1	1	0	1
…											

FES

F_4 / F_5

SAT solvers

Crossbred

BoolSolve

Hybrid

Summary

(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis

FES

Simple

FXL

F_4 / F_5

SAT solvers

Crossbred

BoolSolve

Hybrid

Summary

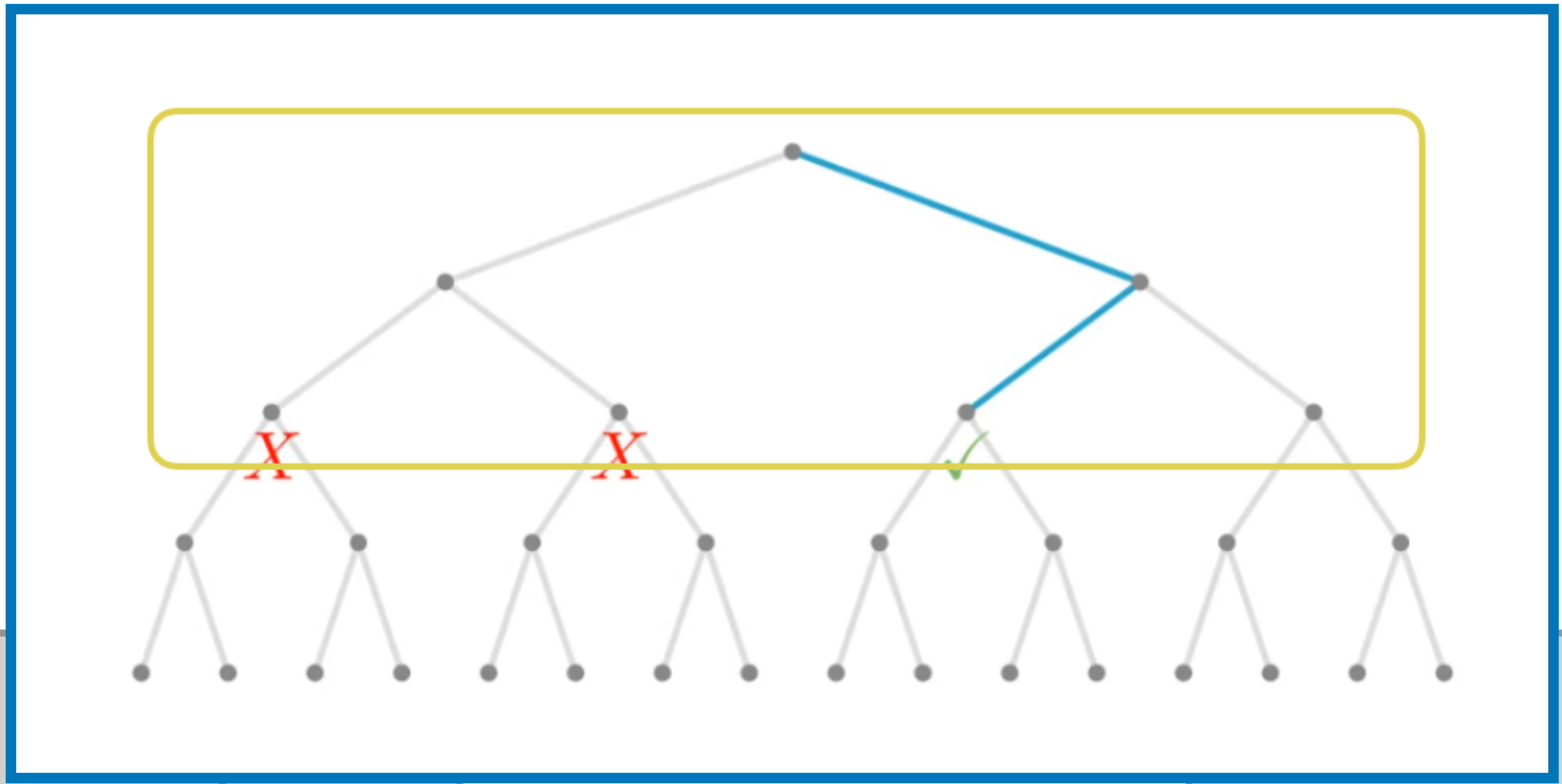
(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis



FES

F_4 / F_5

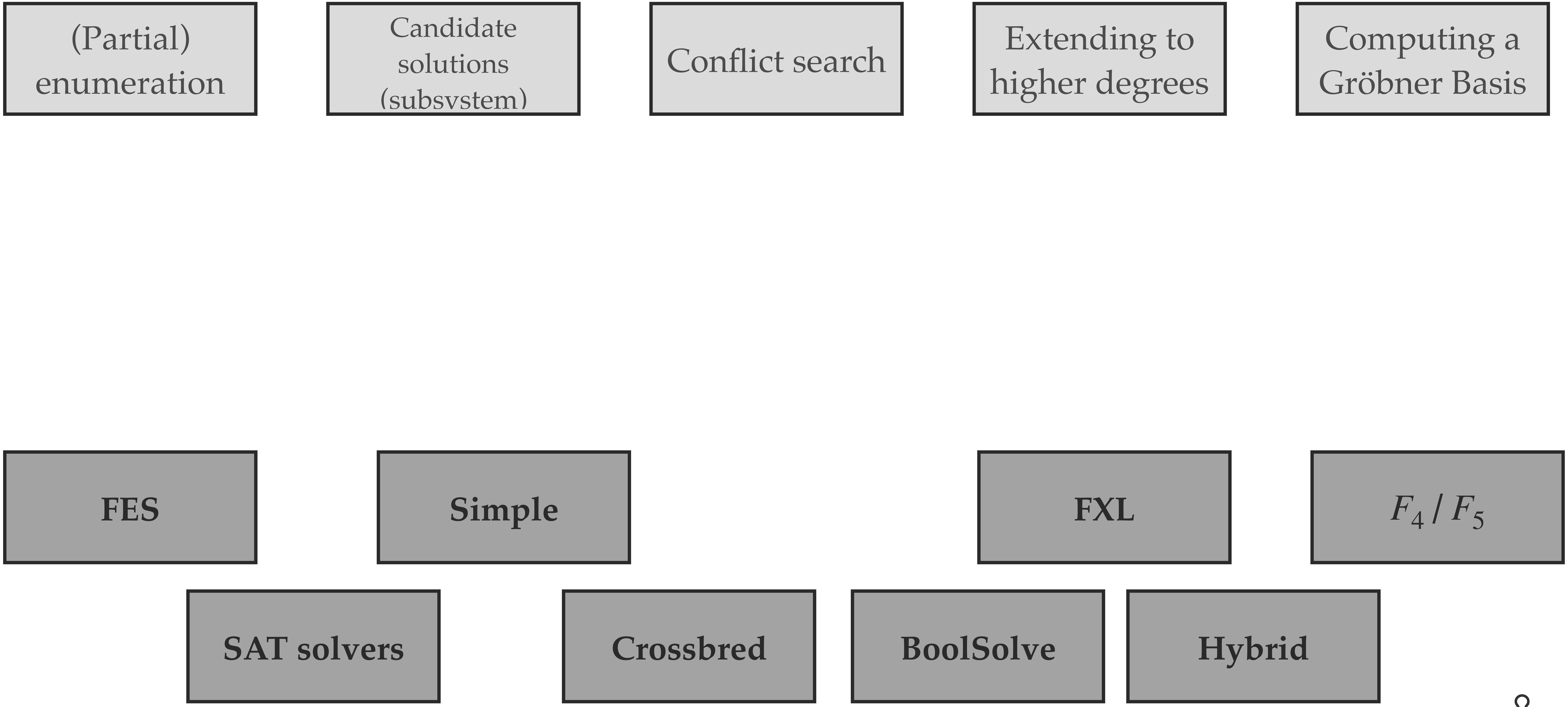
SAT solvers

Crossbred

BoolSolve

Hybrid

Summary


(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

Summary

(Partial) enumeration

Candidate solutions

Conflict search

Extending to higher degrees

Computing a Gröbner Basis

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$
f_1	0	1	0	1	0	1	0	0	1	1	0					
f_2	0	0	1	1	1	0	1	1	0	1	0					
f_3	0	0	0	1	0	1	0	1	1	0	1					
f_4	1	1	0	1	1	0	0	0	1	1	1					
f_5	1	0	1	1	1	0	0	0	1	0	0					
f_6	0	1	1	1	0	0	1	1	1	1	0					
x_1f_1																
x_2f_1																
…																
$x_1x_2f_1$																
$x_1x_3f_1$																

