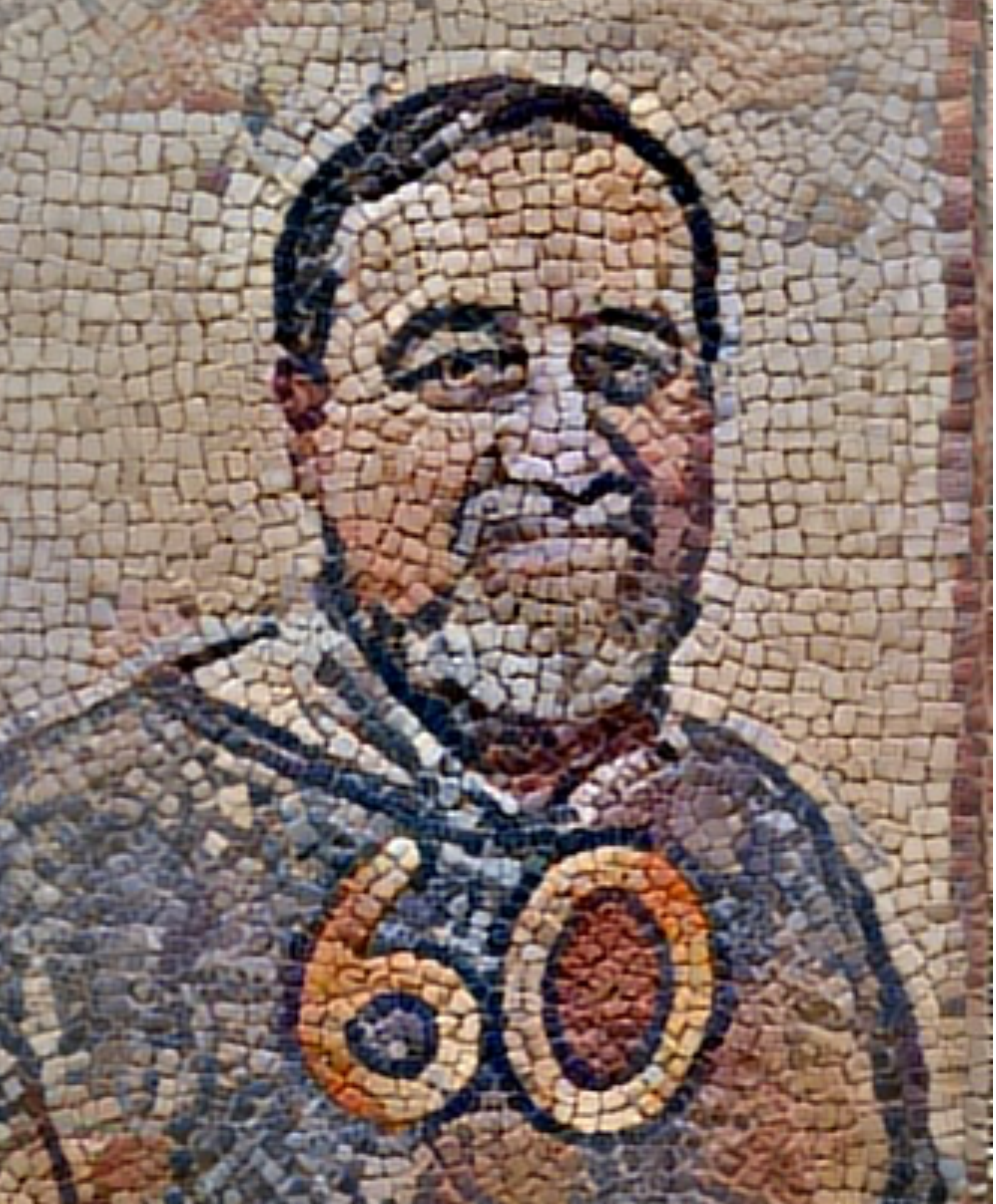
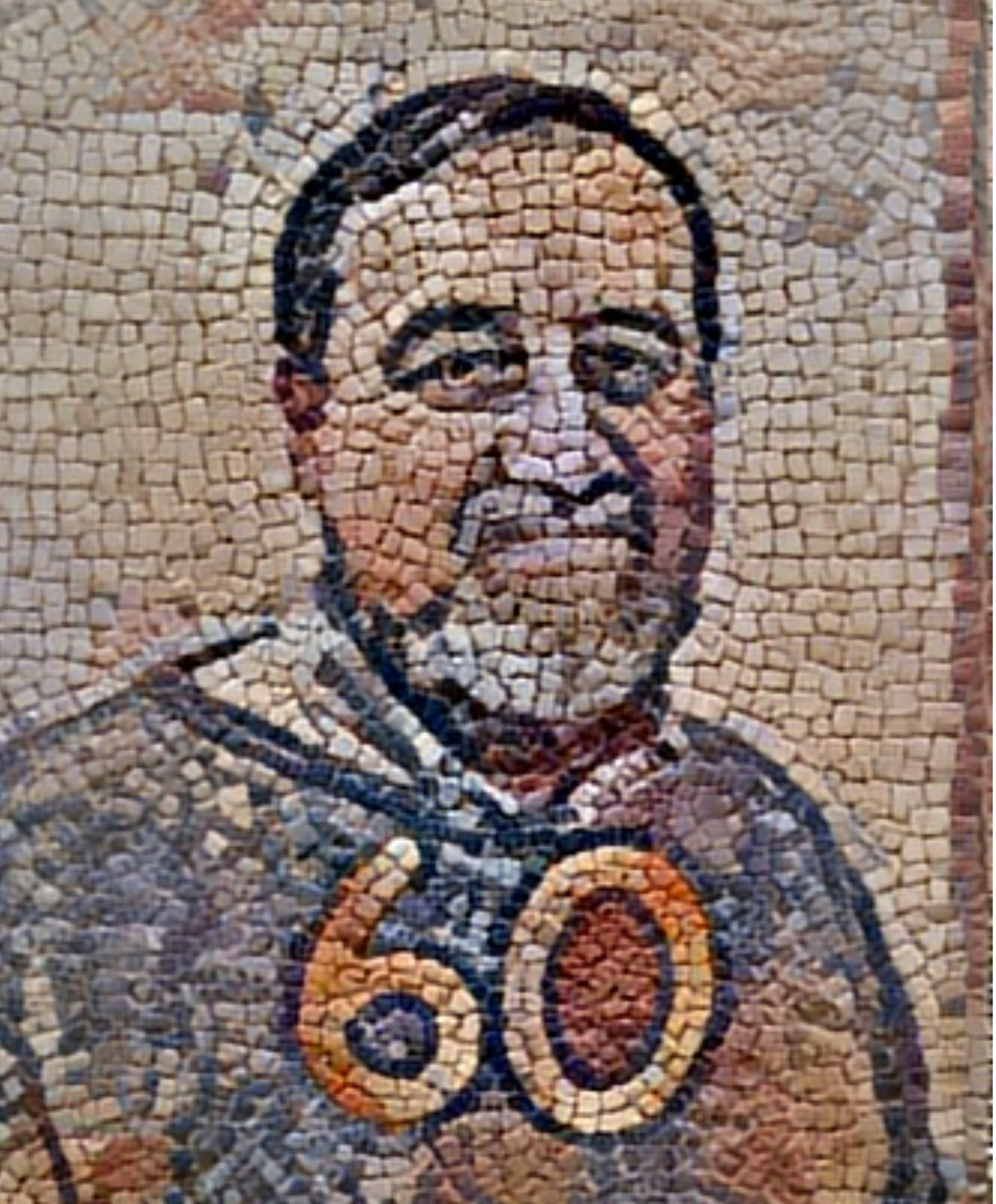
Backdoors for Constraint Satisfaction

Stefan Szeider
TU Wien, Vienna, Austria

ALGORITHMS AND COMPLEXITY GROUP

Gregory Gutin's 60th Birthday Conference
January 7th & 8th 2017


60



…every stone
a theorem

Gregory and Constraint Satisfaction

Gregory has published several papers on constraint satisfaction problems (Max-r-CSP, Max-Permutation-CSP, Workflow-CSP)

Backdoors for Constraint Satisfaction

Joint work with
Robert Ganian, Serge Gaspers, Neeldhara Misra,
Sebastian Ordyniak, M.S. Ramanujan, and Standa Živný

Different Problems $\subseteq$ CSP

- Graph k-Coloring, k-SAT, k-Clique, Graph Homomorphism, Conjunctive Database Query, ….
- All these problems (and many more) can be seen as special cases of a very general problem, the **Constraint Satisfaction Problem**
- Introduced by Montanari 1974, has been focus of intensive research (TCS, AI, Combinatorics, Algebra,…)

The Constraint Satisfaction Problem (CSP)

- Instance: **I=(V,D,C)** where
- **V** is a finite set of variables
- domain **D** is a finite set of values
- **C** is a finite set **constraints**
- $\mathbf{f}: V \to D$ satisfies $C=((x_1,...,x_r),R^r)$ if $(\mathbf{f}(x_1),...,\mathbf{f}(x_r)) \in R^r$

