Convexity Makes Multiple AllDifferent Constraints Tractable

Michael Fellows, Tobias Friedrich, Danny Hermelin, Nina Naraodytska, **Frances Rosamond**

Parameterized Complexity Research Unit, University of Newcastle, Australia; Max-Planck-Institute for Informatics, Saarbrucken, Germany; University of New South Wales, Sydney, Australia

© TemplatesWise.com

Itinerary

1. General MAD-CSP
2. MAD-CSP with convex domains
3. MAD-CSP with convex constraints
4. MAD-CSP bi-convex
5. Summary of parameterized results
6. Lem 2.1 MAD-CSP is NP-complete, convex domain
7. Constraint graph
8. Thm 2.2 CSP solved in $m^{t+1} n^{O(1)}$ where t is the treewidth of the constraint graph (using Gottlob et al.)
9. Parameter: *Number of Values*
10. Parameter*: Maximum domain size*

Open Problem

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

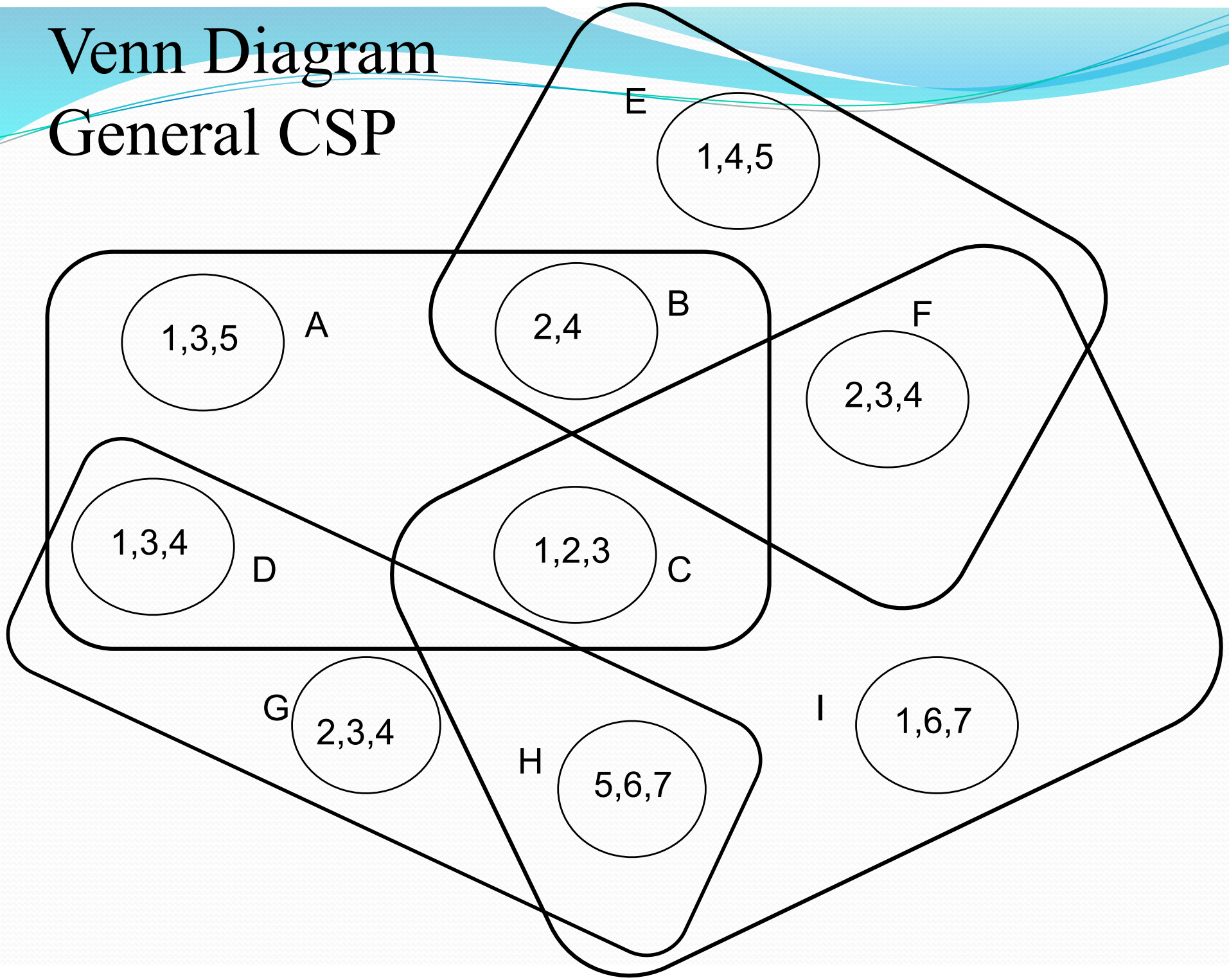
- $\mathcal{X}$, set of **variables**
- $\mathcal{U}$, universe of possible **values**
- $\mathcal{D}$, assigns **domains** to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, **All-Diff constraints**: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

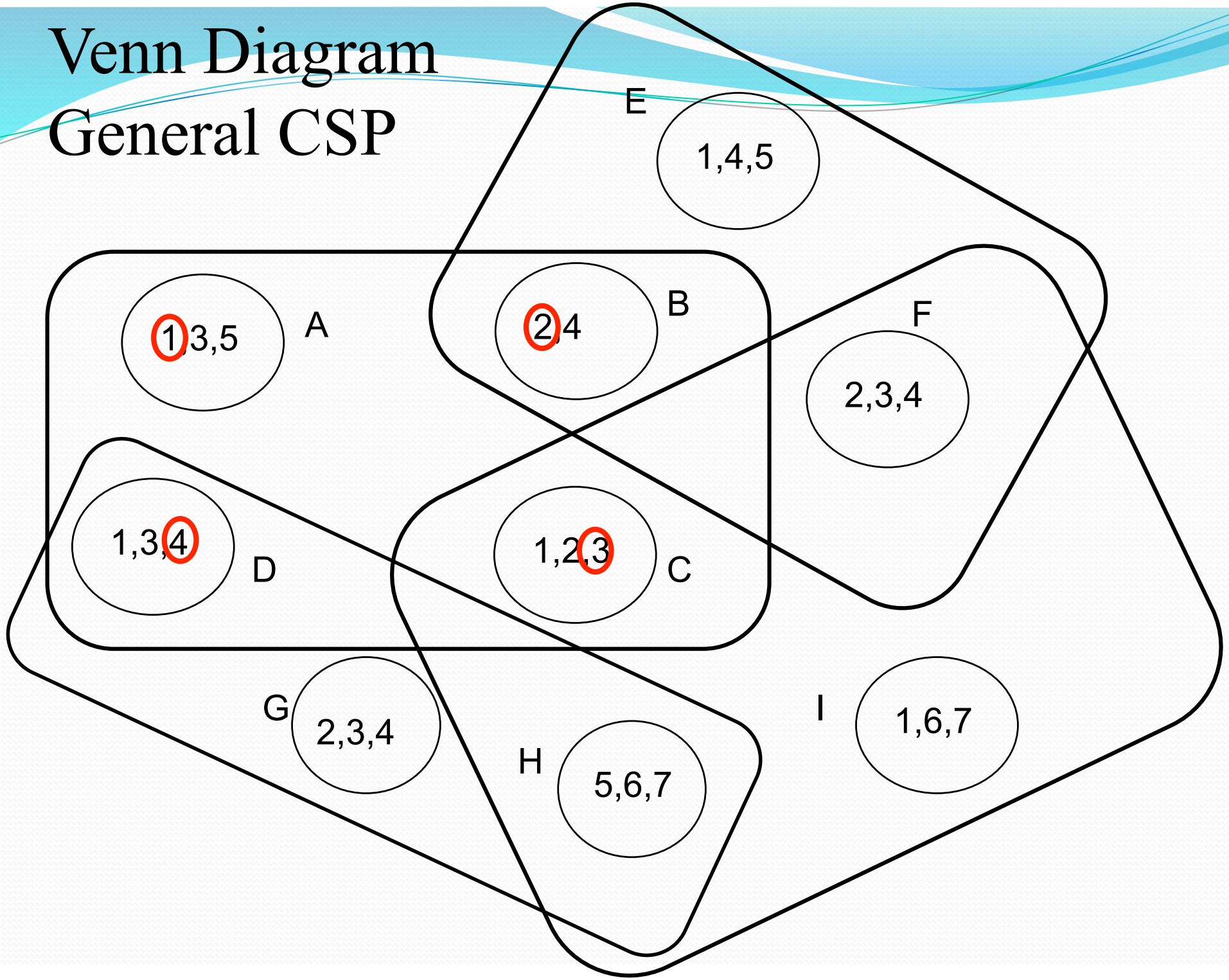
(i) $\forall x \in \mathcal{X}, \tau(x) \in \mathcal{D}(\mathcal{X})$

(ii) $\forall\, C \in \mathcal{C}$ and $\forall x,y \in C$: $x \neq y \Rightarrow \tau(x) \neq \tau(y)$