FES

F_4 / F_5

SAT solvers

Crossbred

BoolSolve

Hybrid

Summary

(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis

FES

Simple

FXL

F_4 / F_5

SAT solvers

Crossbred

BoolSolve

Hybrid

Summary

(Partial) enumeration

Candidate solutions (subsystem)

Conflict search

Extending to higher degrees

Computing a Gröbner Basis

$$f_1' : x_1 + x_6 = 0$$
$$f_2' : x_2 + x_6 = 0$$
$$f_3' : x_3 + x_6 = 0$$
$$f_4' : x_4 + x_6 + 1 = 0$$
$$f_5' : x_5 = 0$$

FES

Simple

FXL

F_4 / F_5

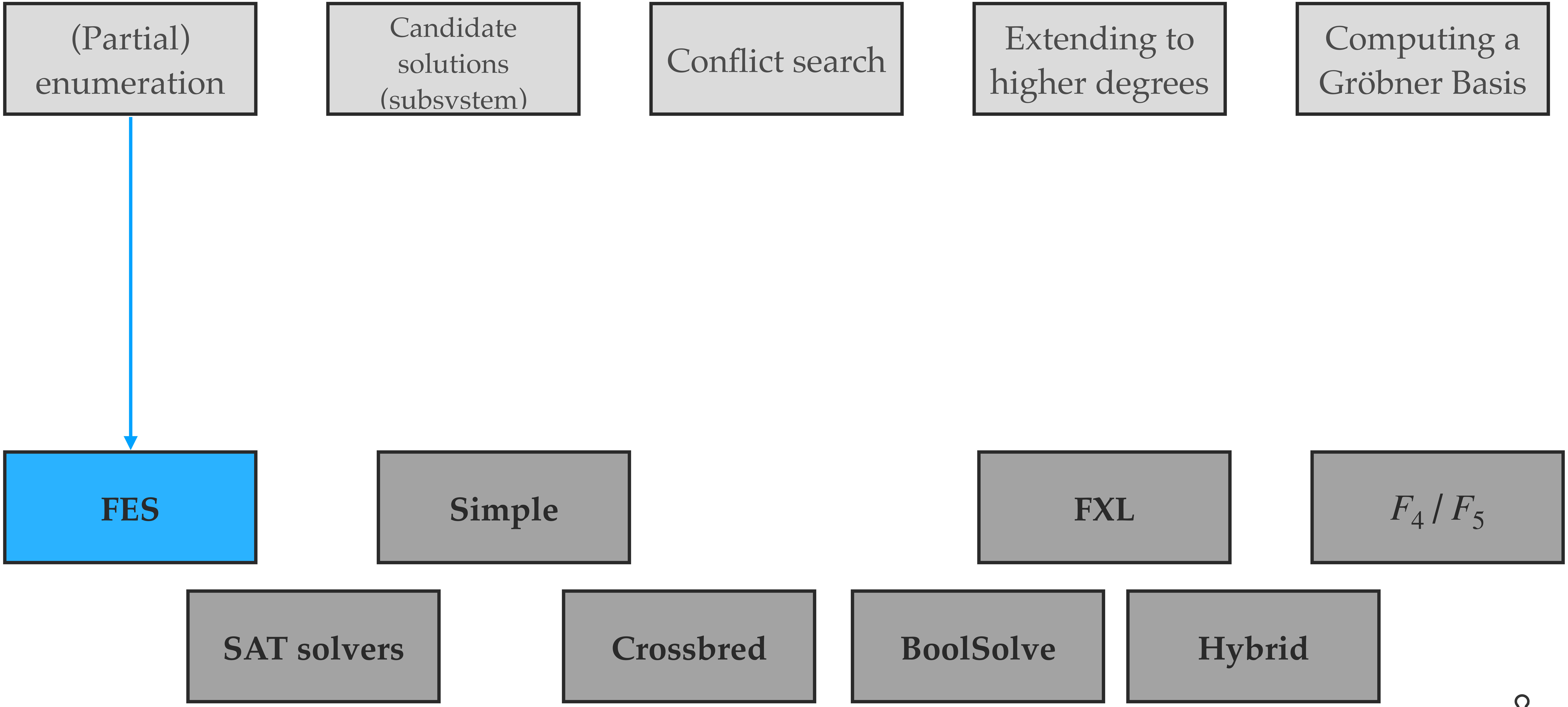
SAT solvers

Crossbred

BoolSolve

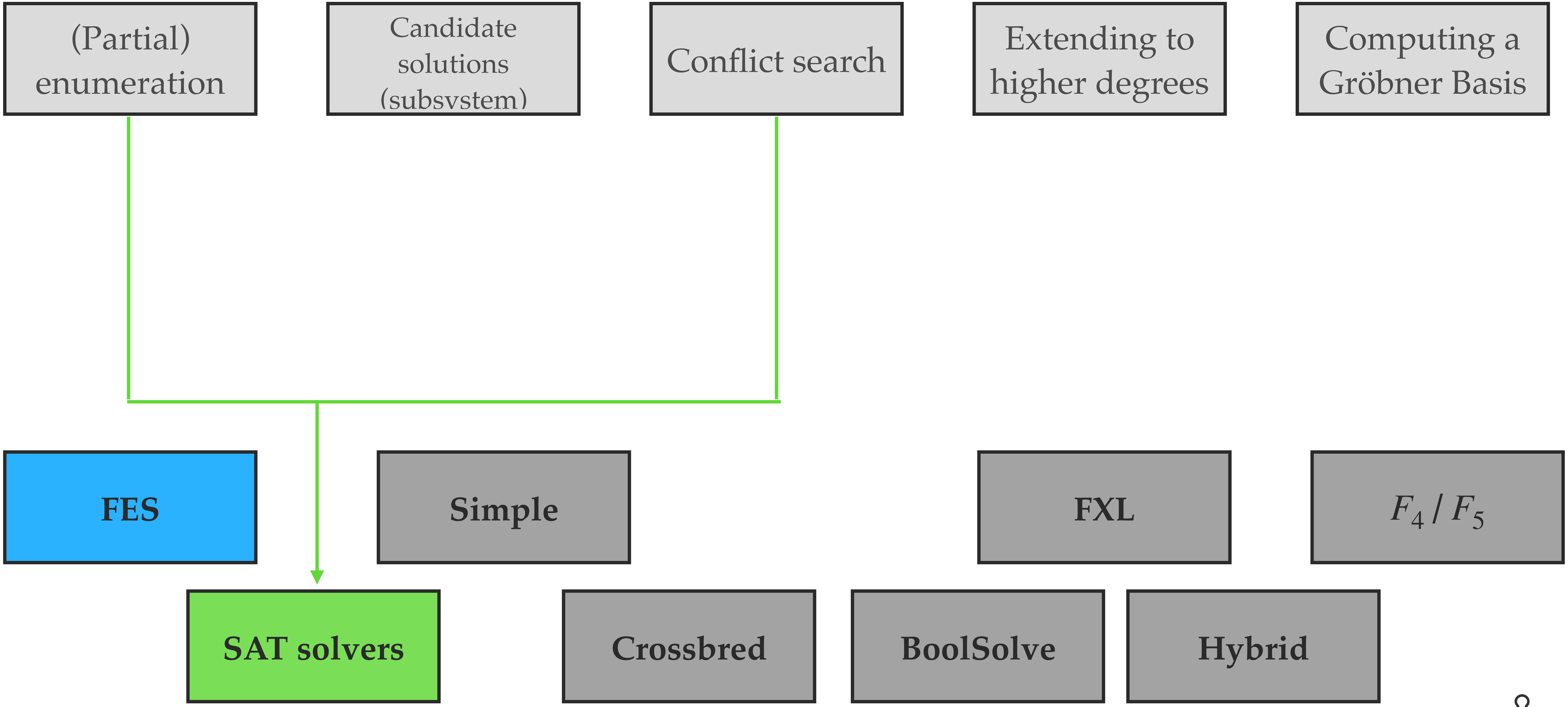
Hybrid

Summary



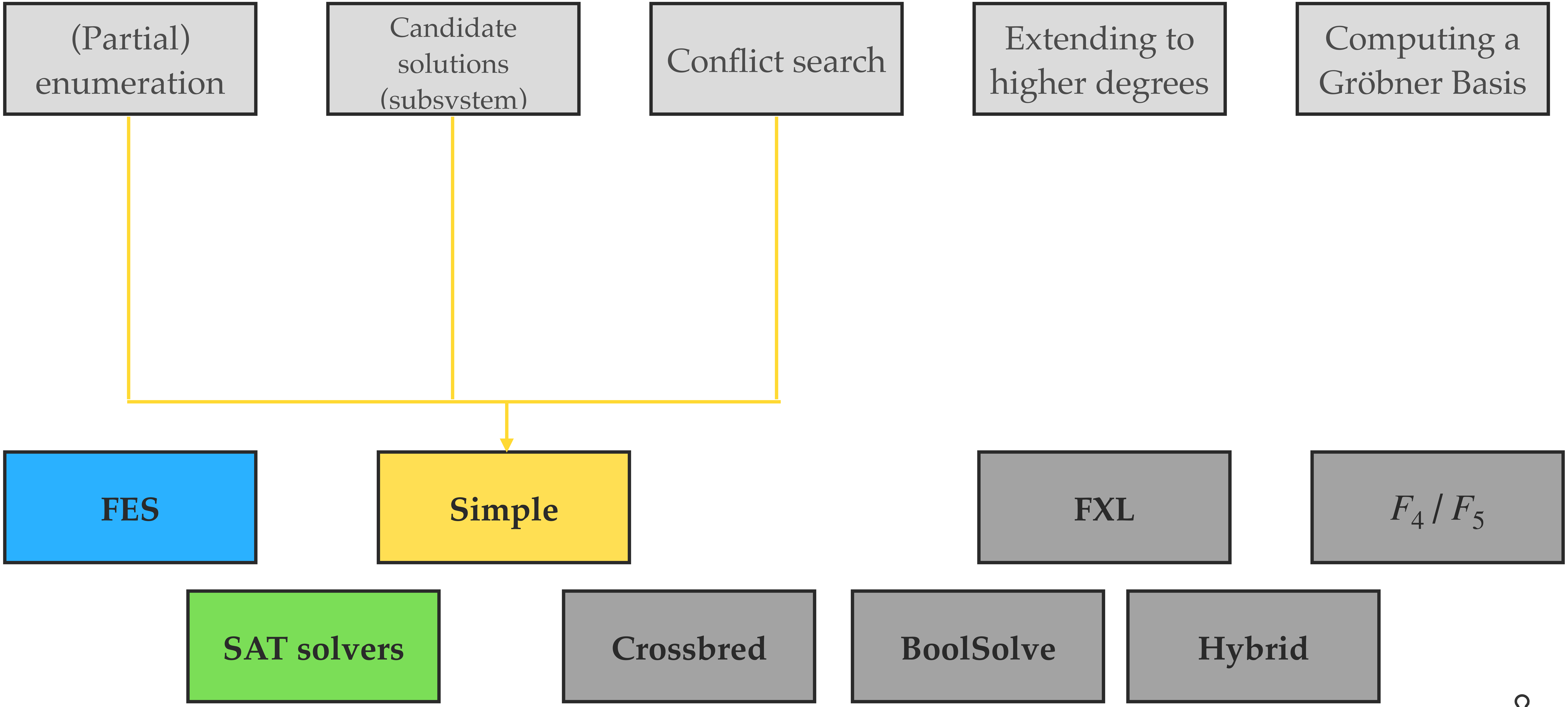
(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