A constraint for 3-coloring

u	v
Red	Blue
Red	Green
Blue	Red
Blue	Green
Green	Red
Green	Blue

instantiating variables

u	v	w
1	1	1
1	0	1
0	1	1
0	0	1

set u=1

instantiating variables

u	v	w
1	1	1
1	0	1
~~0~~	~~1~~	~~1~~
0	0	1

set u=1

instantiating variables

u	v	w
1	1	1
1	0	1
~~0~~	~~1~~	~~1~~
~~0~~	~~0~~	~~1~~

set u=1

instantiating variables

u	v	w
1	1	1
1	0	1
~~0~~	~~1~~	~~1~~
~~0~~	~~0~~	~~1~~

set u=1

instantiating variables

~~u~~	v	w
~~1~~	1	1
~~1~~	0	1
~~0~~	~~1~~	~~1~~
~~0~~	~~0~~	~~1~~

set u=1

v	w
1	1
0	1

instantiating variables

~~u~~	v	w
~~1~~	1	1
~~1~~	0	1
~~0~~	~~1~~	~~1~~
~~0~~	~~0~~	~~1~~

set u=1

v	w
1	1
0	1

Generalizes to: partial assignment **f** applied to CSP instance **I** yields a new CSP instance **I[f]**

Complexity of CSP

- CSP is NP-complete (even restricted to Boolean domain or binary constraints)
- Tractable subproblems (**islands of tractability**)
  - **language restrictions:** how do the constraint relations look like, no matter how the variables and constraints interact
  - **structural restrictions:** how do variables and constraints interact, no matter how the constraint relations look like

Language Restrictions

- **CSP(Γ)** consists of all CSP instances whose constraints use relations from the set **Γ** (called the **constraint language**)

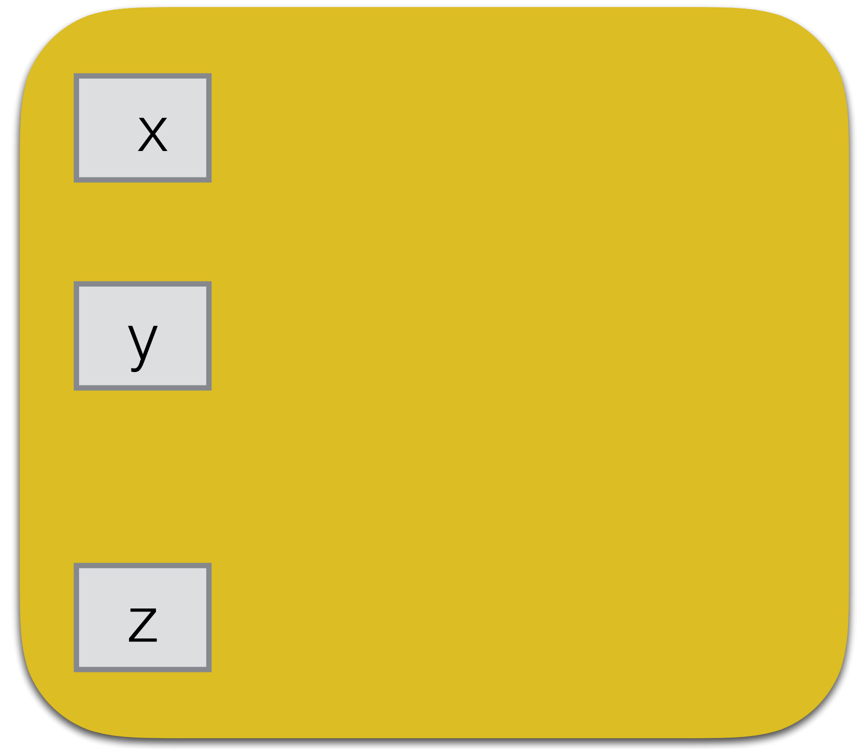
- **Schaefer's Dichotomy Theorem (1978)** *(1800 citations)*
For every *Boolean* Γ , CSP(Γ) is either tractable or NPc.

- **Feder-Vadi-Conjecture (1993)** *(2000 citations)*
For every Γ , CSP(Γ) is either tractable or NPc (there circulate rumours that the conjecture has been established)

How to deal with instances that are close to an island of tractability?

Backdoors

Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**

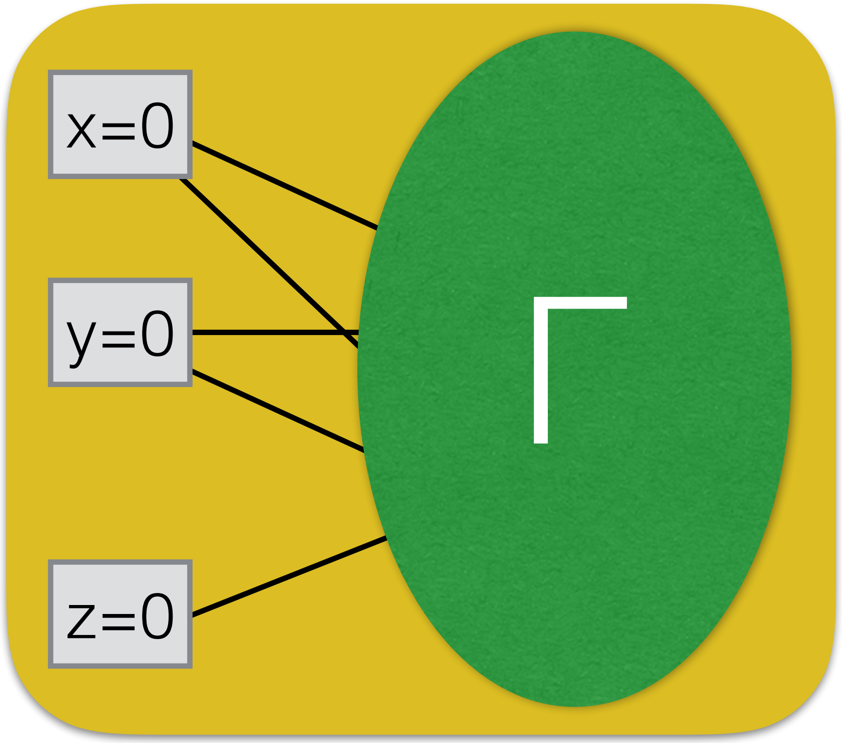
- Call **C** the **base class** of the backdoor

- Backdoors where introduced by [Williams, Gomes, Selman 2003]

- Distance to island: **size of smallest backdoor**

Backdoors

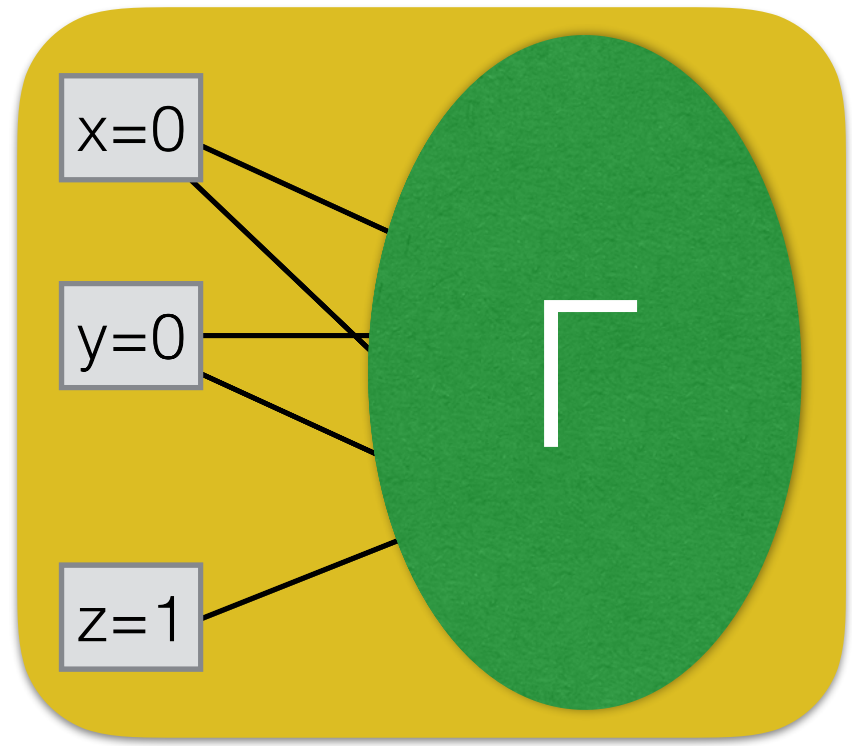
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