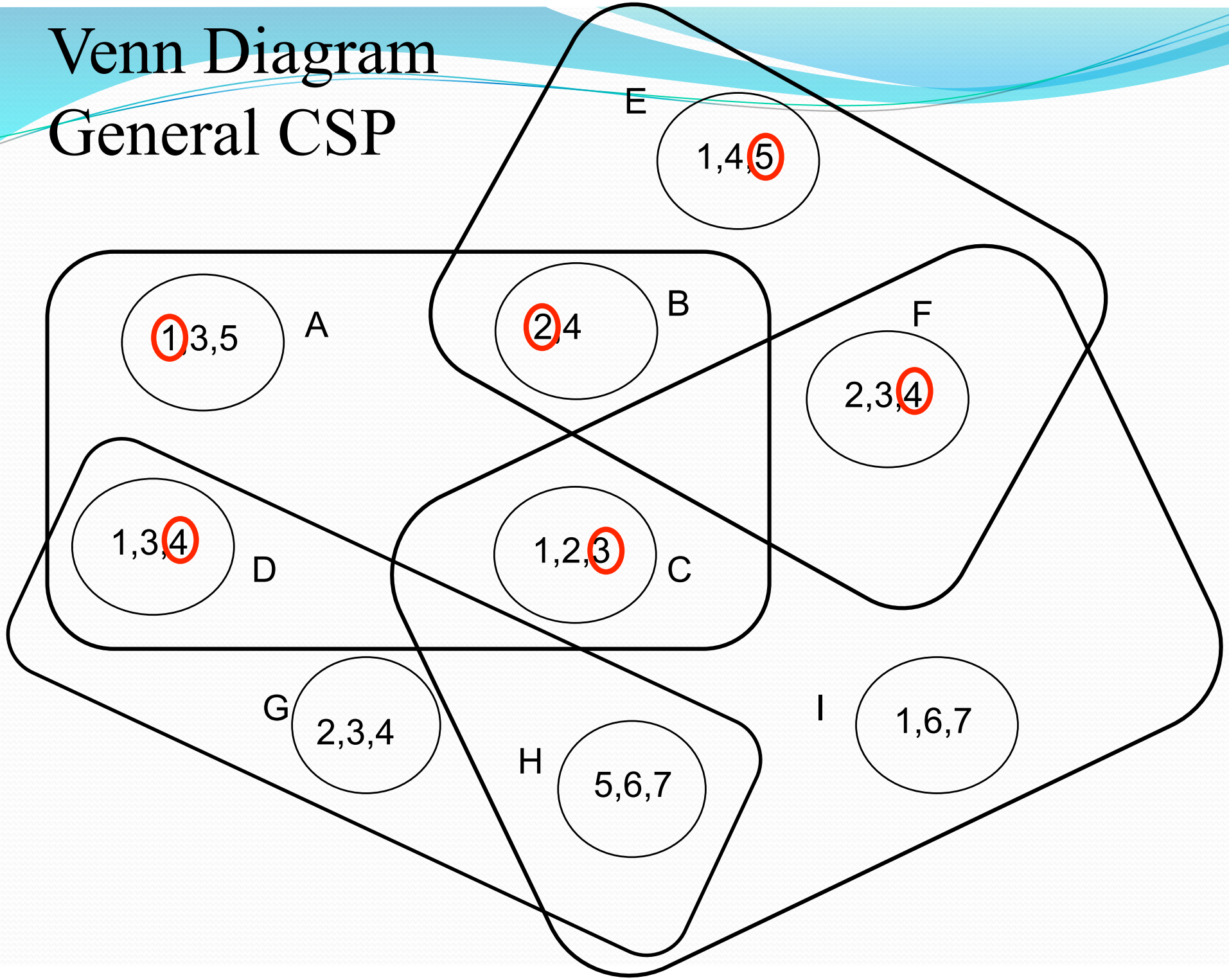
Venn Diagram
General CSP



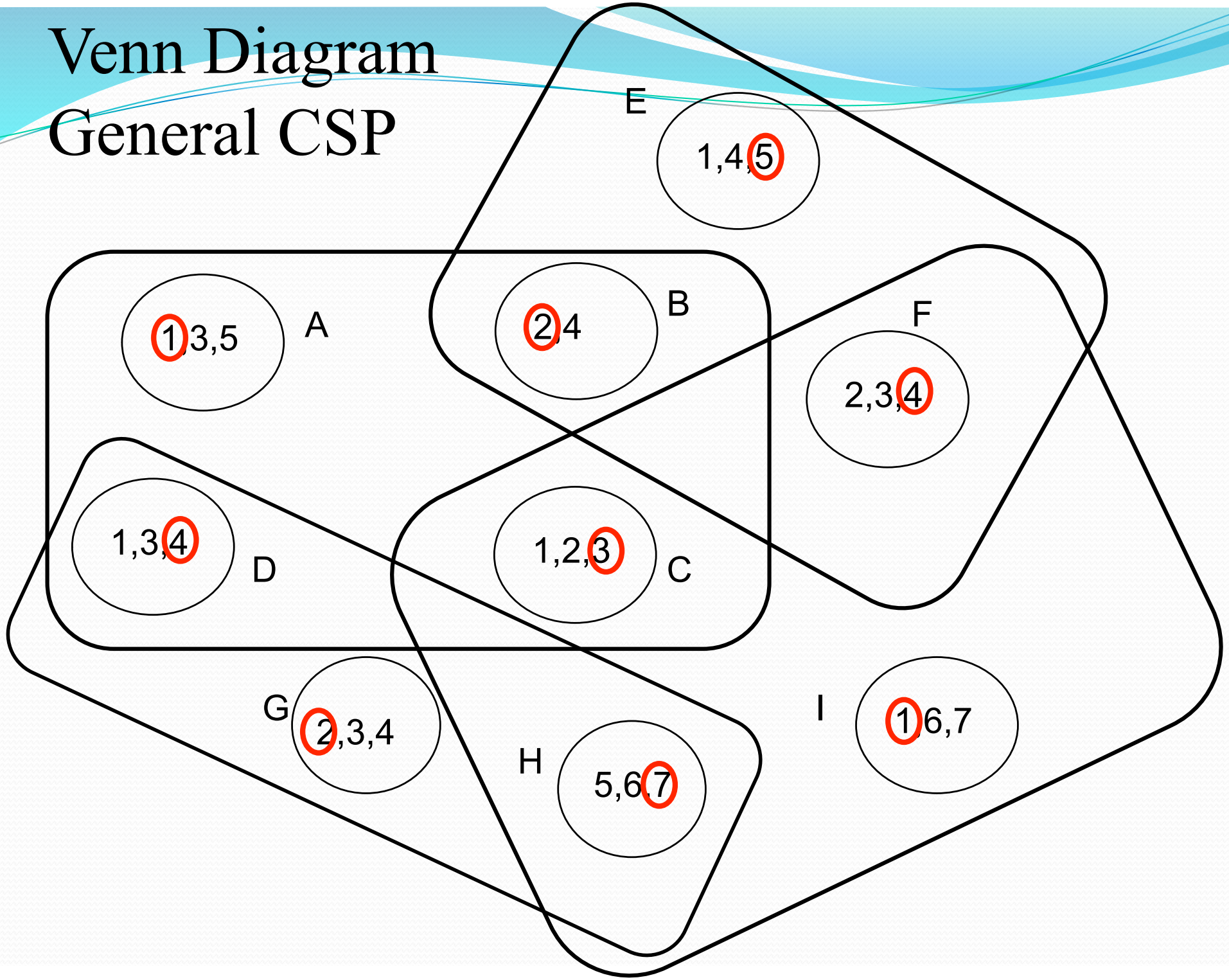
Venn Diagram
General CSP



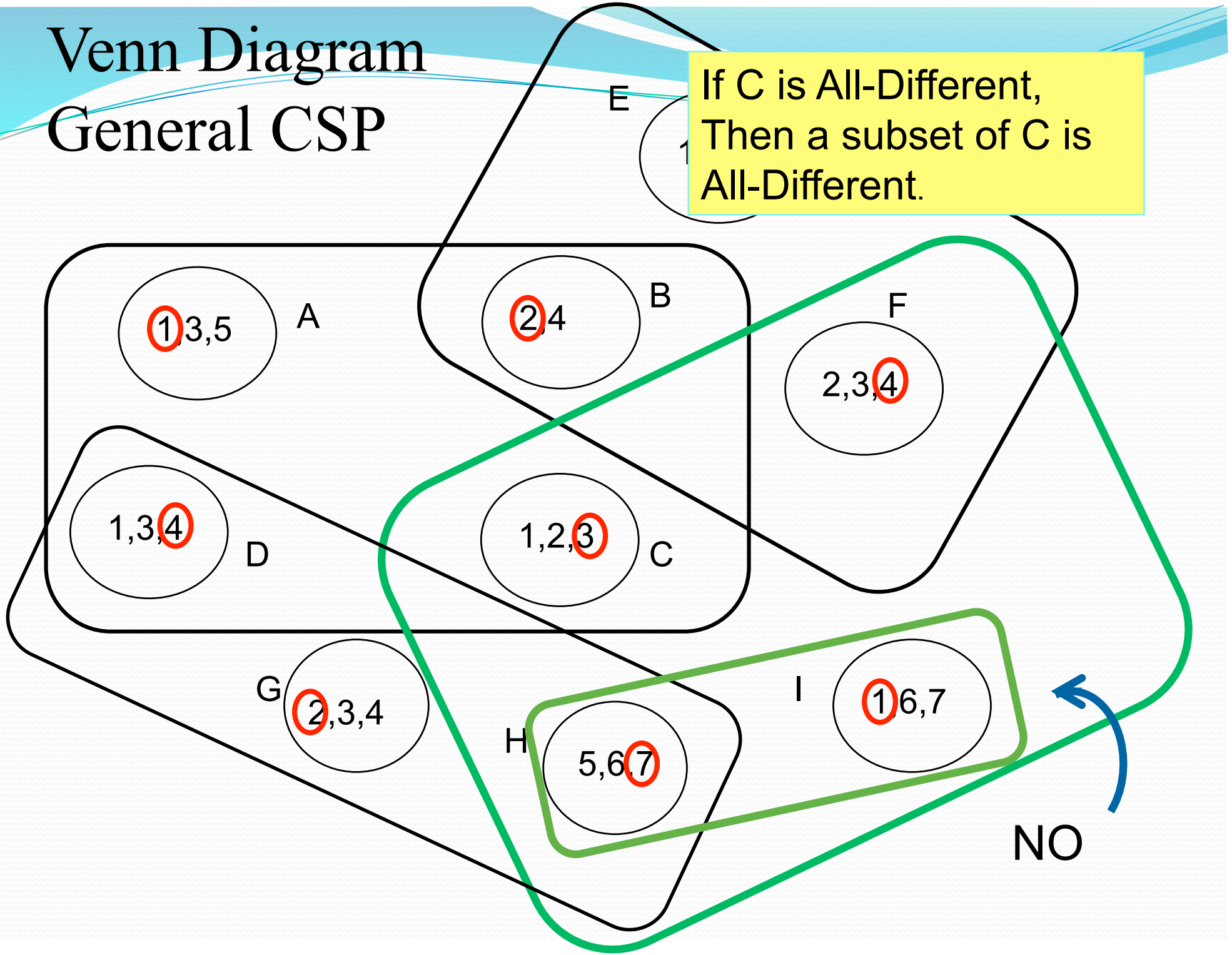
Venn Diagram
General CSP



Venn Diagram
General CSP



Venn Diagram
General CSP



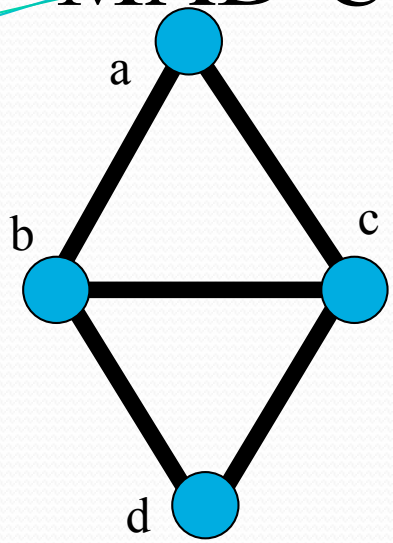
MAD-CSP is a quadruple ($\mathcal{X}$, $\mathcal{U}$, $\mathcal{D}$, $\mathcal{C}$)

1. No constraint is a proper subset of another
2. No empty constraints
3. No value which is not used by any variable
4. No variable with an empty domain

- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible
- $\mathcal{D}$, assigns domains to
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A MAD-CSP is a conjunction of an arbitrary number of AllDiff constraints.