Summary

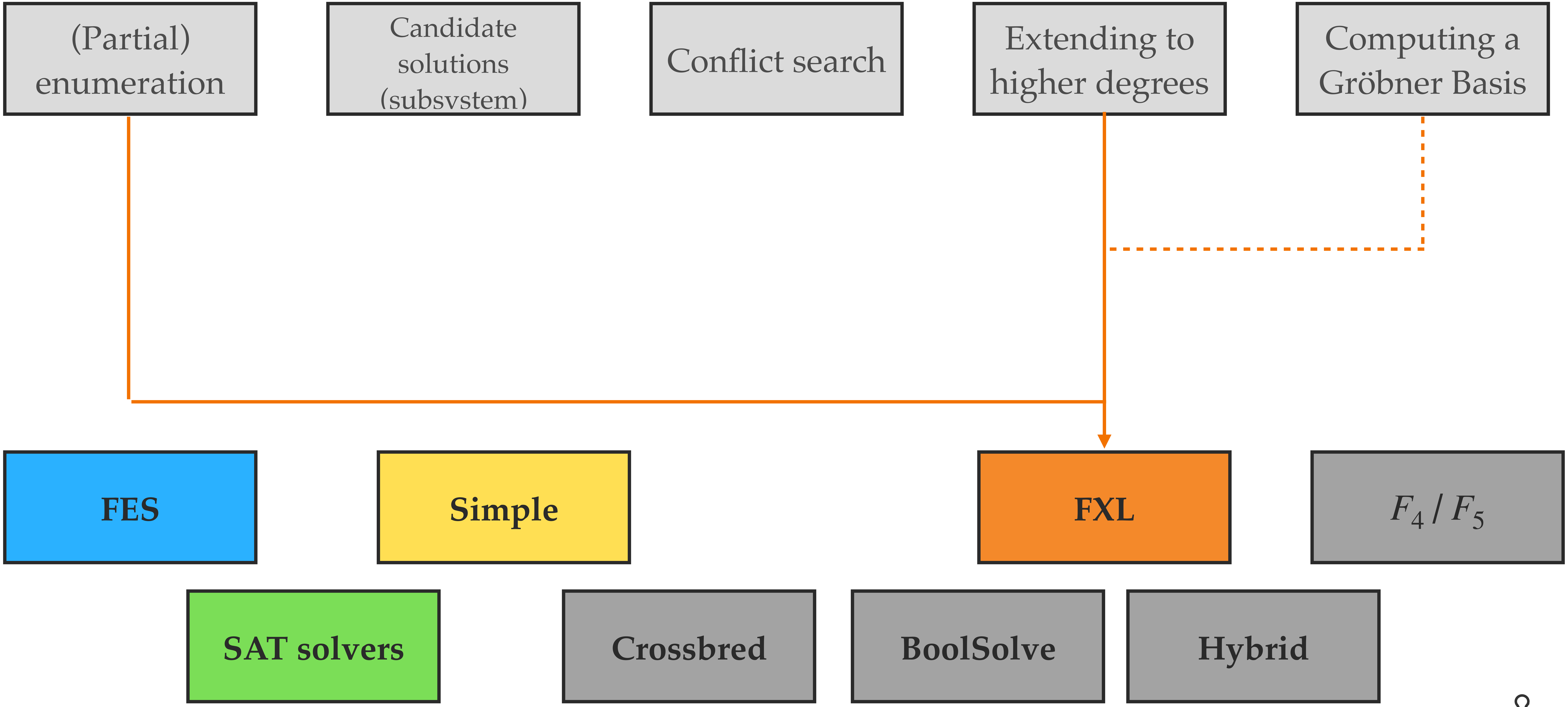


(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

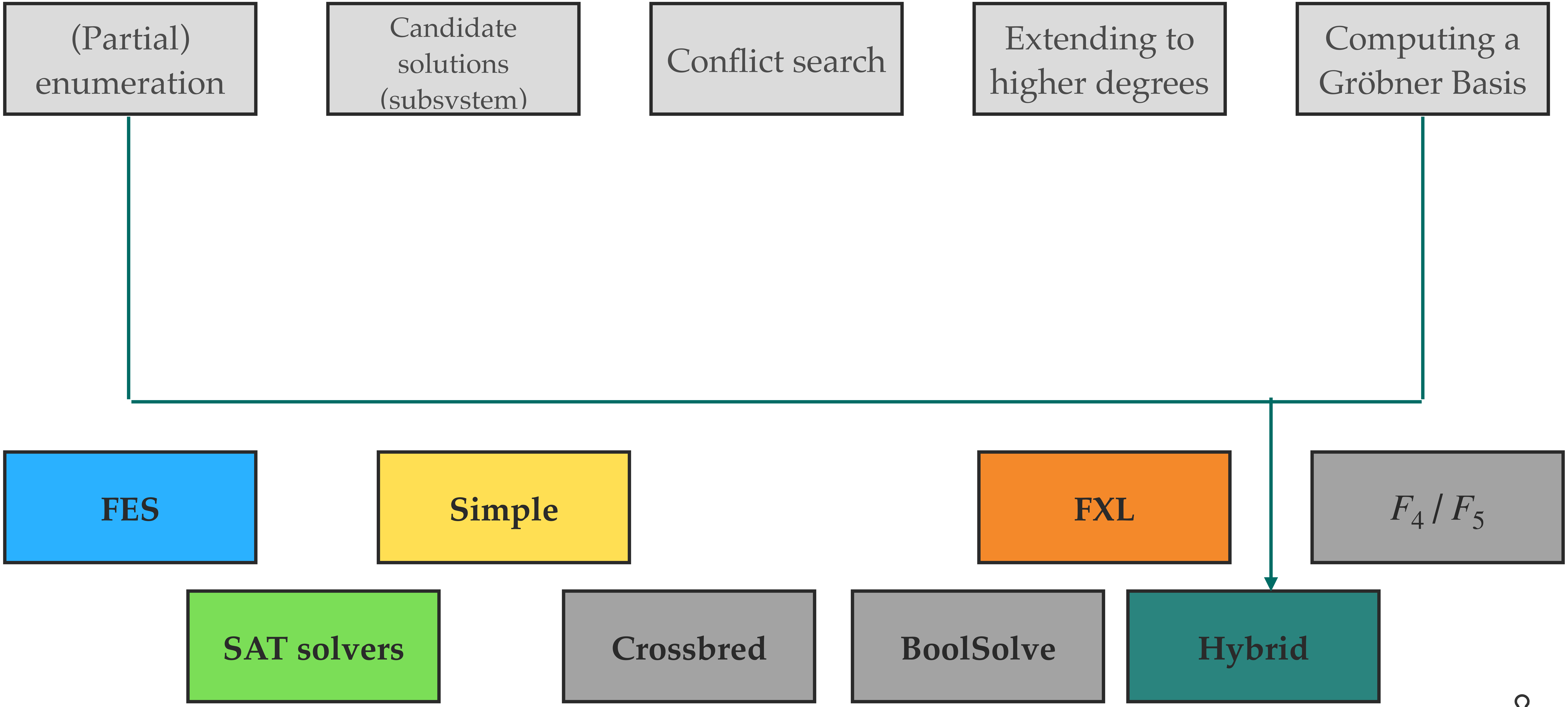
Summary


(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

Summary

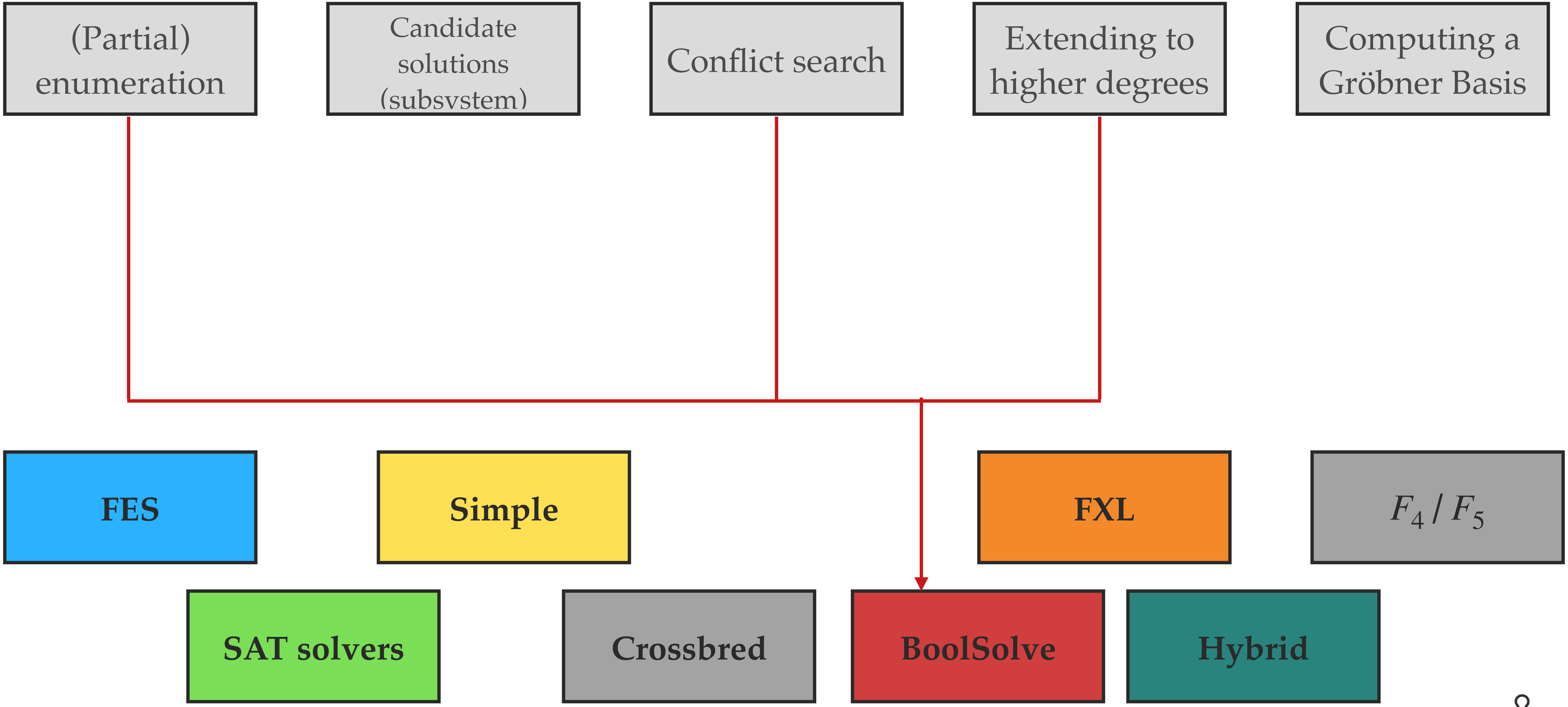