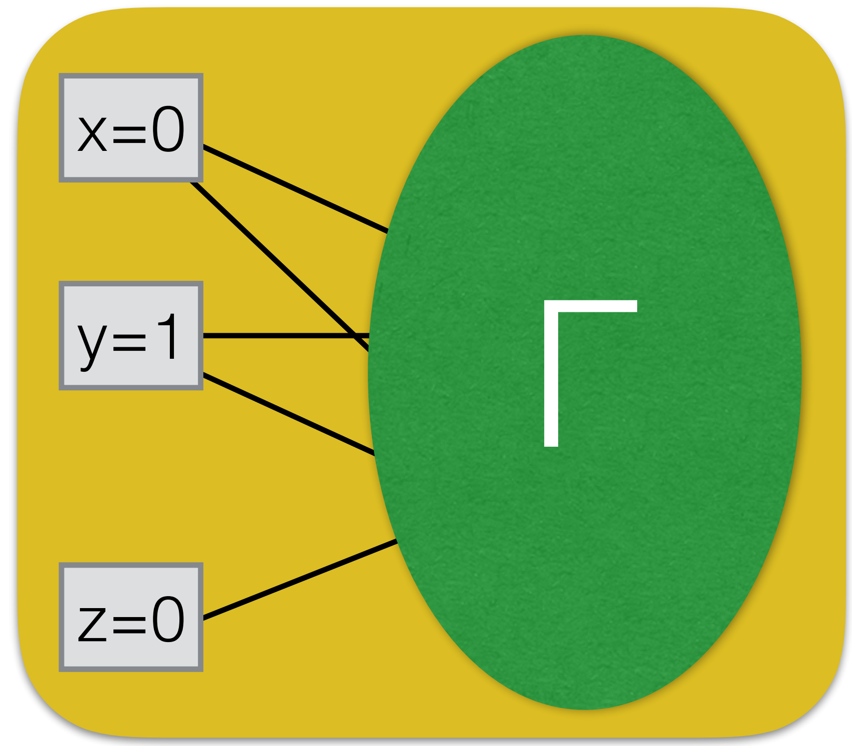
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

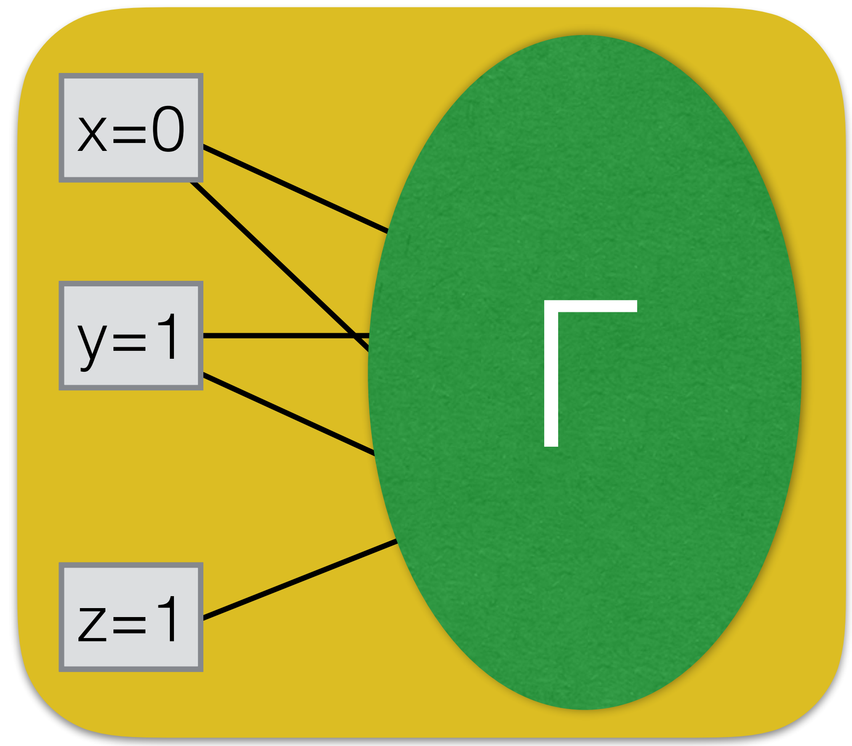
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

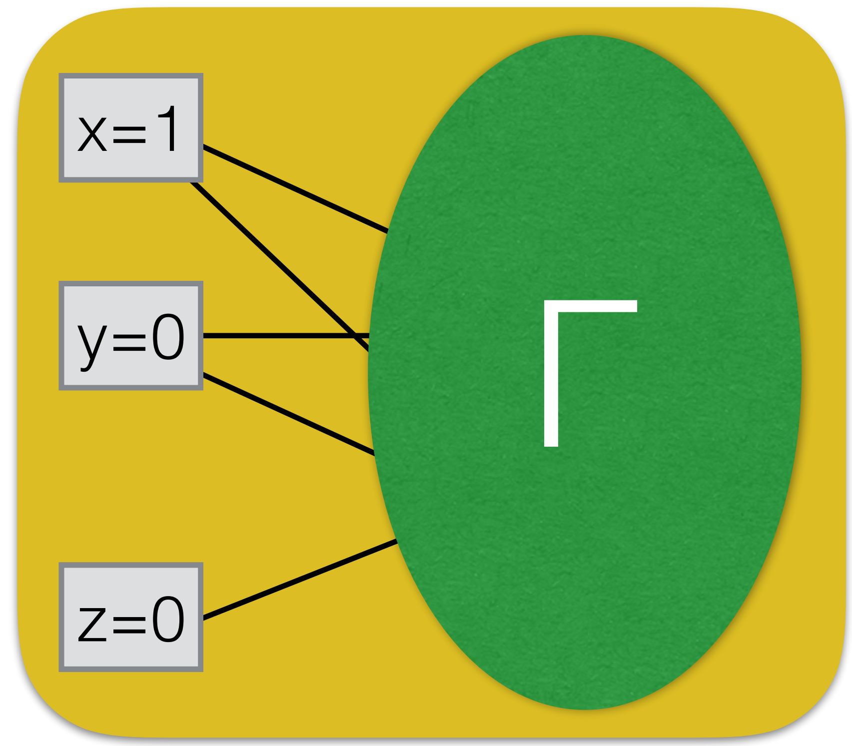
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

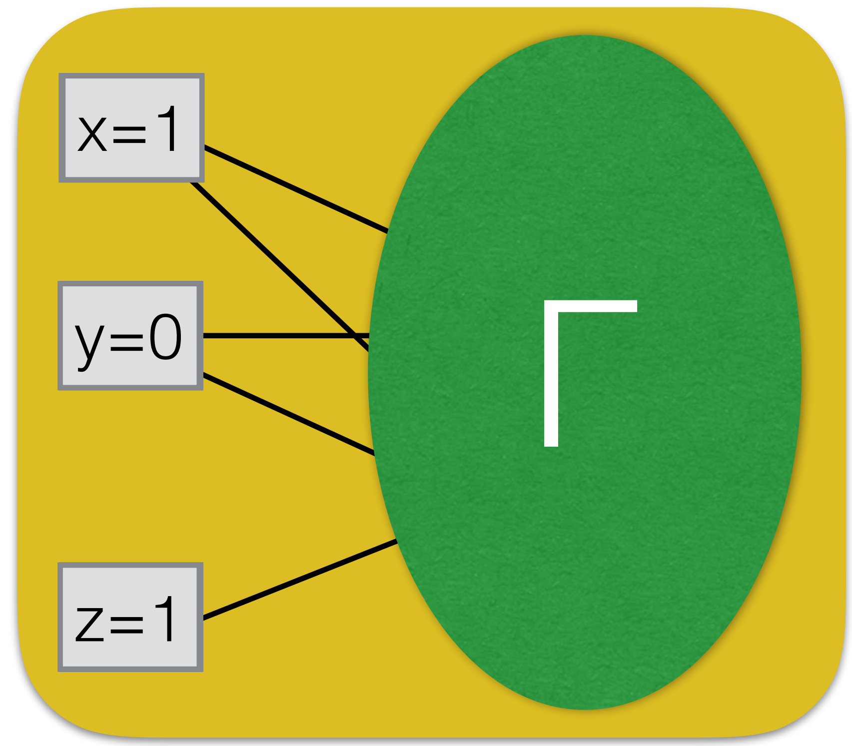
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

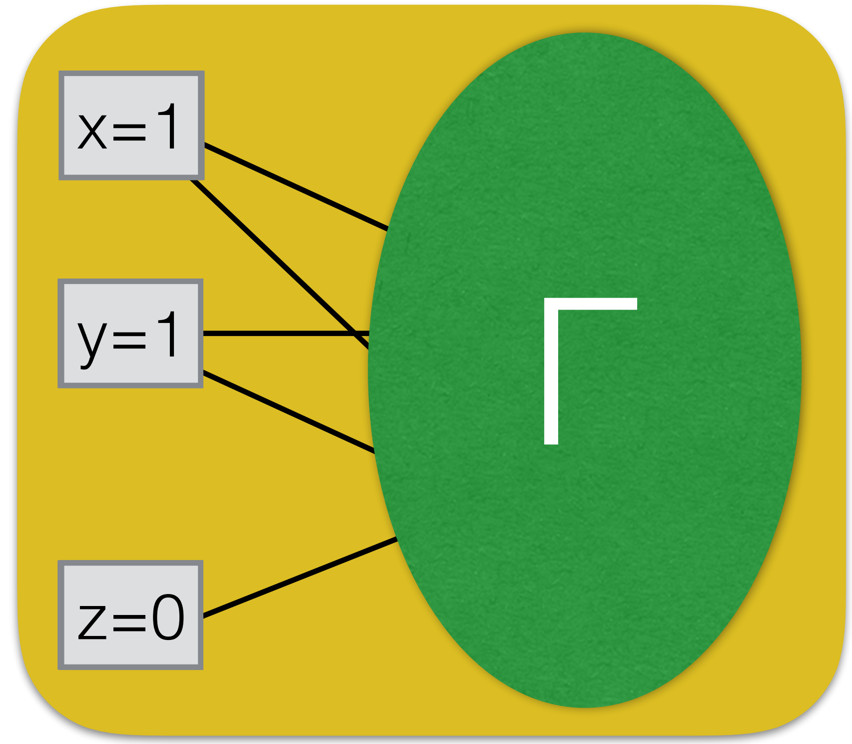
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