MAD-CSP models GRAPH COLORING

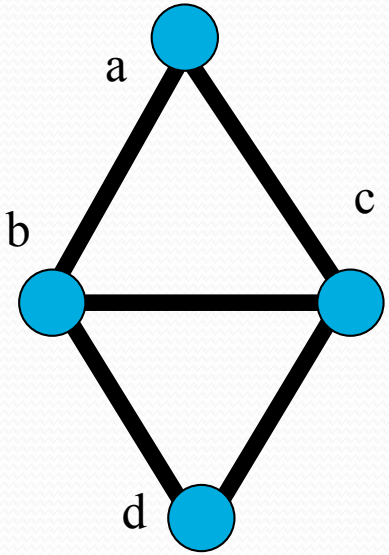


Input: ***G (V, E), k***

Output: A *k*-coloring e.g., $f: V \rightarrow \{1, 2, \ldots k\}$ with $f(u) \neq f(v), \forall \{u, v\} \in E$.

X: Variables are the vertices: a, b, c, d

MAD-CSP models GRAPH COLORING



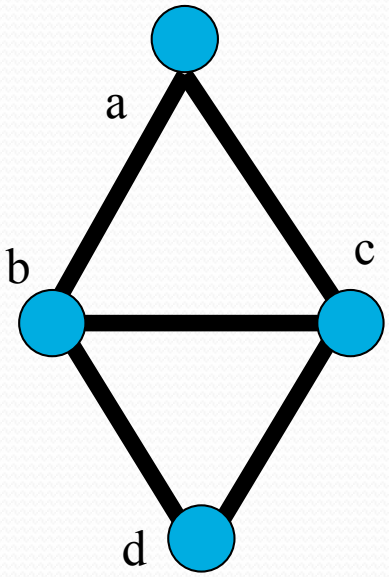
Input: ***G (V, E), k***

Output: A *k*-coloring e.g., $f: V \rightarrow \{1, 2, \ldots k\}$ with $f(u) \neq f(v), \forall \{u, v\} \in E$.

$\mathcal{X}$: Variables are the vertices: a, b, c, d

Domain of each variable are the three colors
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: ***G (V, E), k***

Output: A *k*-coloring e.g., $f: V \rightarrow \{1, 2, \ldots k\}$ with $f(u) \neq f(v), \forall \{u, v\} \in E$.

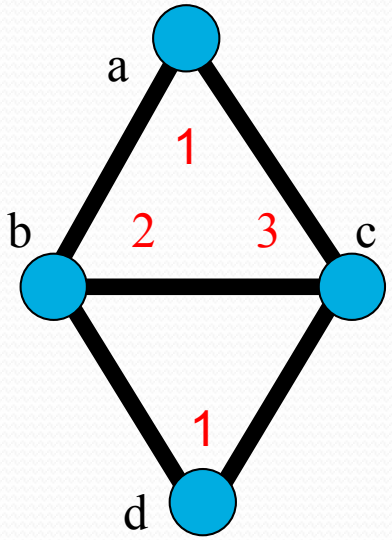
$\mathcal{X}$: Variables are the vertices: a, b, c, d

Domain of each variable are the three colors
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

All Different Constraints are the edges E
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

MAD-CSP is satisfiable iff G has a *k*-coloring.

MAD-CSP models GRAPH COLORING



Input: ***G (V, E), k***

Output: A *k*-coloring e.g., $f: V \rightarrow \{1, 2, \ldots k\}$ with $f(u) \neq f(v), \forall \{u, v\} \in E$.

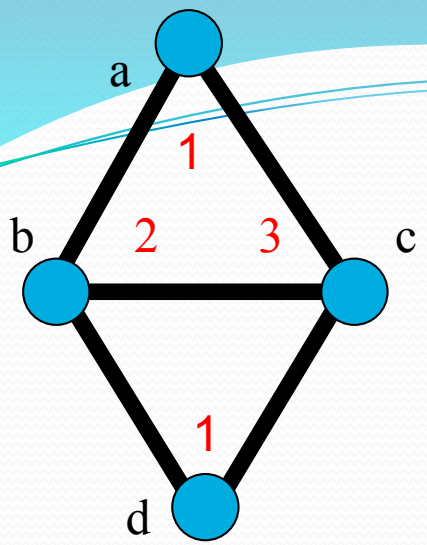
$\mathcal{X}$: Variables are the vertices

Domain of each variable are the three colors
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

All Different Constraints are the edges E
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f \mid_C$ is injective.

CONVEXITY



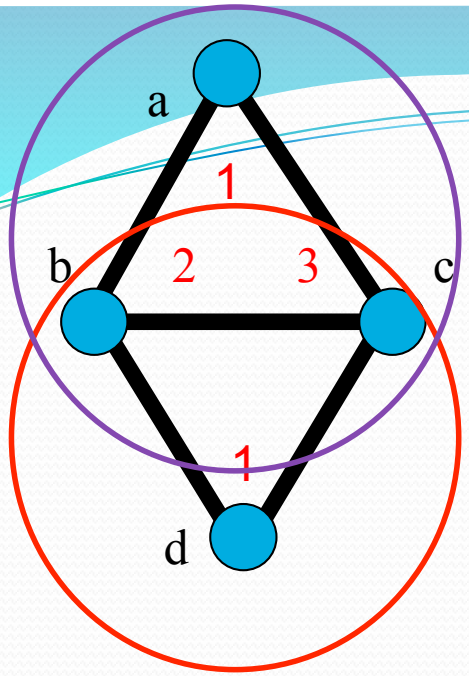
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An instance has ***domain convexity*** if the values are ordered and each domain is an **interval** of the set of values. e.g., $\mathcal{U} = \{1, 2, 3\}$ and each $D(x) = \{1, 2, 3)$.

An instance has constraint convexity if $C \in \mathcal{C}$ is *convex* if the variables are an interval.

NOT constraint convex

CONVEXITY



Two Constraints

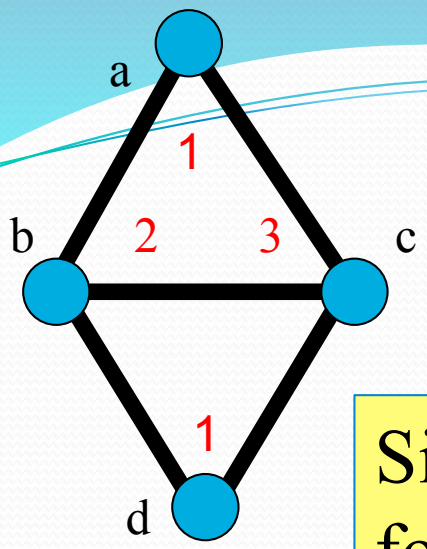
C_1 = {a, b, c}
C_2 = {b, c, d}

An instance has ***domain convexity*** if the values are ordered and each domain is an **interval** of the set of values. e.g., $\mathcal{U}$ = {1, 2, 3} and each D (x) = {1, 2, 3).

An instance has ***constraint convexity*** if C ∈ $\mathcal{C}$ is *convex* if the variables are an interval, e.g., $\mathcal{C}$ = {a, b, c, d} and C_1 and C_2.

A MAD-CSP instance is *bi-convex* if both domains and constraints are convex simultaneously.

CONVEXITY



Since GRAPH COLORING is NP-complete for instances with number of colors $k \geq 3$, we have

Lemma 2.1 MAD-CSP is NP-complete even when restricted to **convex domain** instances with size of the
universe $|\mathcal{U}| = m = 3$ *colors*,
maximum constraint size $= k = 2$ (edges), and
maximum domain size $= \ell = 3$.

Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible values
- l = maximum domain size
- k = maximum constraint size

Fixed Parameter Tractable (FPT)

A Decision Problem is FPT if it can be computed in time $O(f(k)n^c)$

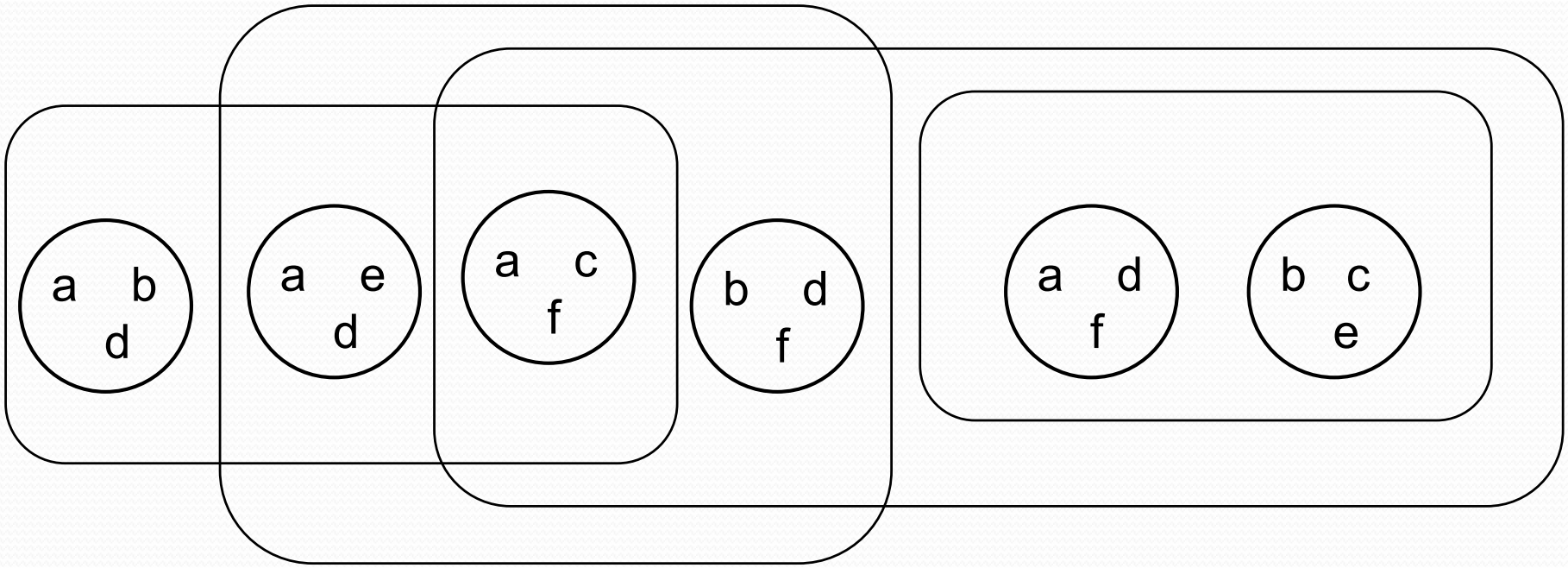
- Instance of size n
- Parameter k
- Arbitrary computable function f
- Constant independent of k

Note that FPT means that each 'slice' of the problem (obtained by fixing k to a constant) is decidable in polynomial time.