(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

Summary

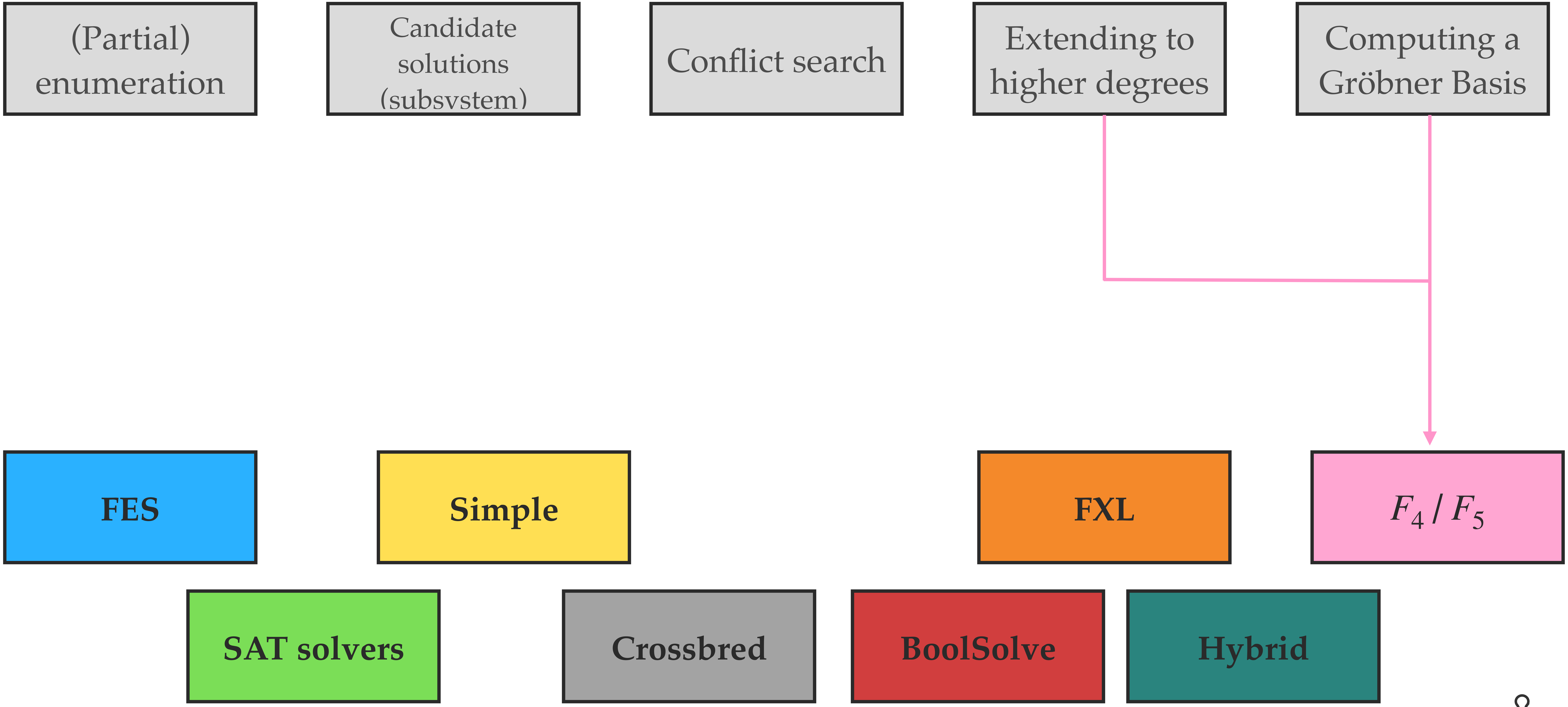


(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

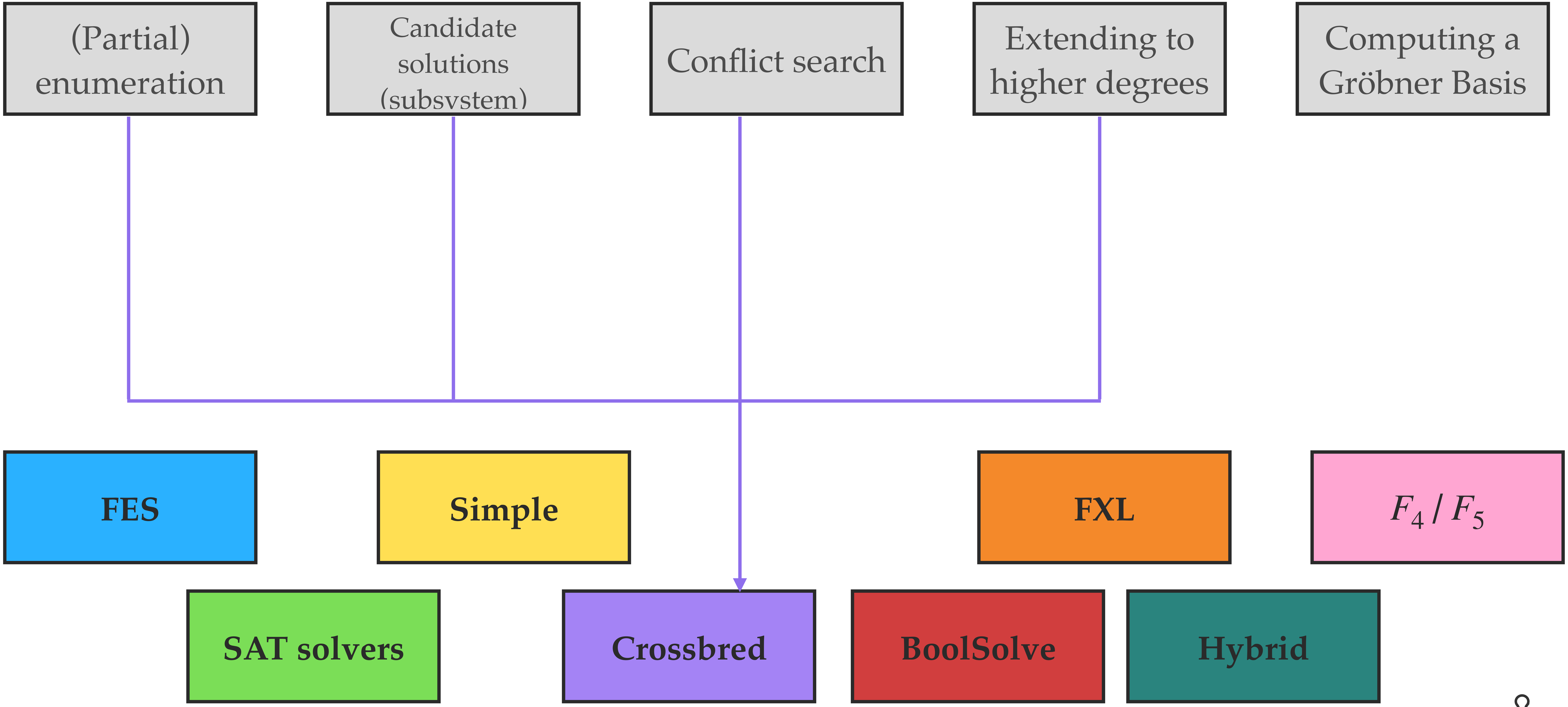
Summary


(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

Summary