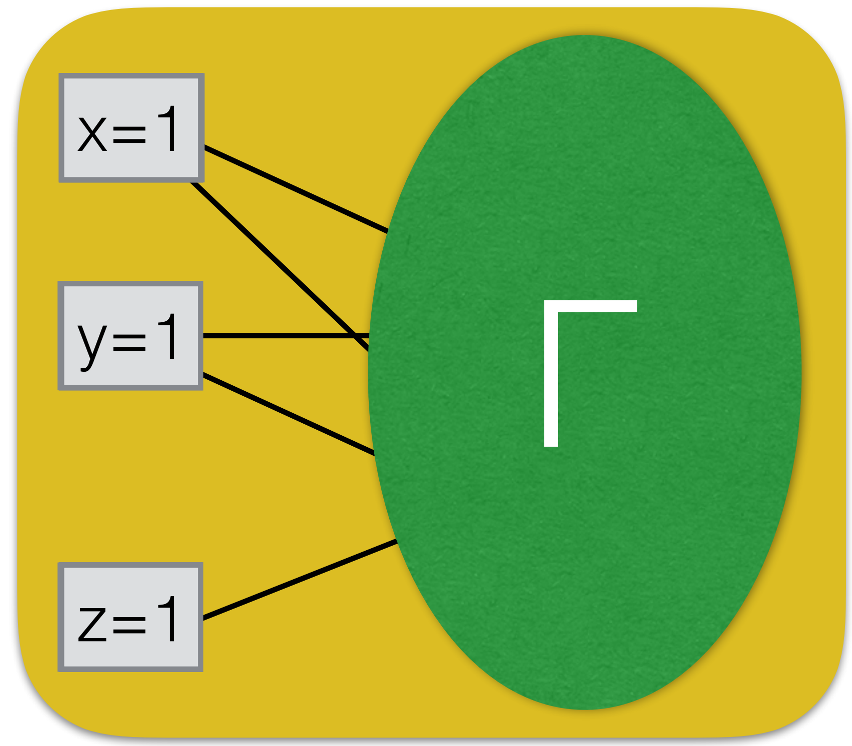
Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Backdoors

Instance belongs to CSP(Γ) for all possible instantiations of backdoor variables



- A strong **backdoor** of a CSP instance into an island of tractability **C** is a set B of variables such that **for each possible instantiation** of the variables in B, the reduced instance belongs to **C**
- Call **C** the **base class** of the backdoor
- Backdoors where introduced by [Williams, Gomes, Selman 2003]
- Distance to island: **size of smallest backdoor**

Two phases-approach

Two phases-approach

- **Phase 1** (**backdoor detection**): given a CSP instance and an upper bound k, find a backdoor of size ≤ k or decide that it doesn't exist.

Two phases-approach

- **Phase 1** (**backdoor detection**): given a CSP instance and an upper bound k, find a backdoor of size ≤ k or decide that it doesn't exist.
- **Phase 2** (**backdoor evaluation**): given a CSP instance and a backdoor of size ≤ k, solve the CSP instance.

Two phases-approach

- **Phase 1** (**backdoor detection**): given a CSP instance and an upper bound k, find a backdoor of size ≤ k or decide that it doesn't exist.

- **Phase 2** (**backdoor evaluation**): given a CSP instance and a backdoor of size ≤ k, solve the CSP instance.

solve $|D|^k$ many tractable instances

Two phases-approach

- **Phase 1** (**backdoor detection**): given a CSP instance and an upper bound k, find a backdoor of size ≤ k or decide that it doesn't exist.

- **Phase 2** (**backdoor evaluation**): given a CSP instance and a backdoor of size ≤ k, solve the CSP instance.

solve $|D|^k$ many tractable instances

FPT if D is constant

Two phases-approach

- **Phase 1** (**backdoor detection**): given a CSP instance and an upper bound k, find a backdoor of size ≤ k or decide that it doesn't exist.

challenging algorithmic problem

- **Phase 2** (**backdoor evaluation**): given a CSP instance and a backdoor of size ≤ k, solve the CSP instance.

solve $|D|^k$ many tractable instances

FPT if D is constant

Two phases-approach

- **Phase 1** (**backdoor detection**): given a CSP instance and an upper bound k, find a backdoor of size ≤ k or decide that it doesn't exist.

challenging algorithmic problem

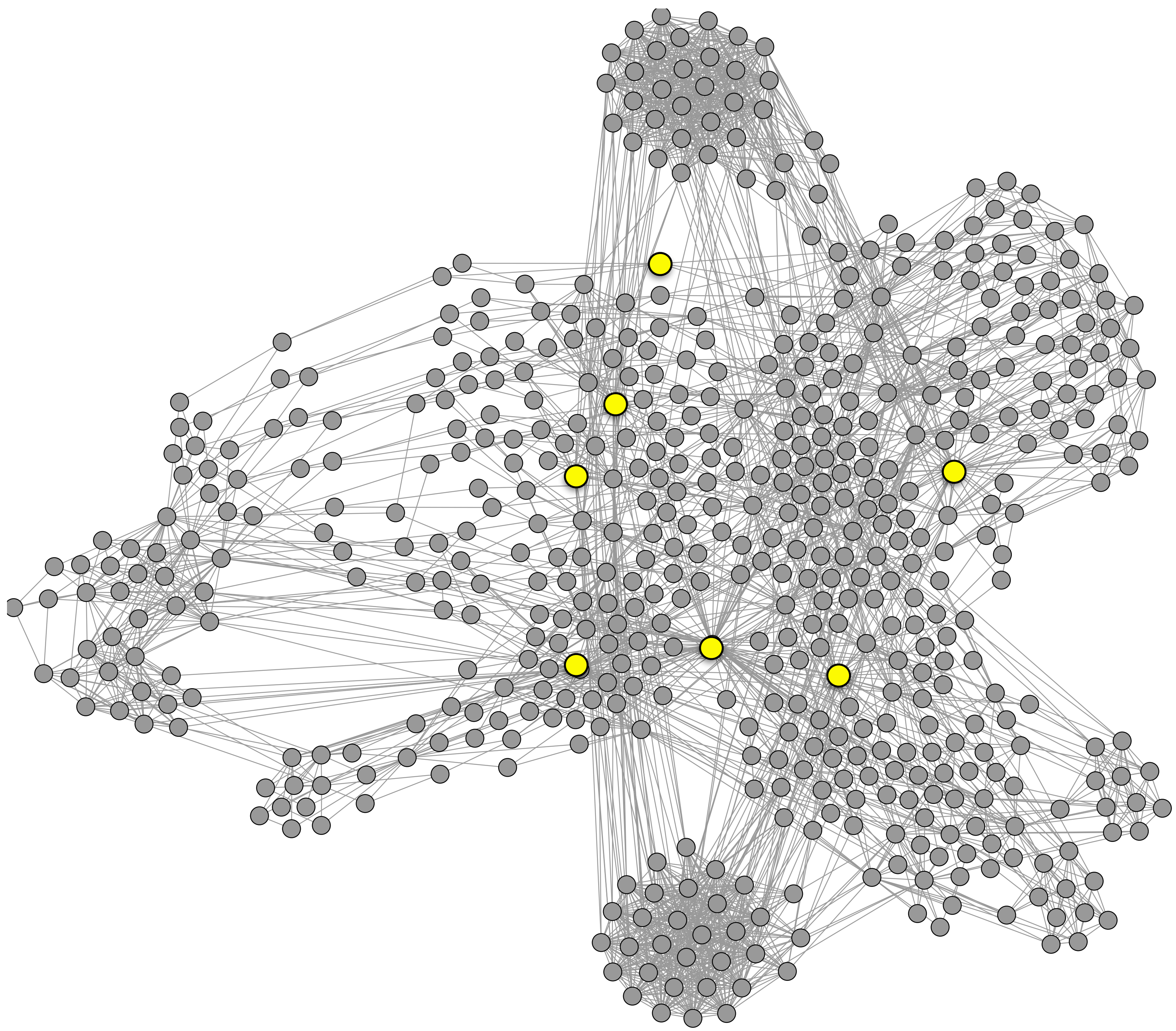
when FPT?

- **Phase 2** (**backdoor evaluation**): given a CSP instance and a backdoor of size ≤ k, solve the CSP instance.

solve $|D|^k$ many tractable instances

FPT if D is constant

backdoor can be a needle in the haystack!



Basic Result

- If **domain size** is unbounded, then backdoor evaluation is not FPT
- If **arity** is unbounded, then backdoor detection is not FPT (unless FPT=W[2]) by standard reduction from HS
- Thus, for CSP(Γ) we must restrict ourselves to **finite** constraint languages Γ
- CSP(Γ) recognisable in poly time (Γ is **recognizable**)
- CSP(Γ) tractable in poly time (Γ is **tractable**)

Basic Result

- If **domain size** is unbounded, then backdoor evaluation is not FPT
- If **arity** is unbounded, then backdoor detection is not FPT (unless FPT=W[2]) by standard reduction from HS
- Thus, for CSP(Γ) we must restrict ourselves to **finite** constraint languages Γ
- CSP(Γ) recognisable in poly time (Γ is **recognizable**)
- CSP(Γ) tractable in poly time (Γ is **tractable**)

THM: CSP is FPT parameterised by the size of a smallest backdoor into CSP(Γ) for any finite, recognisable, and tractable Γ.