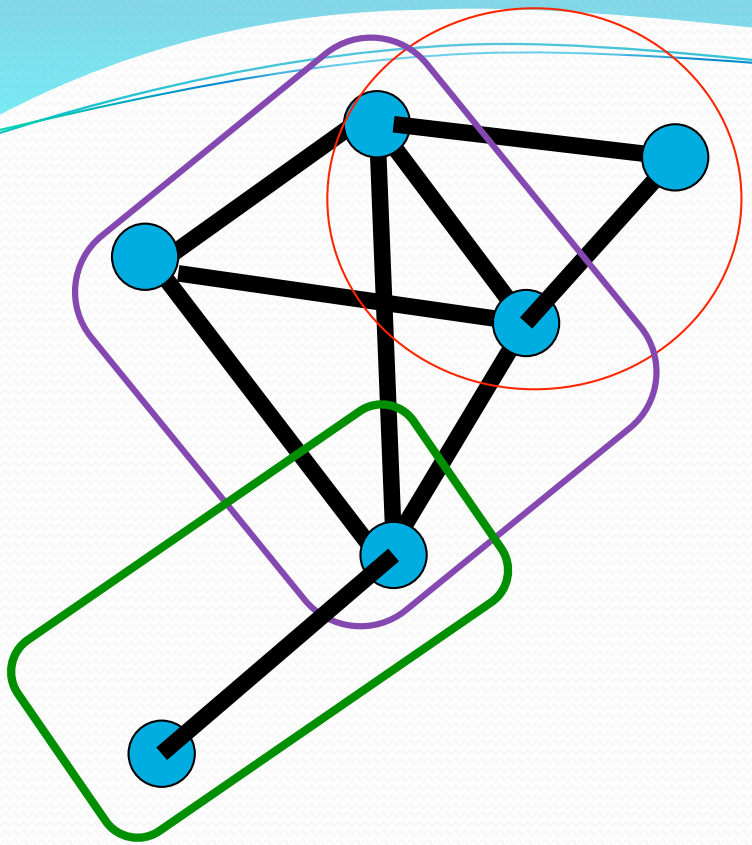
Parameter	n = number of variables	m = size of the universe	k = maximum constraint size	l = maximum domain size
No convexity	FPT	Para-NPC	Para-NPC	Para-NPC
Convex domains	FPT	Para-NPC	Para-NPC	Para-NPC
Convex constraints	FPT	FPT	FPT	Para-NPC
Bi-convex	FPT	FPT	FPT	OPEN

LIST COLORING with convex lists, parameter is the list size and bi-convex is LCIG.

We are not talking about the similar INTERVAL CONSTRAINED COLORING PROBLEM



The objective of ICCP is to assign a color to a set of variables which specifies exactly the number of variables that belong to each color class. Not **All-Different**.

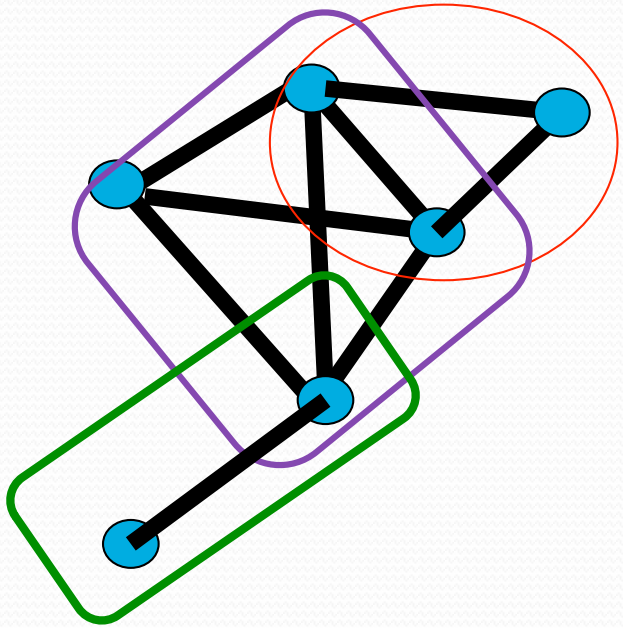


Constraint Graph

Constraint Graph is the underlying hypergraph $(\mathcal{X}, \mathcal{C})$.

Two variables are connected by an edge iff there is a constraint containing both of them.

Constraint Graph



Thm 2.2 CSP can be solved in $m^{t+1} n^{O(1)}$ where t is the treewidth of the constraint graph; e.g., linear for all fixed t, linear time FPT.

G.Gottlob, F. Scarcello, M. Sideri. *Fixed-Parameter complexity in AI and nonmonotonic reasoning.* AI 2002.

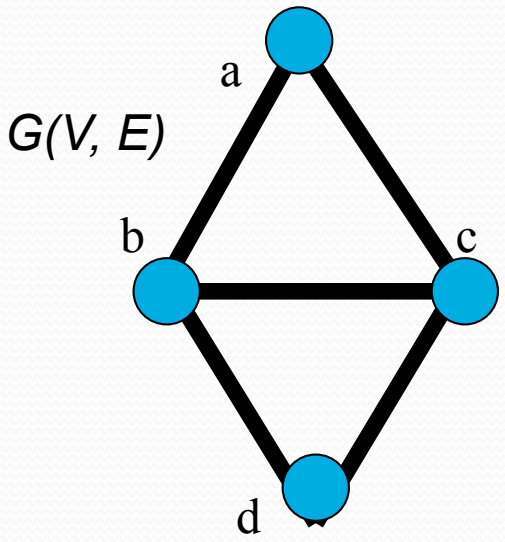
Parameter: *Number of Values* $m := |\mathcal{U}|$

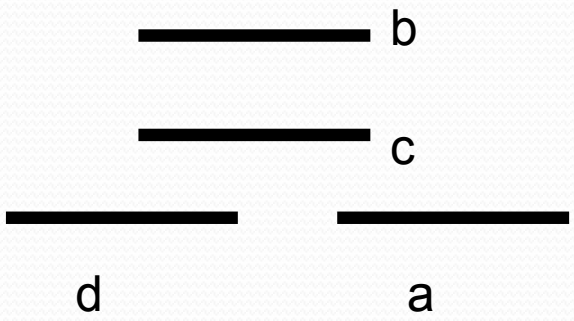
MAD-CSP with convex constraints can be solved in time $m^{m+1} n^{O(1)}$

1. MAD-CSP with convex constraints is equivalent to LCIG.
2. If there is any constraint of size greater than m, then the given MAD-CSP instance cannot be satisfied.
3. If any constraint in a given MAD-CSP instance with convex constraints is of size at most k, then the constraint graph of the instance has pathwidth at most $k - 1$.
4. Finally, directly use Theorem 2.2.

MAD-CSP with convex constraints is equivalent to the LIST COLORING INTERVAL GRAPHS (LCIG) problem.

- **Interval graph**: *G(V, E)* is an interval graph if there exists a set of intervals of the real line with a 1-1 bijection *I* between *V* and the intervals such that $\{u, v\} \in E \Leftrightarrow I(u) \cap I(v) \neq \varnothing$

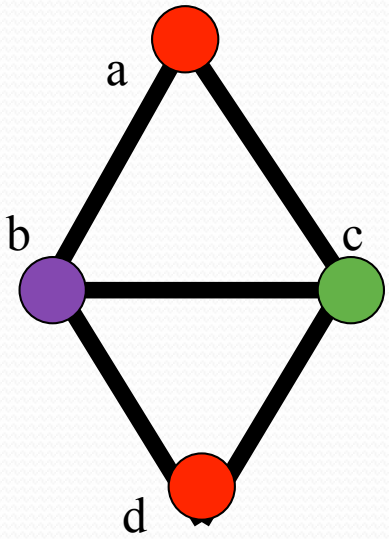




Given an interval graph, Linear time to find its interval representation. K.S. Booth, G.S. Lueker 1976.

MAD-CSP with convex constraints is equivalent to the LIST COLORING INTERVAL GRAPHS (LCIG) problem.

- **LCIG:** Given an interval graph *G*, each vertex *v* has a list of colors *L(v)*. Goal is to assign a color to each vertex from its list such that adjacent vertices are assigned different colors.



$L(a) = L(b) = L(c) = L(d) = \{1, 2, 3\}$

The constraint graph of an arbitrary MAD-CSP with convex constraints is an interval graph.

1. A given graph is an interval graph iff its corresponding maximal-clique matrix has the *consecutive 1s* property. [D R. Fulkerson, O.A.Gross. 1965]

2. The set of constraints is the set of all maximal cliques in *G*. (No constraint is properly contained in another constraint).

3. The maximal-clique matrix of our constraint graph *G* has the consecutive 1s property. Therefore, our constraint graph is an interval graph.

The constraint graph of an arbitrary MAD-CSP with convex constraints is an interval graph.

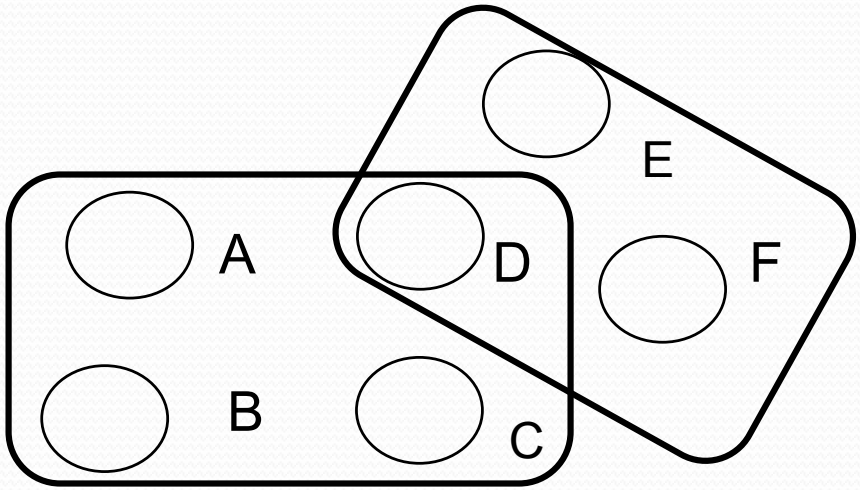
1. A given graph is an interval graph iff its corresponding **maximal-clique matrix** has the ***consecutive 1s*** property. [D R. Fulkerson, O.A.Gross. 1965]

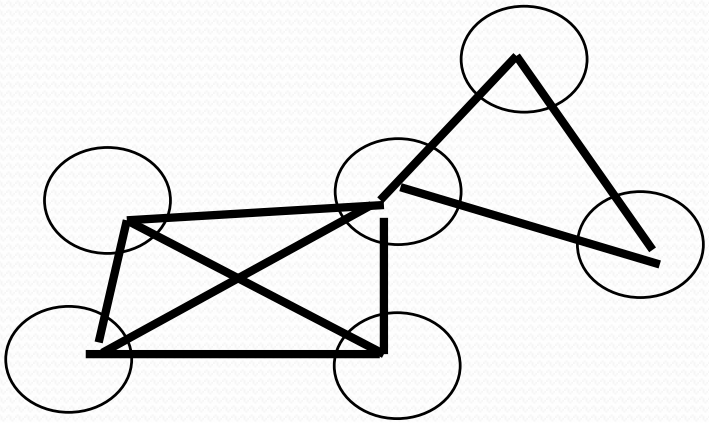
2. The set of constraints is the set of all maximal cliques in *G*. (No constraint is properly contained in another constraint).