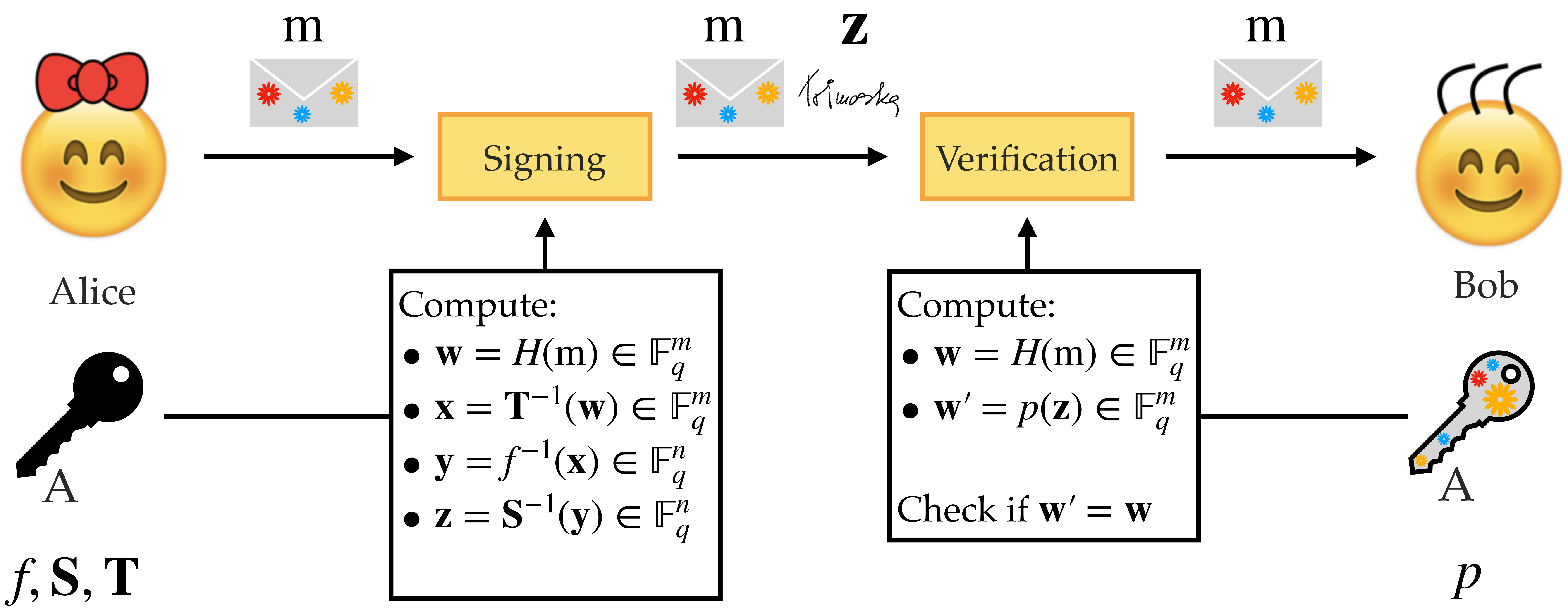

(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

Summary


(Partial) enumeration
Candidate solutions (subsystem)
Conflict search
Extending to higher degrees
Computing a Gröbner Basis
FES
Simple
FXL
F_4 / F_5
SAT solvers
Crossbred
BoolSolve
Hybrid