Basic Result

- If **domain size** is unbounded, then backdoor evaluation is not FPT
- If **arity** is unbounded, then backdoor detection is not FPT (unless FPT=W[2]) by standard reduction from HS
- Thus, for CSP(Γ) we must restrict ourselves to **finite** constraint languages Γ
- CSP(Γ) recognisable in poly time (Γ is **recognizable**)
- CSP(Γ) tractable in poly time (Γ is **tractable**)

THM: CSP is FPT parameterised by the size of a smallest backdoor into CSP(Γ) for any finite, recognisable, and tractable Γ.

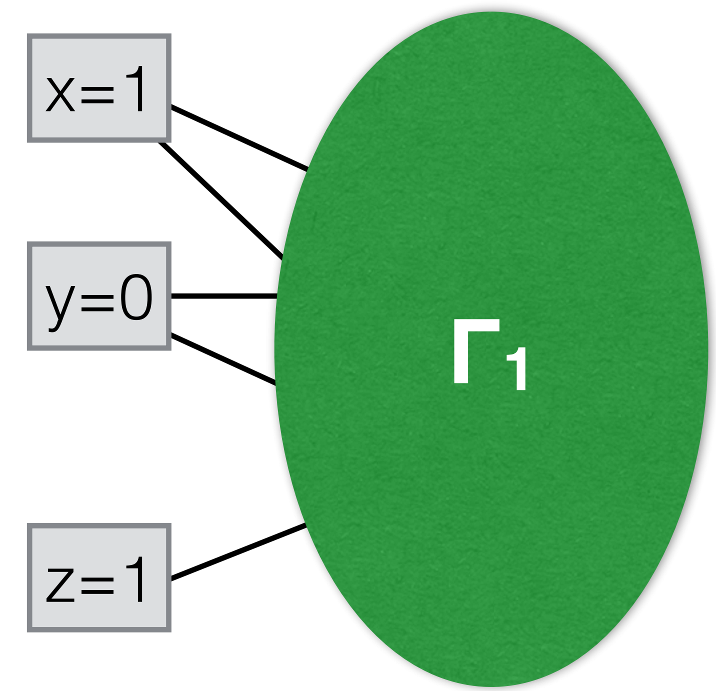
bounded search tree

Extensions: **Heterogeneous** base classes

Each assignment to backdoor can put the formula on a different island

Can yield arbitrarily smaller backdoors

notation: $\mathbf{CSP(\Gamma_1)} \cup \cdots \cup \mathbf{CSP(\Gamma_r)}$

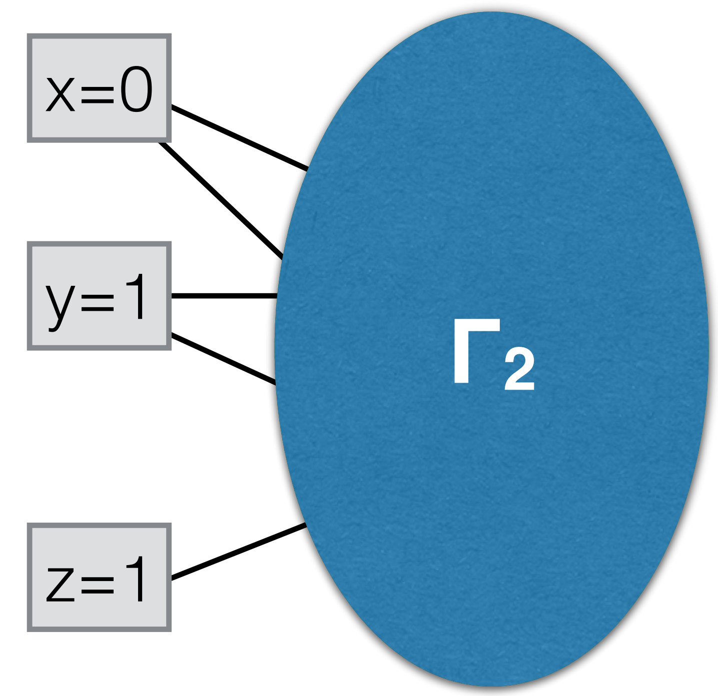


Extensions: **Heterogeneous** base classes

Each assignment to backdoor can put the formula on a different island

Can yield arbitrarily smaller backdoors

notation: $\mathbf{CSP(\Gamma_1)} \cup \cdots \cup \mathbf{CSP(\Gamma_r)}$

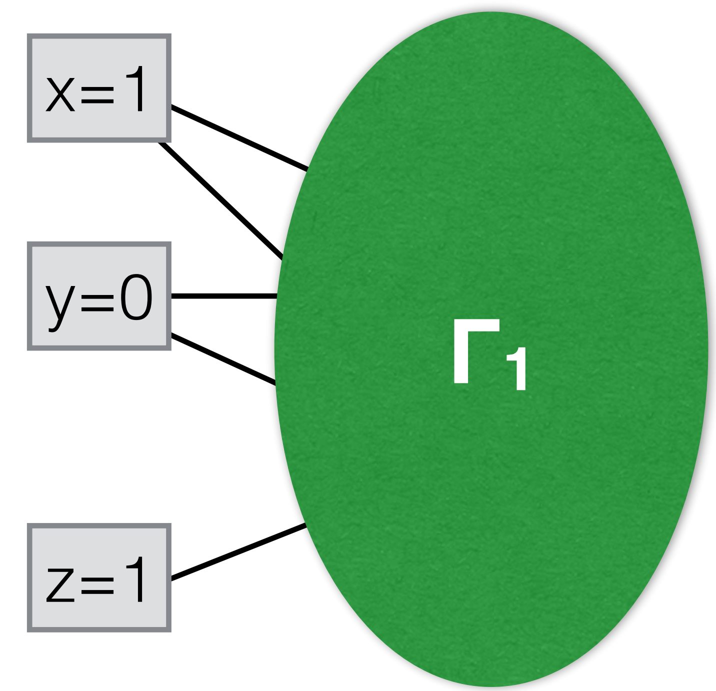


Extensions: **Heterogeneous** base classes

Each assignment to backdoor can put the formula on a different island

Can yield arbitrarily smaller backdoors

notation: $\mathbf{CSP(\Gamma_1)} \cup \cdots \cup \mathbf{CSP(\Gamma_r)}$

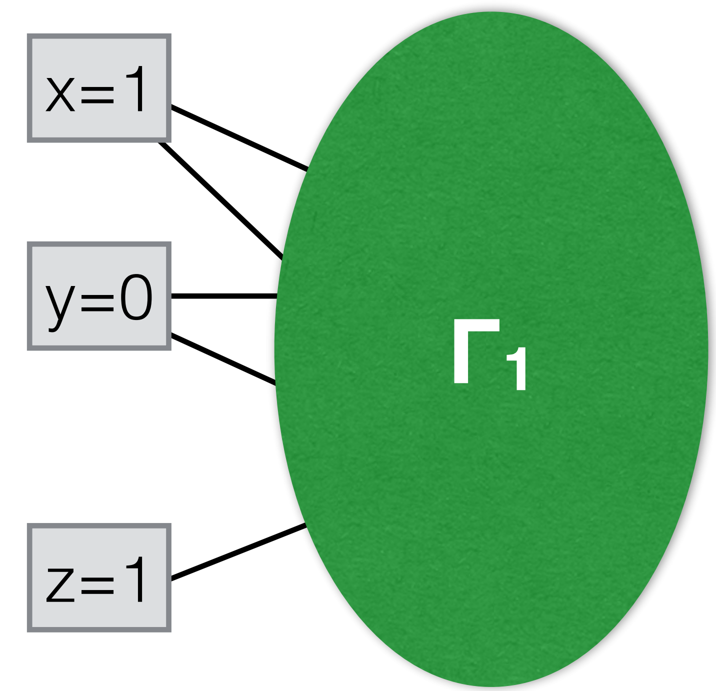


Extensions: **Heterogeneous** base classes

Each assignment to backdoor can put the formula on a different island

Can yield arbitrarily smaller backdoors

notation: **CSP(Γ_1)**$\cup\cdots\cup$ **CSP(Γ_r)**



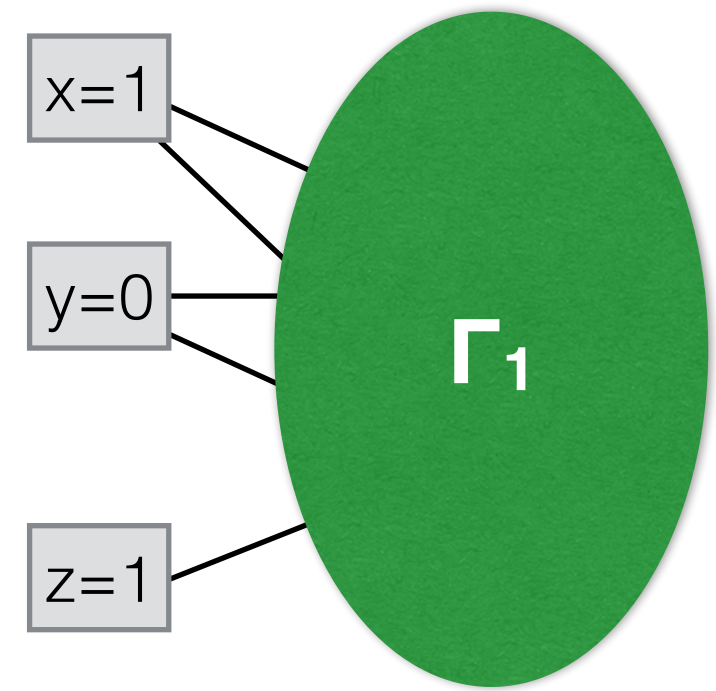
THM: CSP is FPT parameterised by the size of a smallest backdoor into CSP(Γ_1)$\cup\cdots\cup$ CSP(Γ_r) for Γ_i finite, recognizable, and tractable.