3. The maximal-clique matrix of our constraint graph *G* has the consecutive 1s property. Therefore, our constraint graph is an interval graph.

1. The set of constraints $\mathcal{C}$ is the set of all maximal cliques in *G*.

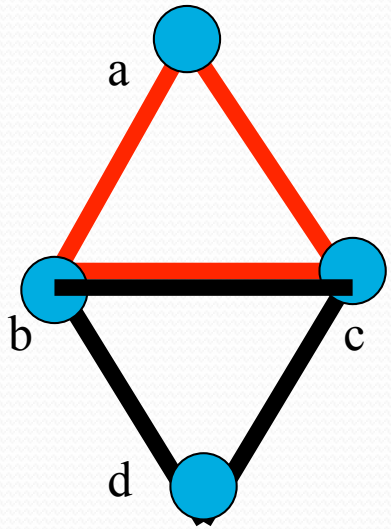
Let C be an arbitrary constraint. All the variables in C have an edge between them in the constraint graph, forming a clique. Each constraint corresponds to a maximal clique in *G*.





Conversely, a maximal clique in *G* must correspond to a constraint due to the convexity of the constraints.

2. Maximal Clique Matrix and consecutive 1s



$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \{a,b,c\} \quad \{b,c,d\} \\ \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \end{array}$$

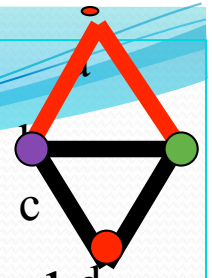
Maximal-Clique Matrix:
Row for each vertex, column for each maximal clique.

Consecutive 1s property: columns can be permuted so that the 1s appear consecutively in each row.

A graph is an interval graph iff its maximal-clique graph has the consecutive 1s property. D R. Fulkerson,O.A.Gross. 1965.

The maximal cliques of our constraint graph *G* is precisely $\mathcal{C}$, and C are intervals over χ, so the maximal-clique matrix of *G* has the consecutive 1s property, so the **constraint graph of a MAD-CSP with convex constraints is an interval graph**.

Thm 4.1 MAD-CSP with convex constraints is equivalent to the LCIG problem.



- The constraint–graph is an interval graph. Set the domain of each variable as its list of colors.
- A LCIG instance has a valid list-coloring iff MAX-CSP is satisfiable since a list coloring is valid iff each vertex in any maximal clique is assigned a different color.

Lem 4.3 If there is any constraint $C \in \mathcal{C}$ of size greater than *m*, then the given instance cannot be satisfied.

- Any satisfying assignment of the given MAD-CSP instance must assign all variables in C different values, this would not be possible if |C| greater than the number of values altogether.

Lemma 4.4 If any constraint in a given MAD-CSP instance with convex constraints is at most k, then the constraint graph of the instance has pathwidth at most $k - 1$.

- The constraint graph is an interval graph.
- A graph has pathwidth k iff it is a subgraph of an interval graph with maximum clique-size $k + 1$.
- Pathwidth of a given graph is always bounded from below by its treewidth. Apply Thm 2.2

Thm 4.5 MAD-CSP with convex constraints can be solved in $m^{m+1} n^{O(1)}$ time; e.g. linear for all fixed m, linear time FPT.

Parameter:

Maximum Domain Size $\ell := \max_{x \in \chi} |\mathcal{D}(x)|$

Th 5.5 MAD-CSP with convex domains can be solved in $\ell^{2cl^2} n^{O(1)}$ time, where c is the number of constraints in the given instance and ℓ is the maximum domain size.

Th 5.5 MAD-CSP with convex domains can be solved in $\ell^{2c\,\ell\,\hat{}2}\, n^{O(1)}$ time, where c is the number of constraints and ℓ is the maximum domain size.

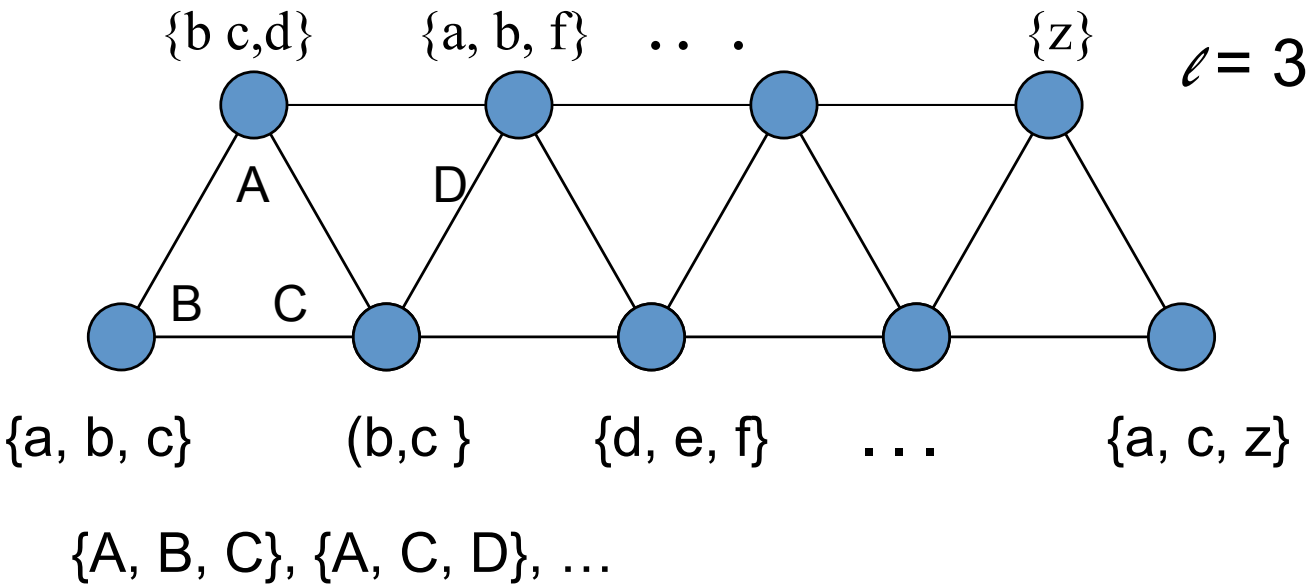
1. Lem 5.2 MAD-CSP can be solved in $\ell^{t+1}\, n^{O(1)}$ where t is the tw of the **constraint** graph, no convexity assumptions.
2. Def 5.1 *Domain graph* of a MAD-CSP
3. Lem 5.3 MAD-CSP can be solved in $\ell^{t+1}\, n^{O(1)}$ where t is the tw of the **domain** graph, no convexity assumptions.
4. Bound the number of variables by $c\,\ell$, (c defined to be $|\mathbb{C}|$)
5. Finally, observe that domain graphs are interval graphs with maximum clique-size $\leq 2c\ell^2$, and therefore tw $\leq 2c\ell^2 - 1$. Now apply Lem 5.3.

Lem 5.2 MAD-CSP can be solved in $\ell^{t+1} n^{O(1)}$ where t is the treewidth of the *constraint* graph, no convexity assumptions.

Bounded pathwidth.

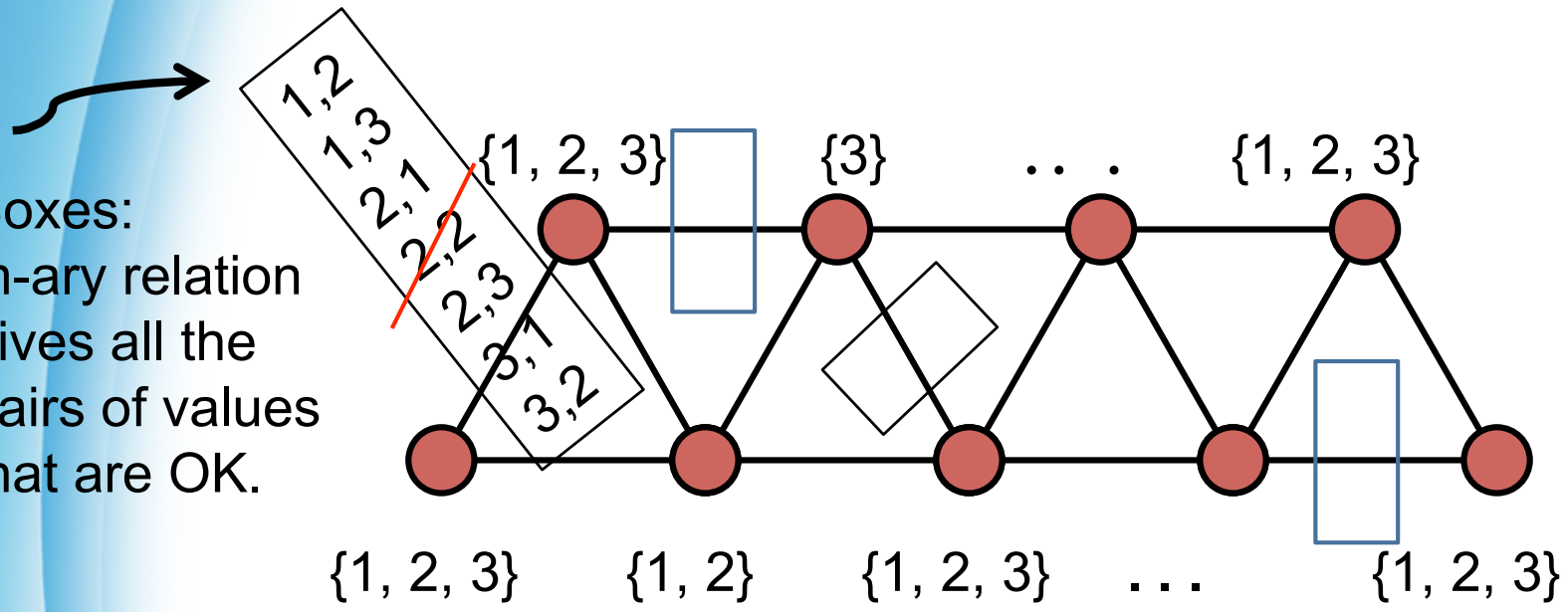
Edge means the vertices are being told they must receive different colors.