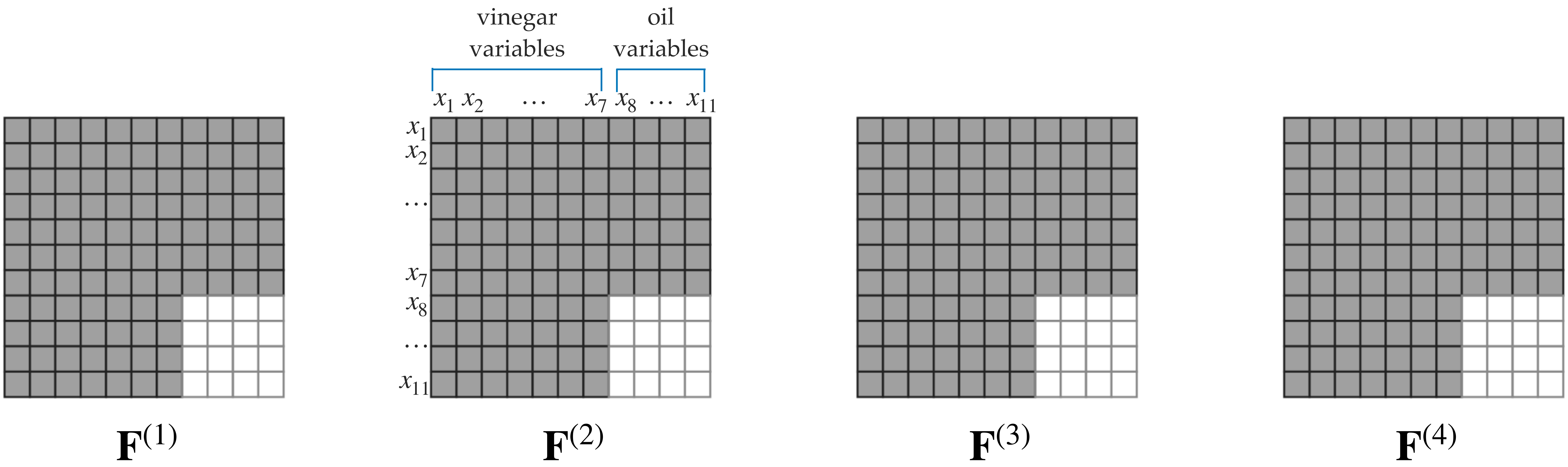
Modelisation: Attacks on UOV

The trapdoor construction (recall)



The UOV central map (recall)

Toy example: $v = 7, m = 4$



*Grayed areas represent the entries that are possibly nonzero; blank areas denote the zero entries;

Attacks on UOV

- Direct attack
- Reconciliation attack
- Kipnis-Shamir attack
- Intersection attack

Direct attack

Direct attack

- Try to forge a signature with only the knowledge of the public key.

Direct attack

Try to forge a signature with only the knowledge of the public key.

Constraint for modelisation

For a target $\mathbf{w}$, find $\mathbf{z}$ such that $p(\mathbf{z}) = \mathbf{w}$.

Direct attack

Try to forge a signature with only the knowledge of the public key.

Constraint for modelisation

For a target $\mathbf{w}$, find $\mathbf{z}$ such that $p(\mathbf{z}) = \mathbf{w}$.

Equations:

$$
\begin{aligned}
\mathbf{z}^\top \mathbf{P}^{(1)} \mathbf{z} &= w_1 \\
\mathbf{z}^\top \mathbf{P}^{(2)} \mathbf{z} &= w_2 \\
&\dots \\
\mathbf{z}^\top \mathbf{P}^{(m)} \mathbf{z} &= w_m
\end{aligned}
$$

Direct attack

Try to forge a signature with only the knowledge of the public key.

Constraint for modelisation

For a target $\mathbf{w}$, find $\mathbf{z}$ such that $p(\mathbf{z}) = \mathbf{w}$.

→ Equations:

$$\mathbf{z}^\top \mathbf{P}^{(1)} \mathbf{z} = w_1$$
$$\mathbf{z}^\top \mathbf{P}^{(2)} \mathbf{z} = w_1$$
$$\dots$$
$$\mathbf{z}^\top \mathbf{P}^{(m)} \mathbf{z} = w_m$$

Reconciliation attack

[Ding, Yang, Chen, Chen, Cheng, 2008]

The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

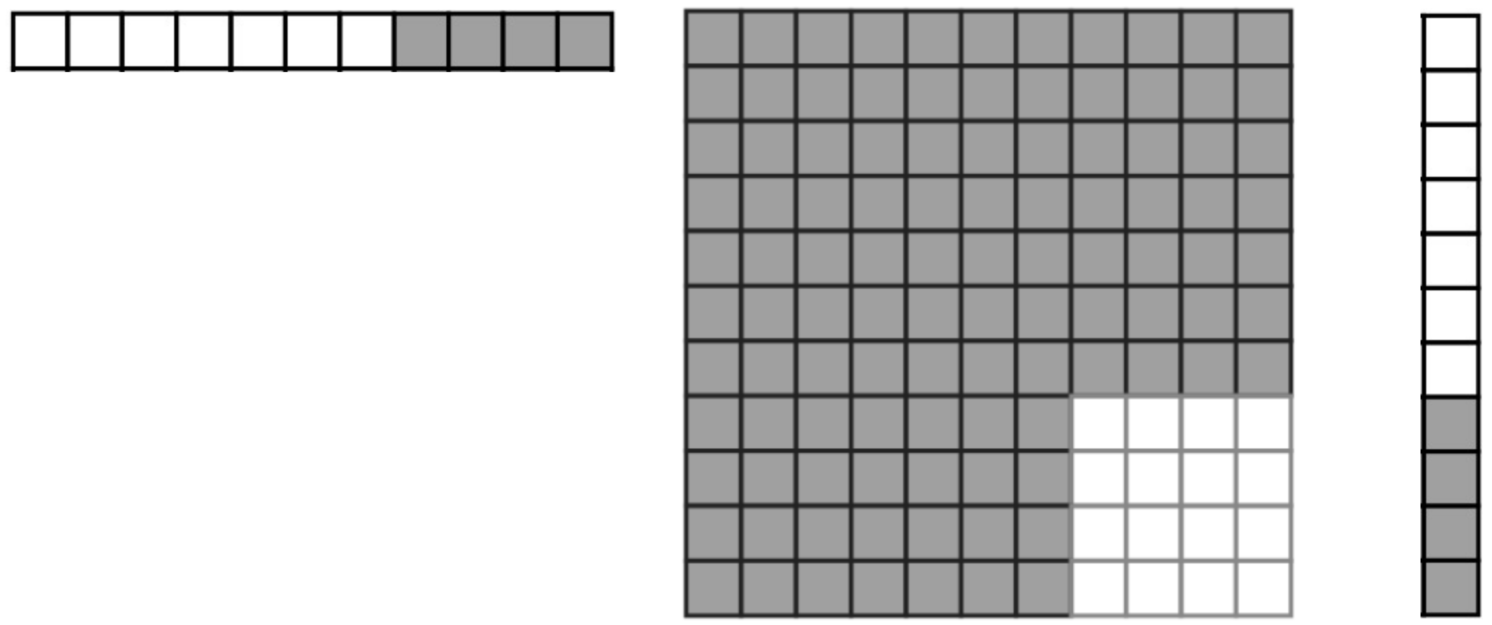
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n - m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \,|\, v_i = 0 \text{ for all } i \leq n - m\}$.