Extensions: **Heterogeneous** base classes

Each assignment to backdoor can put the formula on a different island

Can yield arbitrarily smaller backdoors

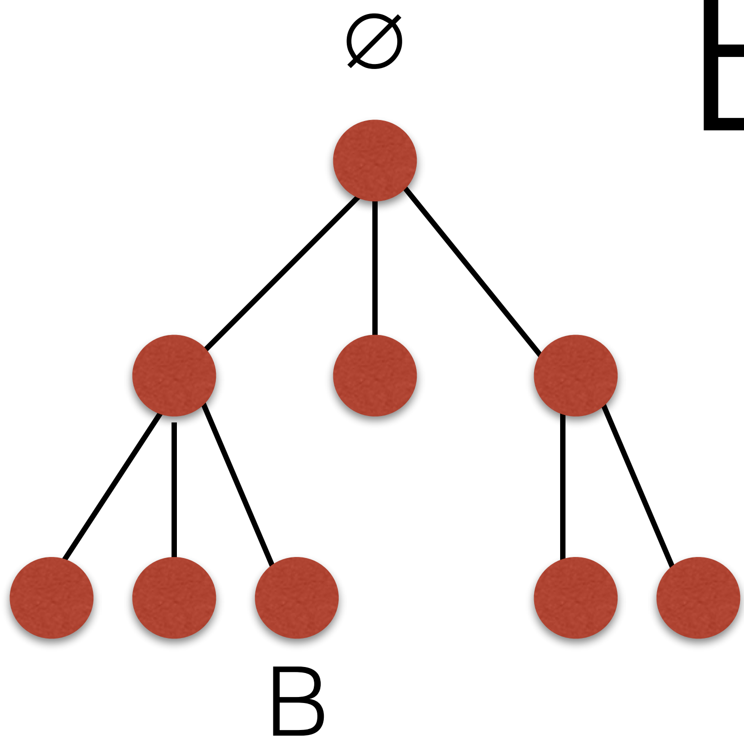
notation: **CSP(Γ_1)**$\cup\cdots\cup$ **CSP(Γ_r)**



THM: CSP is FPT parameterised by the size of a smallest backdoor into CSP(Γ_1)$\cup\cdots\cup$ CSP(Γ_r) for Γ_i finite, recognizable, and tractable.

bounded search tree

Branching



- If B is not a strong backdoor, then there is some **f**:B→D such that **I[f]** $\notin$ CSP(Γ_1)∪···∪ CSP(Γ_r).

- In particular, for each i there exists some $c_i \in$ **I[f]** such that $c_i \notin \Gamma_i$

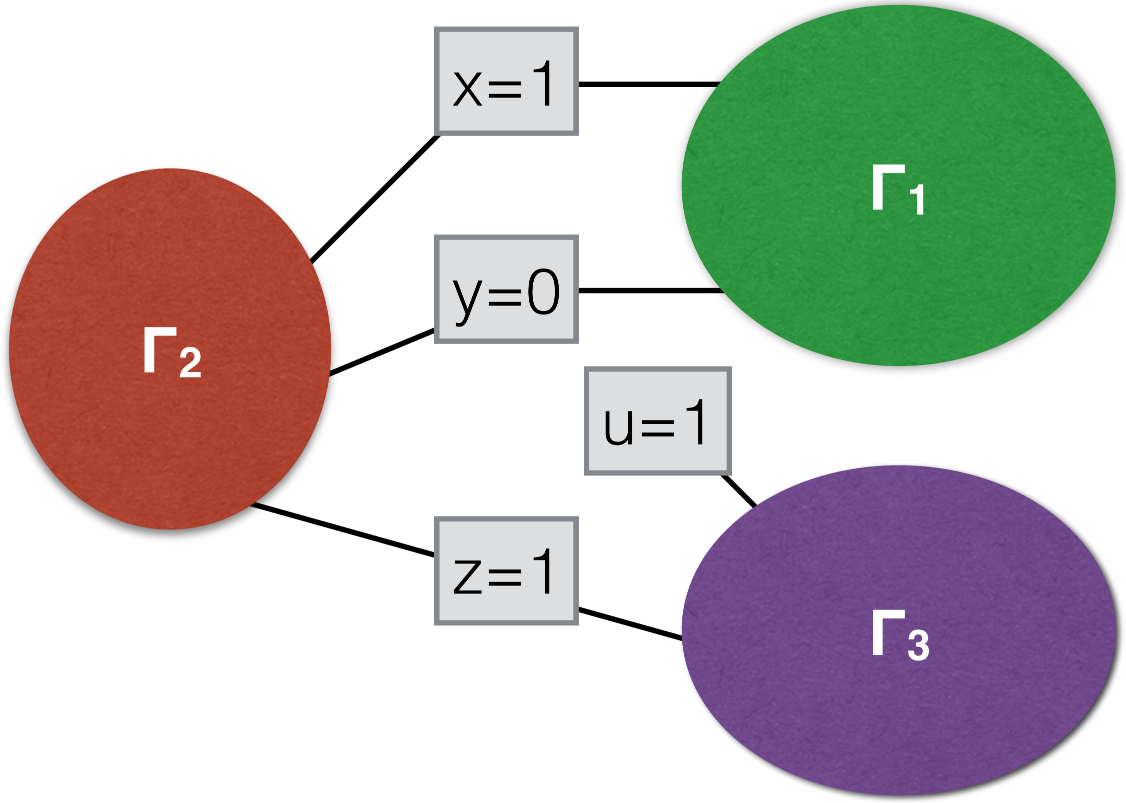
- Now branch into B ∪ {v} for each variable v that appears in c_i for some i.