Size of the universe is unbounded. We cannot apply Thm 2.2 directly.



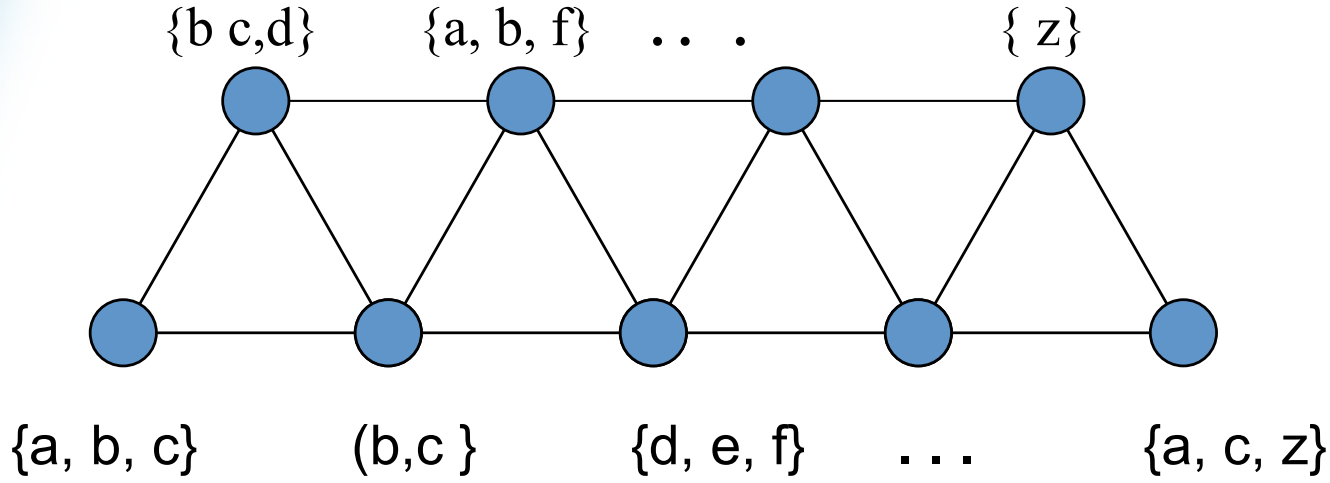
Must reduce to a general CSP with bounded universe size, and the treewidth of the constraint graph remains bounded.

A general CSP instance with max domain size $\ell = 3$ and bounded universe size (3).

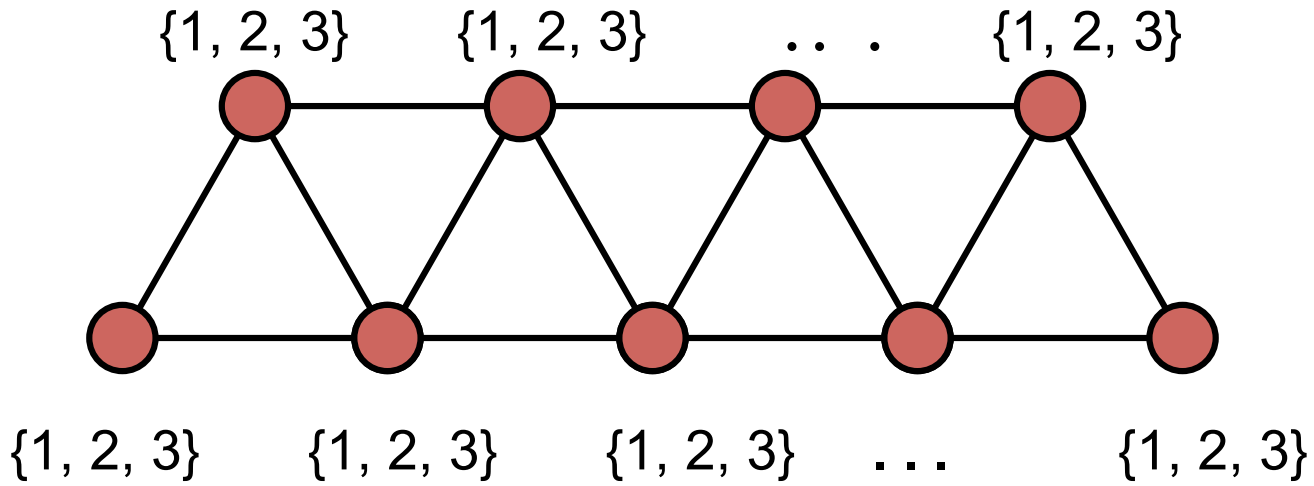


Reduce the unbounded universe size $|\mathcal{U}|$ to $|\mathcal{U}'| = \ell$. Code A into B.

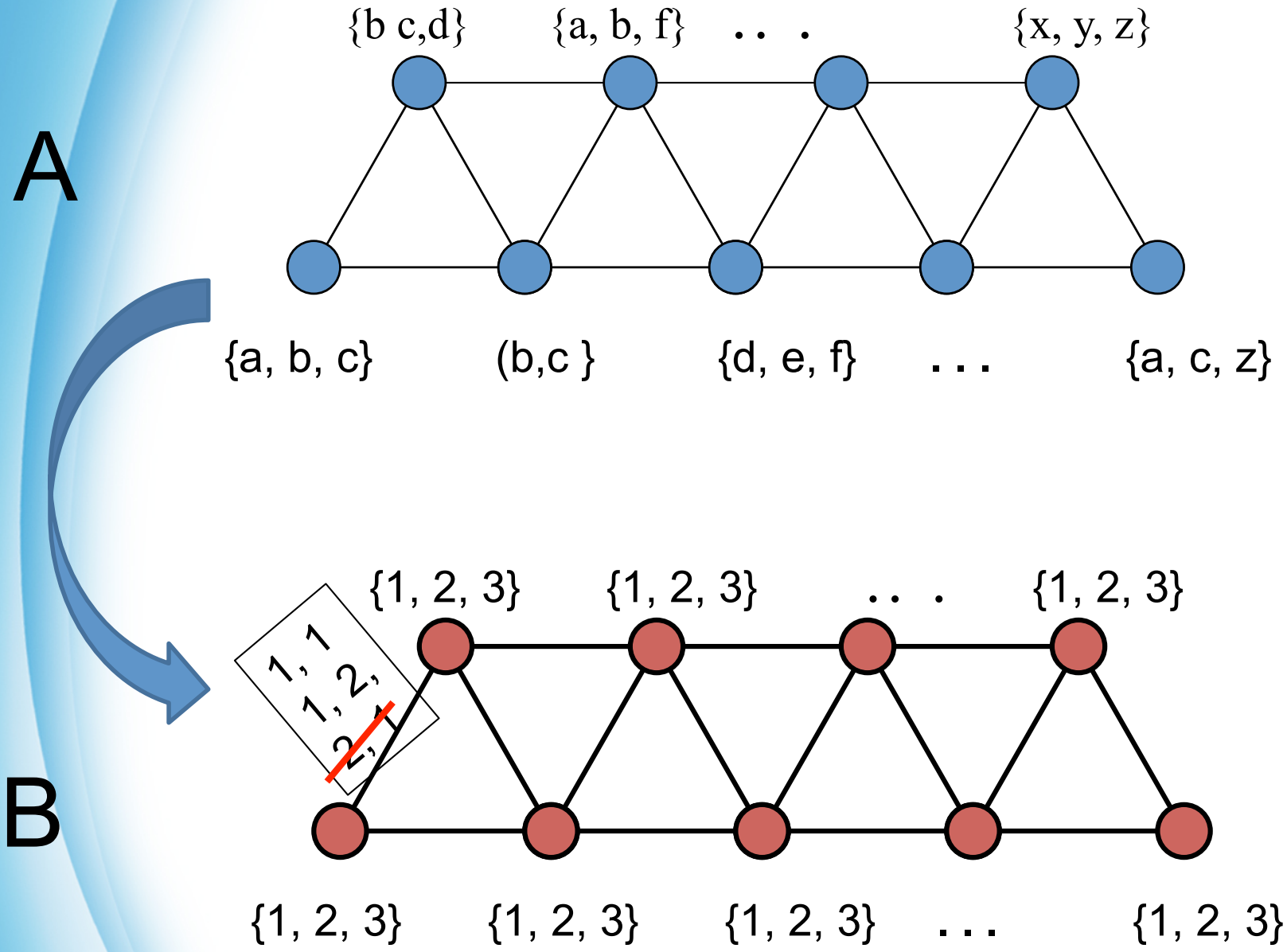
A



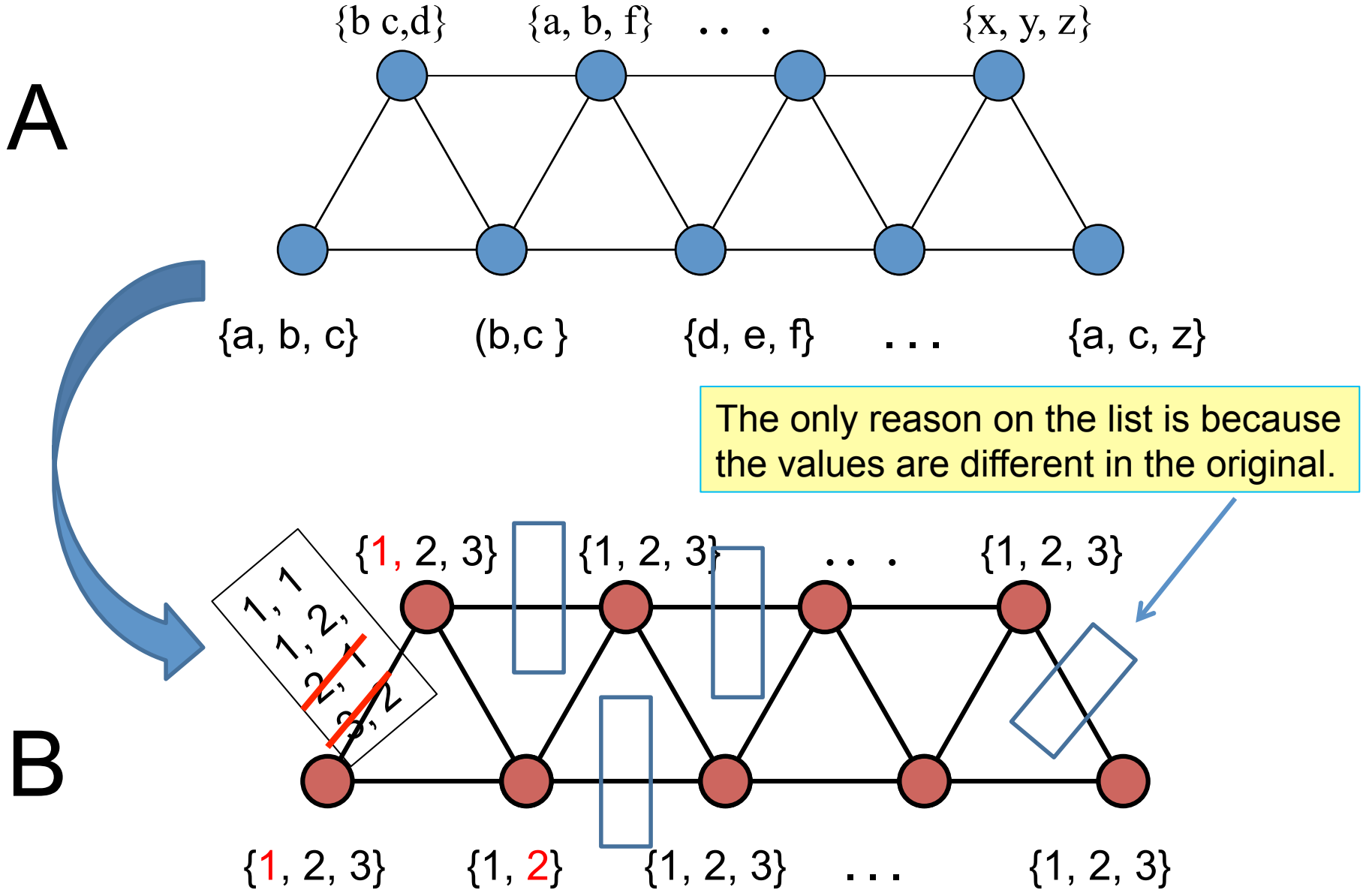
B



An edge (*x*, *y*) in the constraint graph can be formulated as a constraint in the general CSP.