$= 0$

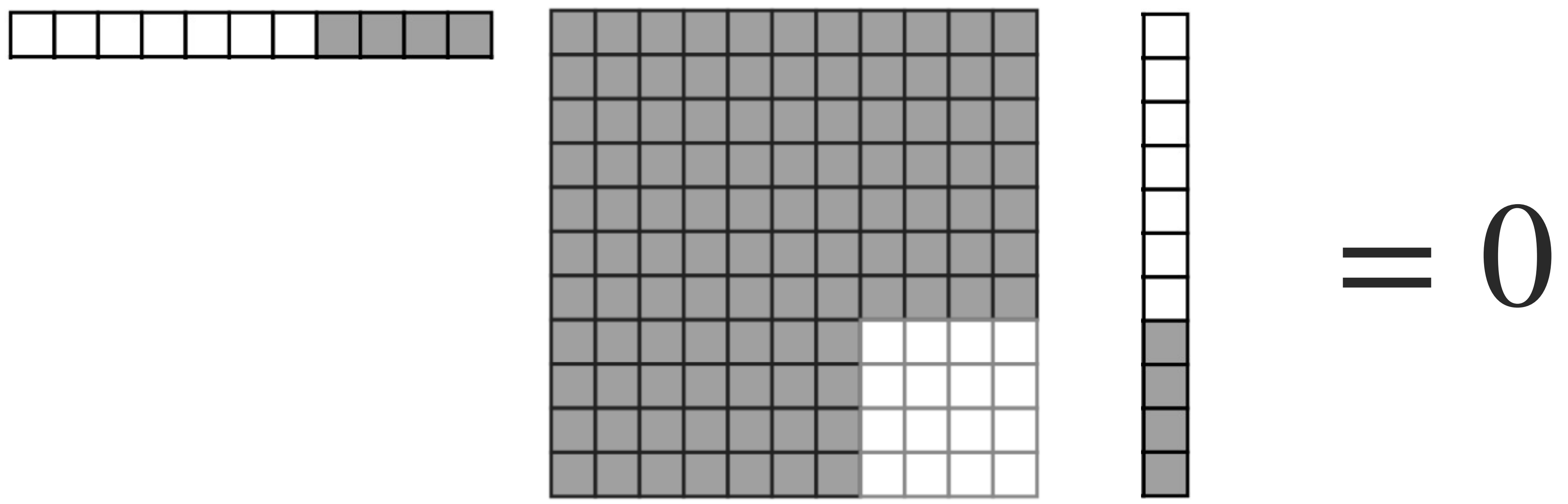
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n - m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n - m\}$.



f vanishes on O'.

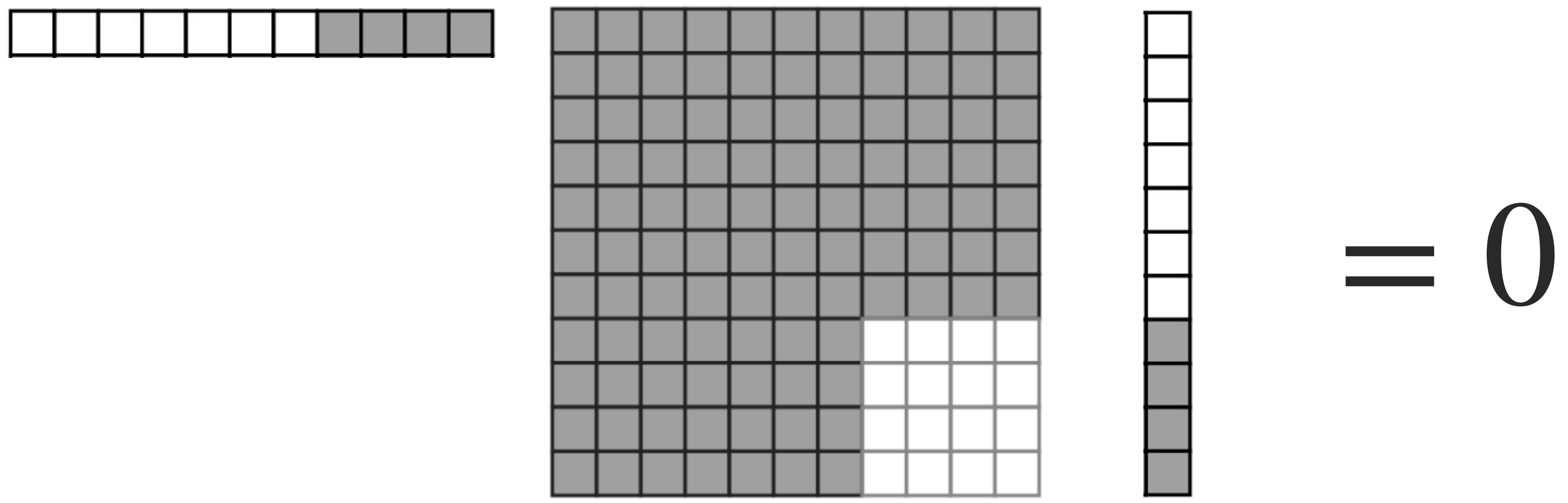
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n-m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n-m\}$.



f vanishes on O'.

Let $O = \mathbf{S}^{-1}(O')$.

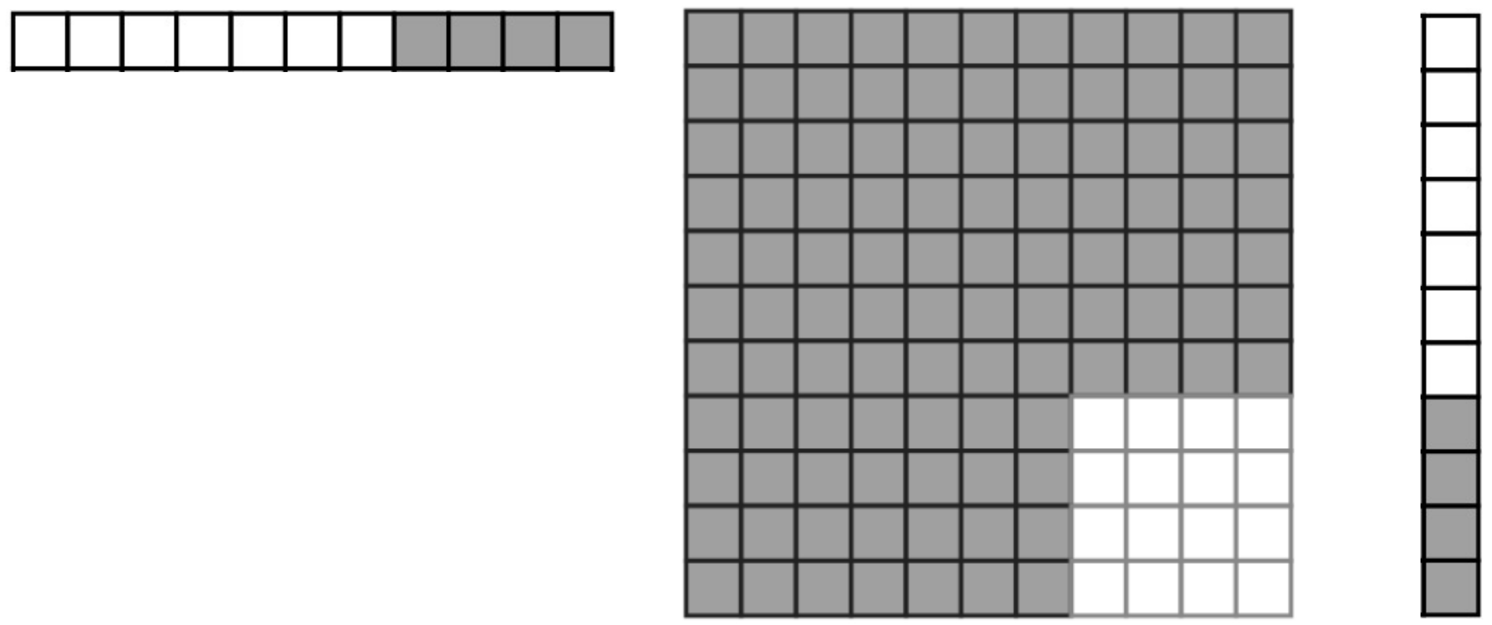
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n-m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n-m\}$.



$= 0$

f vanishes on O'.

Let $O = \mathbf{S}^{-1}(O')$.

p vanishes on O.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p'^{(1)}, \ldots, p'^{(m)})$ such that

$$p'^{(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\} .$$

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p^{'(1)}, \ldots, p^{'(m)})$ such that

$$p^{'(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\}\,.$$

What does $p^{'(k)}(\mathbf{x}, \mathbf{y})$ look like ?