Extensions: **Scattered** base classes

Each connected component of reduced instance may belong to different island

Can yield arbitrarily smaller backdoors than heterogeneous islands

Notation: $\mathbf{CSP(\Gamma_1)} \oplus \cdots \oplus \mathbf{CSP(\Gamma_r)}$

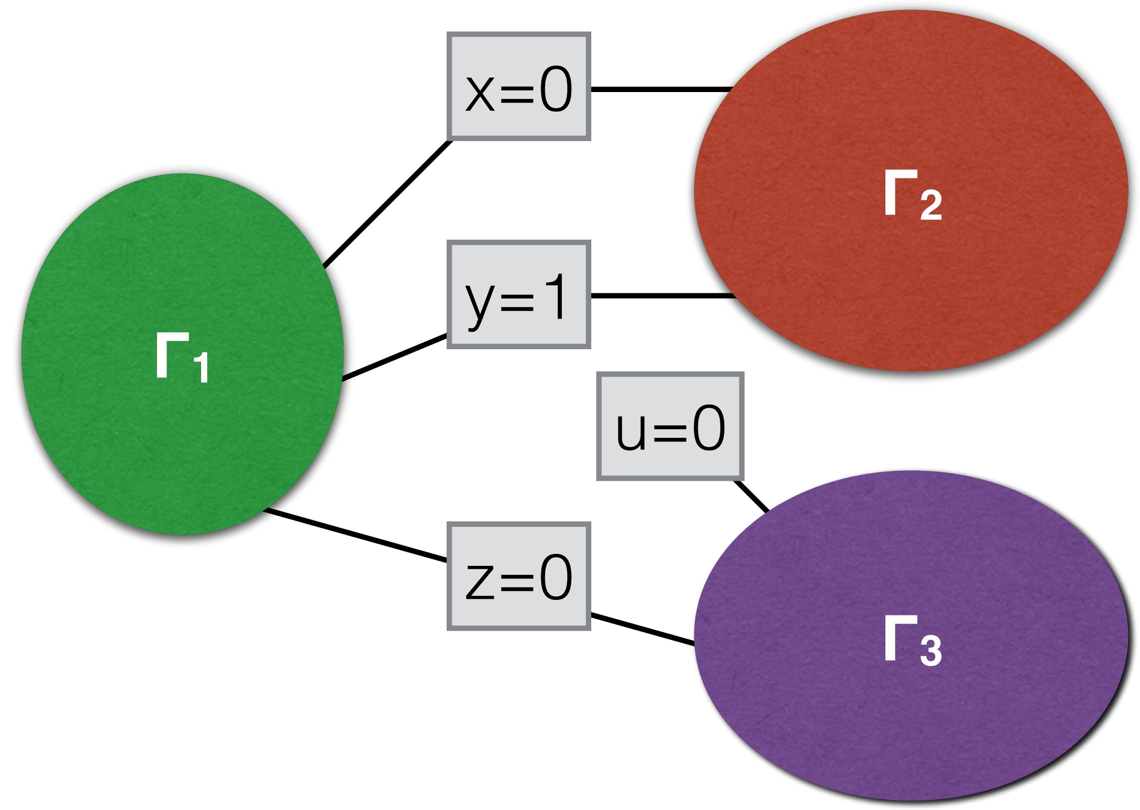


Extensions: **Scattered** base classes

Each connected component of reduced instance may belong to different island

Can yield arbitrarily smaller backdoors than heterogeneous islands

Notation: $\mathbf{CSP(\Gamma_1)} \oplus \cdots \oplus \mathbf{CSP(\Gamma_r)}$

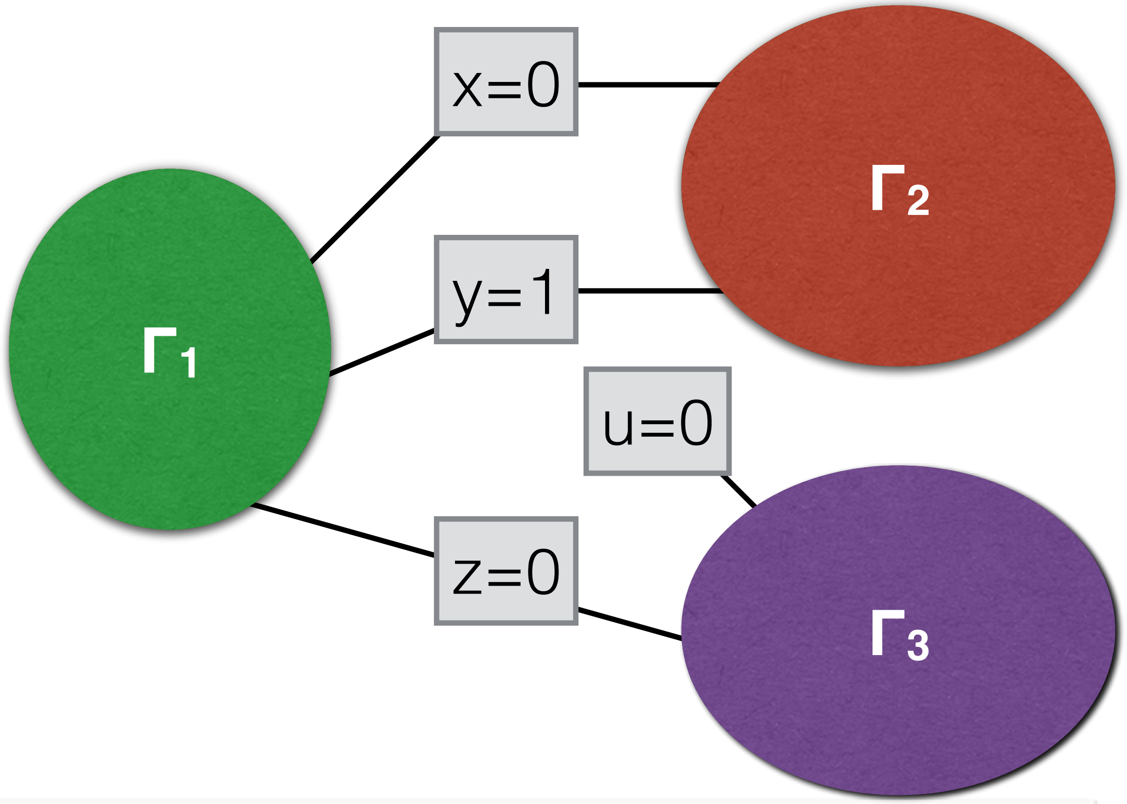


Extensions: **Scattered** base classes

Each connected component of reduced instance may belong to different island

Can yield arbitrarily smaller backdoors than heterogeneous islands

Notation: $\mathbf{CSP(\Gamma_1)} \oplus \cdots \oplus \mathbf{CSP(\Gamma_r)}$



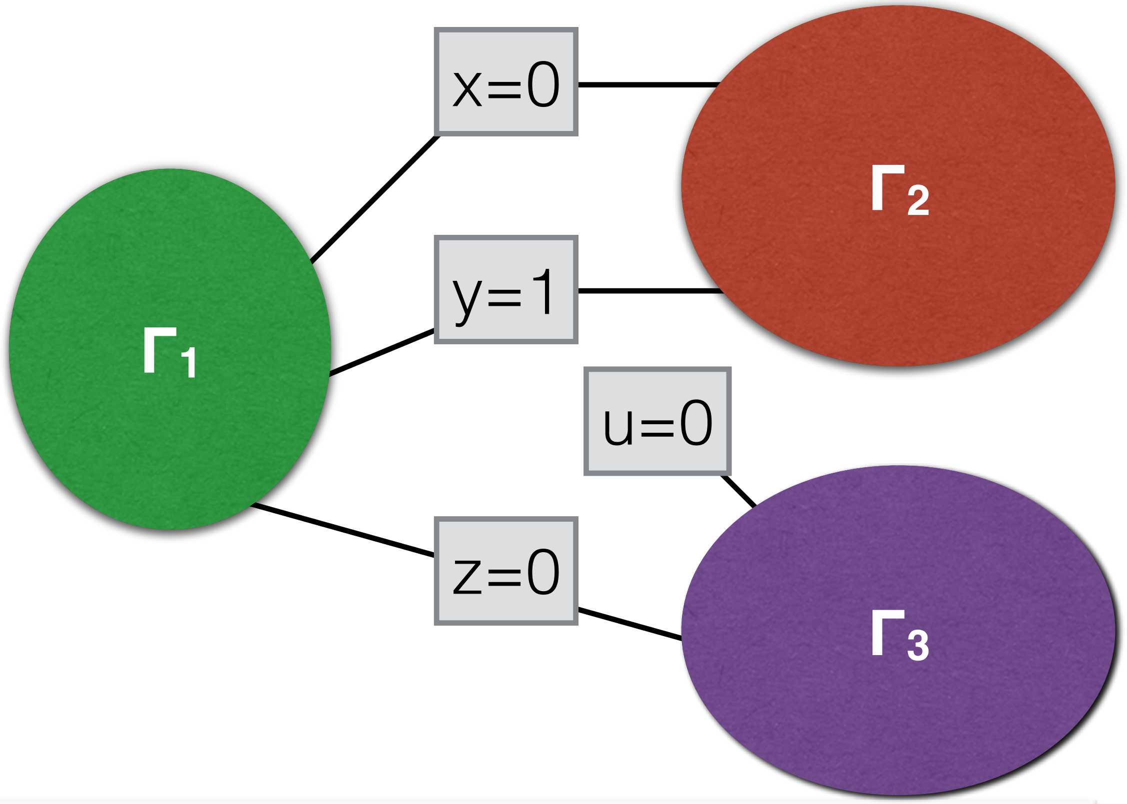
THM: CSP is FPT parameterised by the size of a smallest backdoor into $\mathbf{CSP(\Gamma_1)} \oplus \cdots \oplus \mathbf{CSP(\Gamma_r)}$ for $\mathbf{\Gamma_i}$ finite, recognizable, tractable, and closed under partial assignments.

Extensions: **Scattered** base classes

Each connected component of reduced instance may belong to different island

Can yield arbitrarily smaller backdoors than heterogeneous islands

Notation: $\mathbf{CSP(\Gamma_1) \oplus \cdots \oplus CSP(\Gamma_r)}$



THM: CSP is FPT parameterised by the size of a smallest backdoor into $\mathbf{CSP(\Gamma_1) \oplus \cdots \oplus CSP(\Gamma_r)}$ for $\mathbf{\Gamma_i}$ finite, recognizable, tractable, and closed under partial assignments.

Iterative compression, tight separator sequences, etc.

Tractable CSP due to **structural** restrictions

Tractable CSP due to **structural** restrictions

- **Constraint graph** (aka primal graph) of a CSP instance
    - **vertices**: variables
    - **edges**: two variables are adjacent off they appear together in a constraint

Tractable CSP due to **structural** restrictions

- **Constraint graph** (aka primal graph) of a CSP instance
    - **vertices**: variables
    - **edges**: two variables are adjacent off they appear together in a constraint
- **Freuder 1982**: CSP is polynomial time tractable if the treewidth of the constraint graph is constant. *(870 citations)*

Tractable CSP due to **structural** restrictions

- **Constraint graph** (aka primal graph) of a CSP instance
    - **vertices**: variables
    - **edges**: two variables are adjacent off they appear together in a constraint
- **Freuder 1982**: CSP is polynomial time tractable if the treewidth of the constraint graph is constant. *(870 citations)*
- Actually: FPT by treewidth if domain is bounded