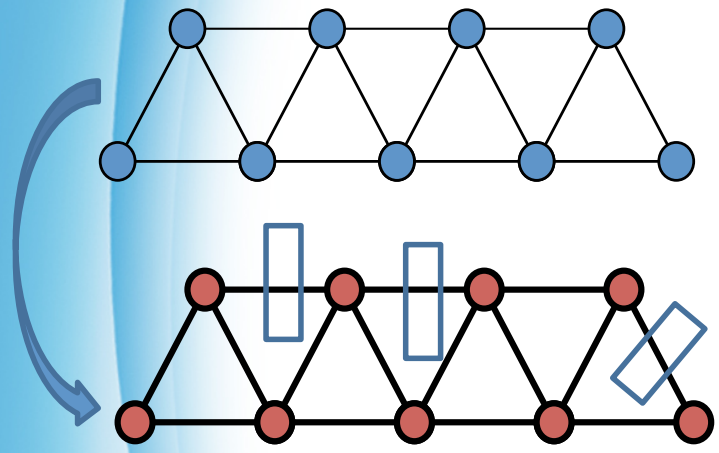


The new domains contain only the indices of the entries of the original domains.



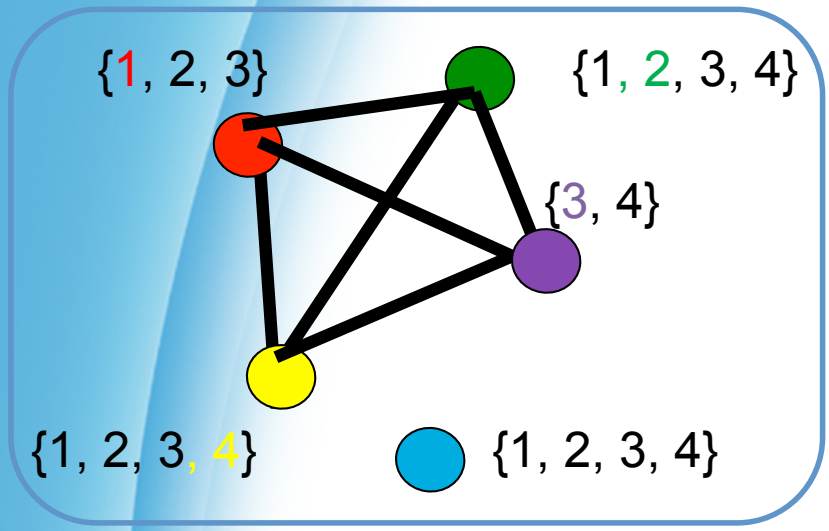
Every solution to the general CSP corresponds to a solution to the original unbounded universe MAD-CSP and vice versa. The constraint graphs of the two instances are isomorphic. The reduction can be carried out in polynomial time.

Now apply Thm 2.2.



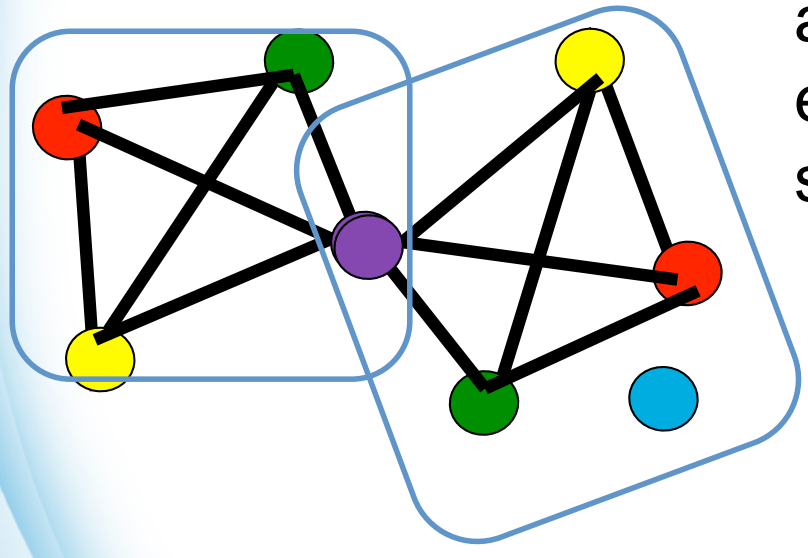
Lem 5.2 MAD-CSP can be solved in $\ell^{t+1} n^{O(1)}$ where t is the treewidth of the *constraint* graph, no convexity assumptions.

Bound the number of variables by ℓ | $\mathcal{C}$ |

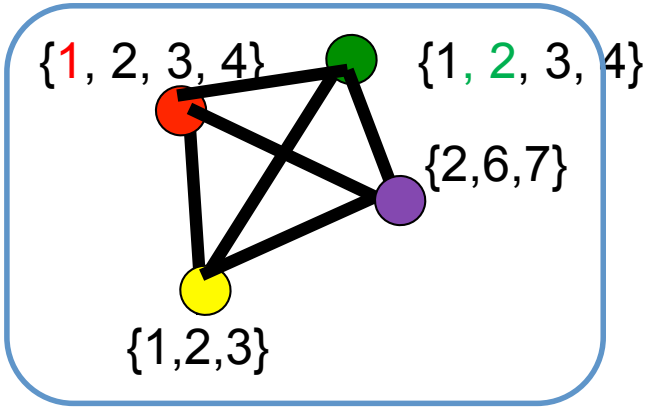


If ℓ is 4, an All-Diff constraint with five variables cannot be satisfied.

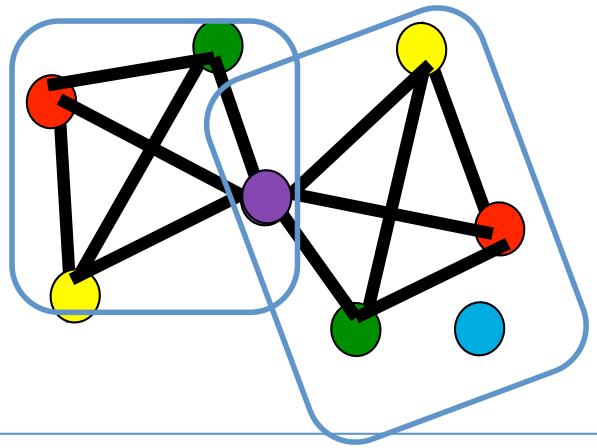
If ℓ is 4 and we have 2 constraints, eg. | $\mathcal{C}$ | = 2, then we cannot have 9 variables all having the same domain, else ($\mathcal{X}$, $\mathcal{U}$, $\mathcal{D}$, $\mathcal{C}$) cannot be satisfied. Pigeonhole.



Bound the number of variables by $\ell \mid \mathcal{C} \mid$



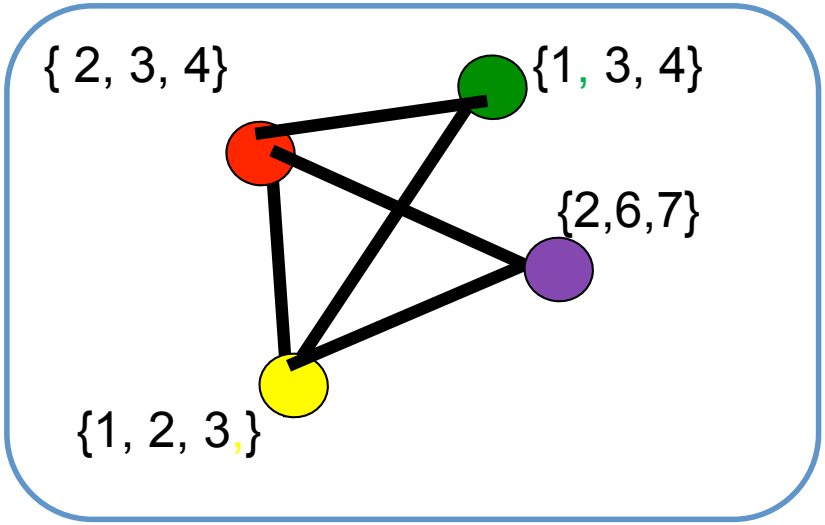
If ℓ is 4, an All-Diff constraint with five variables cannot be satisfied.



If ℓ is 4 and we have 2 constraints, eg. $\mid \mathcal{C} \mid = 2$, then we cannot have 9 variables all having the same domain, else $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$ cannot be satisfied. Pigeonhole.

Lem 5.4 $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$ and D be a domain of some variable in $\mathcal{X}$. If there are more than $\ell \mid \mathcal{C} \mid$ variables $x \in \mathcal{X}$ with $\mathcal{D}(x) = D$, then the instance $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$ cannot be satisfied.