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p'^{(1)}, \ldots, p'^{(m)})$ such that

$$p'^{(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\} .$$

What does $p'^{(k)}(\mathbf{x}, \mathbf{y})$ look like ?

Let $\tilde{\mathbf{P}}^{(k)}$ be the upper triangular representation of $p^{(k)}$.

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p'^{(1)}, \ldots, p'^{(m)})$ such that

$$p'^{(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\} .$$

What does $p'^{(k)}(\mathbf{x}, \mathbf{y})$ look like ?

Let $\tilde{\mathbf{P}}^{(k)}$ be the upper triangular representation of $p^{(k)}$.

$$\begin{aligned} p'^{(k)}(\mathbf{x}, \mathbf{y}) &= p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}) \\ &= (\mathbf{x} + \mathbf{y})^\top \tilde{\mathbf{P}}^{(k)}(\mathbf{x} + \mathbf{y}) - x^\top \tilde{\mathbf{P}}^{(k)}\mathbf{x} - y^\top \tilde{\mathbf{P}}^{(k)}\mathbf{y} \\ &= x^\top \tilde{\mathbf{P}}^{(k)}\mathbf{y} + y^\top \tilde{\mathbf{P}}^{(k)}\mathbf{x} \\ &= x^\top (\tilde{\mathbf{P}}^{(k)} + \tilde{\mathbf{P}}^{(k)\top})\mathbf{y} = x^\top \mathbf{B}^{(k)}\mathbf{y} \end{aligned}$$

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p'^{(1)}, \ldots, p'^{(m)})$ such that

$$p'^{(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\}\,.$$

What does $p'^{(k)}(\mathbf{x}, \mathbf{y})$ look like ?

Let $\tilde{\mathbf{P}}^{(k)}$ be the upper triangular representation of $p^{(k)}$.

$$\begin{aligned} p'^{(k)}(\mathbf{x}, \mathbf{y}) &= p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}) \\ &= (\mathbf{x} + \mathbf{y})^\top \tilde{\mathbf{P}}^{(k)} (\mathbf{x} + \mathbf{y}) - x^\top \tilde{\mathbf{P}}^{(k)} \mathbf{x} - y^\top \tilde{\mathbf{P}}^{(k)} \mathbf{y} \\ &= x^\top \tilde{\mathbf{P}}^{(k)} \mathbf{y} + y^\top \tilde{\mathbf{P}}^{(k)} \mathbf{x} \\ &= x^\top (\tilde{\mathbf{P}}^{(k)} + \tilde{\mathbf{P}}^{(k)\top}) \mathbf{y} = x^\top \mathbf{B}^{(k)} \mathbf{y} \end{aligned}$$

→ So, p' is bilinear and symmetric.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1, \ldots, m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1, \ldots, m\}$.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1, \ldots, m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1, \ldots, m\}$.

→ Equations:

For $i \in \{1, \ldots, m\}$ **do**

$\mathbf{o}_i = (o_1, \ldots, o_v, 0, \ldots, 1_{n-i+1}, 0, \ldots, 0)$

Model:

$$\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0, \text{ for } k \in \{1, \ldots, m\} \text{ and } j < i$$

$$\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0, \text{ for } k \in \{1, \ldots, m\}$$

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1,\ldots,m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1,\ldots,m\}$.

Equations:

For $i \in \{1,\ldots,m\}$ **do**

$\mathbf{o}_i = (o_1, \ldots, o_v, 0, \ldots, 1_{n-i+1}, 0, \ldots, 0)$

Model:

$$\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0, \text{ for } k \in \{1,\ldots,m\} \text{ and } j < i$$

$$\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0, \text{ for } k \in \{1,\ldots,m\}$$

In the first iteration, we have only quadratic equations, so this is the bottleneck. Linear constraints facilitate the resolution of a system.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1,\ldots,m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1,\ldots,m\}$.

Equations:

For $i \in \{1,\ldots,m\}$ **do**

$\mathbf{o}_i = (o_1, \ldots, o_v, 0, \ldots, 1_{n-i+1}, 0, \ldots, 0)$

Model:

$$\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0, \text{ for } k \in \{1,\ldots,m\} \text{ and } j < i$$

$$\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0, \text{ for } k \in \{1,\ldots,m\}$$

In the first iteration, we have only quadratic equations, so this is the bottleneck. Linear constraints facilitate the resolution of a system.

Kipnis-Shamir attack

[Kipnis, Shamir, 1998]

The orthogonal complement of a subspace

Let $V \subset \mathbb{F}_q^n$. The orthogonal complement of V is $V^\perp$ such that

$$V^\perp = \{\tilde{\mathbf{v}}_i \in \mathbb{F}_q^n \,|\, \langle \mathbf{v}_j, \tilde{\mathbf{v}}_i \rangle = 0, \text{ for all } \mathbf{v}_j \in V\}.$$

If V is m-dimensional, then $V^\perp$ is $(n-m)$-dimensional.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)} O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)} O = O^{\perp}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)} \mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top} \mathbf{B}^{(k)} \mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^{\perp} = \mathbf{B}^{(k_2)}O$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^{\perp} = \mathbf{B}^{(k_2)}O$.

Hence, we have that $\mathbf{B}^{(k_1)-1}\mathbf{B}^{(k_2)}O = O$, for all pairs $\mathbf{B}^{(k_1)}, \mathbf{B}^{(k_2)}$.

$$
\begin{aligned}
\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\
&= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\
&= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0
\end{aligned}
$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^\perp$.

Since $\dim(O^\perp) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^\perp$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^\perp = \mathbf{B}^{(k_2)}O$.

Hence, we have that $\mathbf{B}^{(k_1)-1}\mathbf{B}^{(k_2)}O = O$, for all pairs $\mathbf{B}^{(k_1)}, \mathbf{B}^{(k_2)}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^\top \mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Finding a common invariant subspace of a large number of linear maps is easy.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^{\perp} = \mathbf{B}^{(k_2)}O$.

Hence, we have that $\mathbf{B}^{(k_1)-1}\mathbf{B}^{(k_2)}O = O$, for all pairs $\mathbf{B}^{(k_1)}, \mathbf{B}^{(k_2)}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

→ Finding a common invariant subspace of a large number of linear maps is easy.

→ Oil and Vinegar becomes Unbalanced Oil and Vinegar because of this attack.

Intersection attack

[Beullens, 2021]

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \varnothing$, try to find a vector $\mathbf{x}$ in this intersection.

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

If $\mathbf{x}$ is in the intersection $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O$, then both $\mathbf{B}^{(k_1)-1}\mathbf{x}$ and $\mathbf{B}^{(k_2)-1}\mathbf{x}$ are in O.

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

If $\mathbf{x}$ is in the intersection $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O$, then both $\mathbf{B}^{(k_1)-1}\mathbf{x}$ and $\mathbf{B}^{(k_2)-1}\mathbf{x}$ are in O.

→ Equations:

$$p(\mathbf{B}^{(k_1)-1}\mathbf{x}) = 0$$
$$p(\mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$
$$p'(\mathbf{B}^{(k_1)-1}\mathbf{x}, \mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

If $\mathbf{x}$ is in the intersection $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O$, then both $\mathbf{B}^{(k_1)-1}\mathbf{x}$ and $\mathbf{B}^{(k_2)-1}\mathbf{x}$ are in O.

→ Equations:

$$p(\mathbf{B}^{(k_1)-1}\mathbf{x}) = 0$$
$$p(\mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$
$$p'(\mathbf{B}^{(k_1)-1}\mathbf{x}, \mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$

→ The attack can be generalised to find a vector in the intersection of more than two subspaces.

Recap

- The MQ problem is (usually) hard.
- We have a variety of solvers for (over)determined systems.
- Modelisation can be crucial to how efficient an attack is.
- The MQ problem can be easy for some structured systems. We use this to build trapdoors in crypto.
- We saw three different ways to model the recovery of the UOV trapdoor.

Recap

- The MQ problem is (usually) hard.
- We have a variety of solvers for (over)determined systems.
- Modelisation can be crucial to how efficient an attack is.
- The MQ problem can be easy for some structured systems. We use this to build trapdoors in crypto.
- We saw three different ways to model the recovery of the UOV trapdoor.