Incomparable parameters

size of backdoor to CSP(Γ)

treewidth of constraint graph

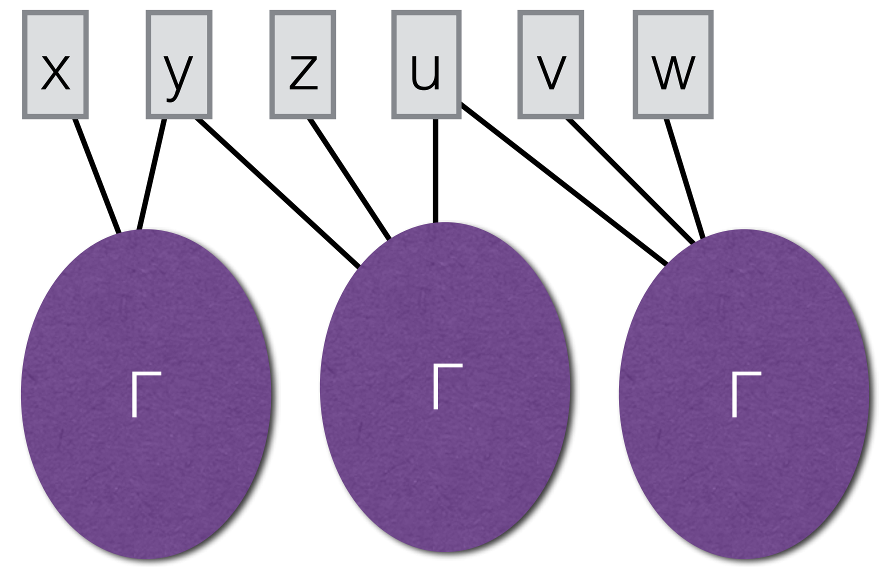
Incomparable parameters

size of backdoor to CSP(Γ)

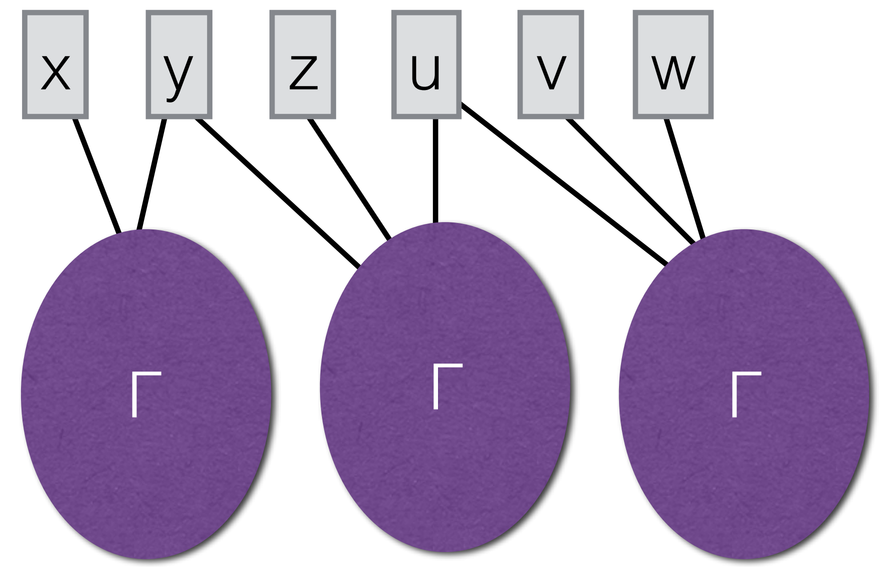
treewidth of constraint graph

backdoor treewidth wrt Γ

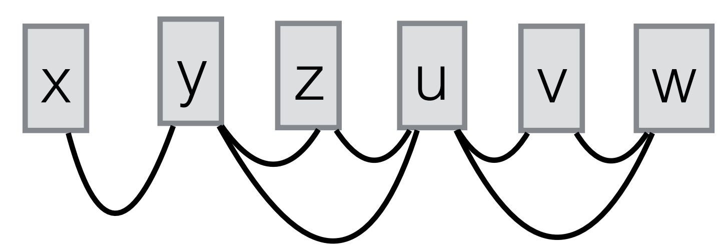
backdoor treewidth



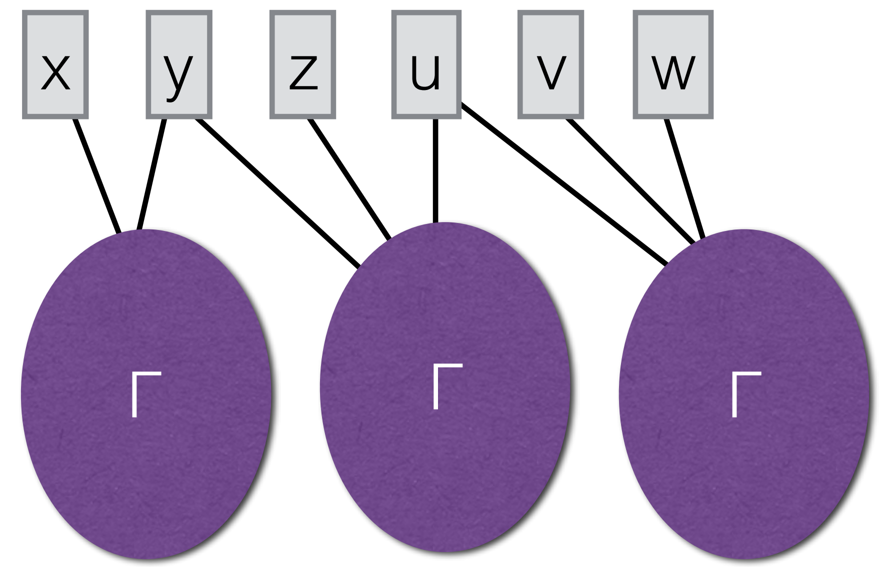
backdoor treewidth



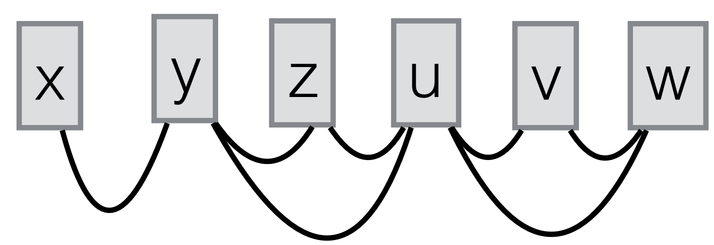
torso graph:



backdoor treewidth

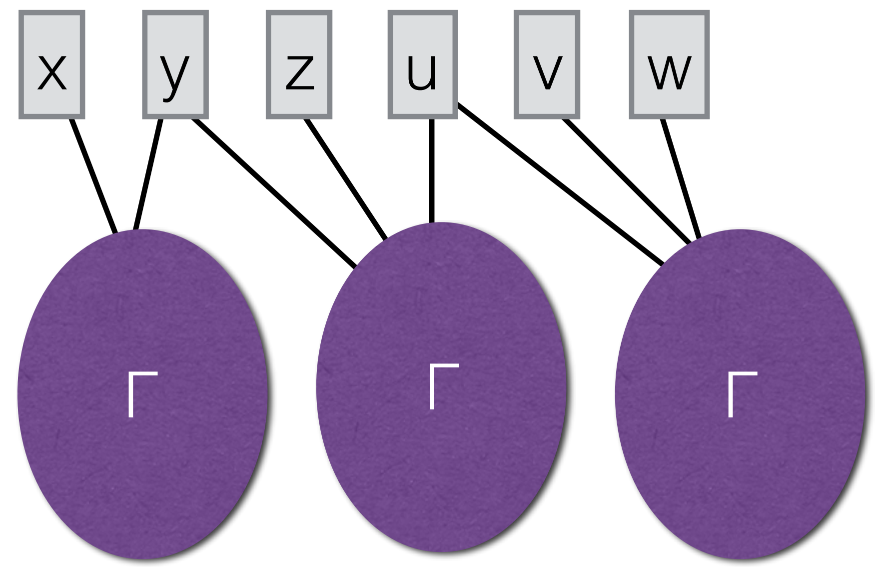


torso graph:

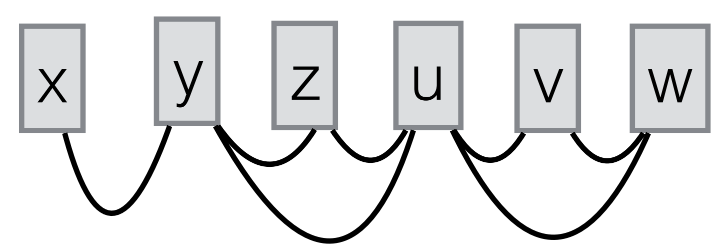


the **backdoor treewidth** wrt Γ is the minimum treewidth over the torso graphs of all strong backdoor sets into CSP(Γ)

backdoor treewidth



torso graph:



the **backdoor treewidth** wrt Γ is the minimum treewidth over the torso graphs of all strong backdoor sets into CSP(Γ)

backdoor treewidth ≤ min{ treewidth, backdoor size into CSP(Γ) }

Results

THM: CSP is FPT parameterised by the backdoor treewidth wrt Γ for any finite, recognisable, and tractable Γ.

- Finding the backdoor that minimises the backdoor treewidth is FPT.

  Algorithm based on ideas related to “boundaried graphs”, “replacement framework”, and “recursive-understanding technique”. Ultimately it relies on a lemma stating a finite state property.

- Once the backdoor is found, the instance can be compiled into a CSP of bounded treewidth, and then solved using Freuder’s algorithm in time D^k

Future work

- Extend result on scattered base classes to infinite sequences of constraint languages $\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \ldots$
- Extend backdoor treewidth to other graph invariants of torso graph wrt specific constraint languages
- Avoid detection and solve directly

Grisha, happy
B-day!