Def 5.1 The *domain graph* of a MAD-CSP instance $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$ is a graph with vertex set $\mathcal{X}$ where two variables $x, y \in \mathcal{X}$ are adjacent iff $\mathcal{D}(x) \cap \mathcal{D}(y) \neq \emptyset$

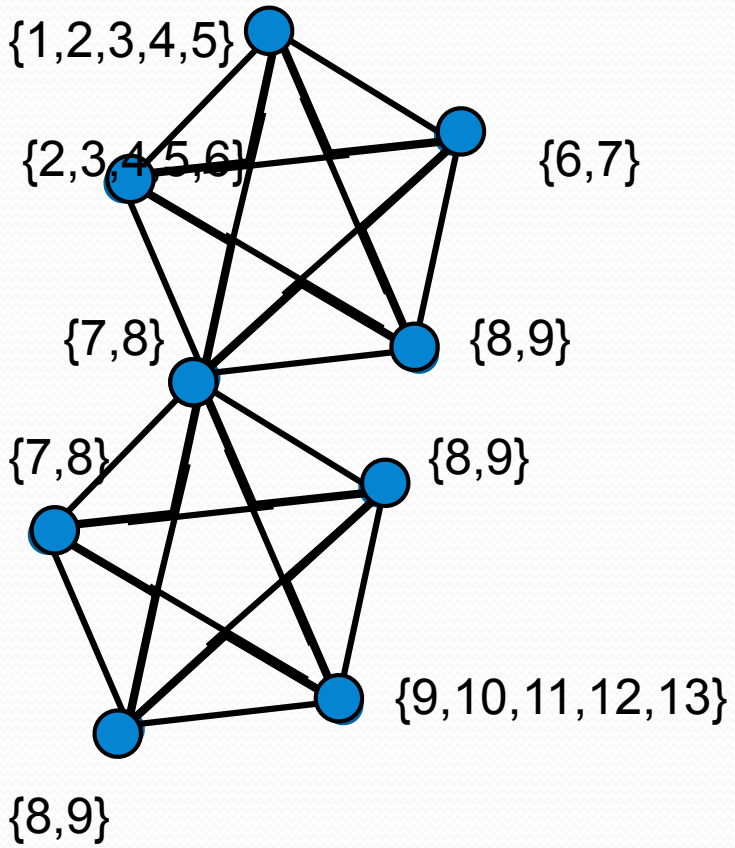


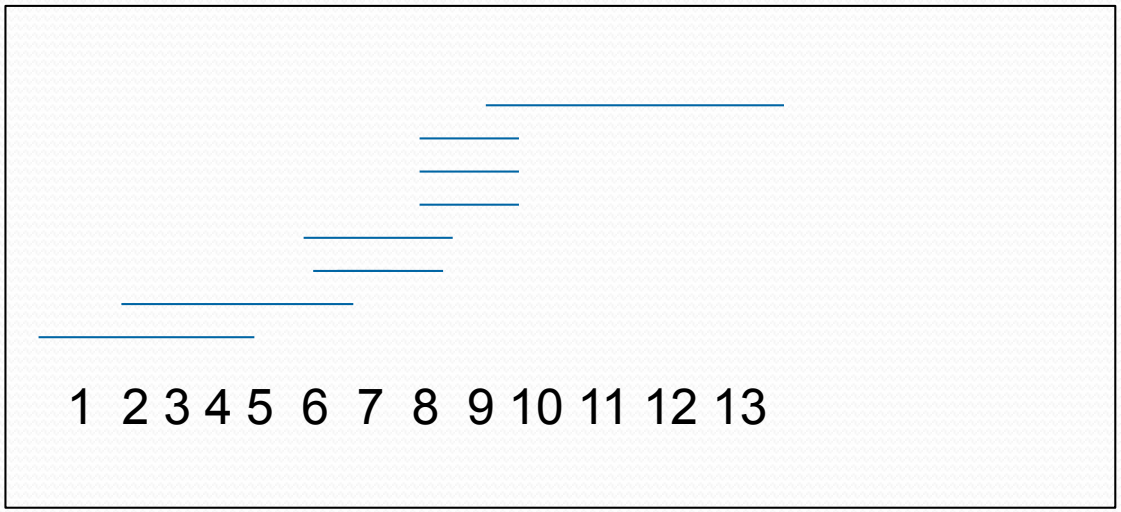
Lemma 5.3 MAD-CSP can be solved in $\ell^{t+1} n^{O(1)}$ where t is the treewidth of the domain graph, independent of any convexity assumption.

Proof:

- Reduce **MAD-CSP** to a general CSP instance.
- The constraint graph of the new instance is a subgraph of the domain graph of the original instance, and has treewidth at most t, since it is a subgraph of a graph with treewidth at most t.
- Apply Thm 2.2.

- The domain graph G of a MAD-CSP instance with convex domains is an interval graph (assign each variable its interval domain).



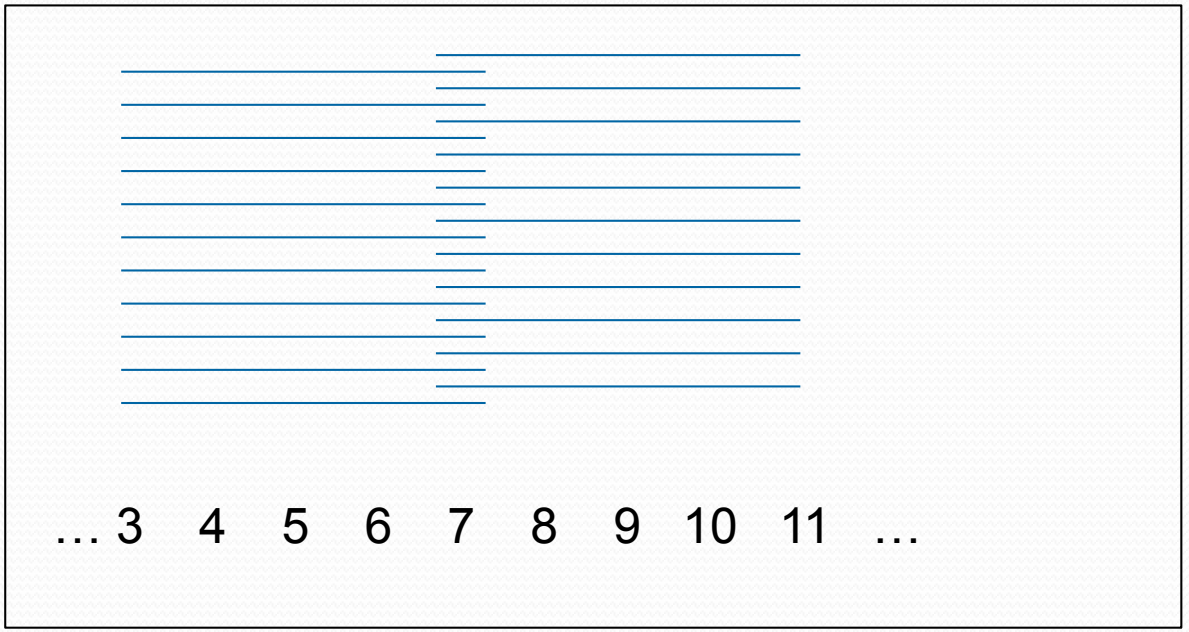


$\ell = 5$. Number of constraints is $|\mathcal{C}| = 2$.
Cannot be more than $\ell \, | \, \mathcal{C} \, | = 10$
variables with domain of size 5.

No domain is contained in another.

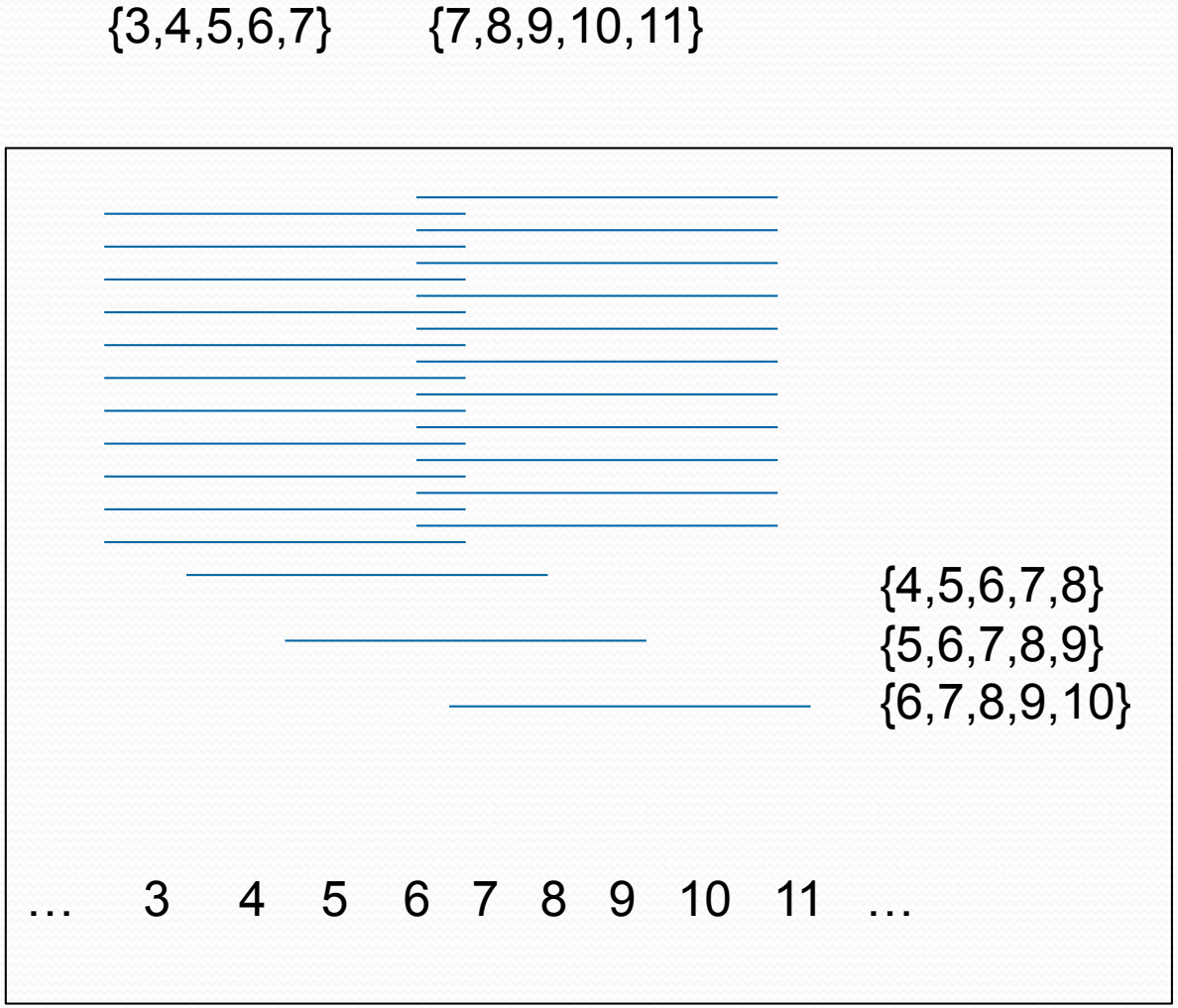
- The number of domains that start or end at a specific value is at most $\ell |C|$.

{3,4,5,6,7} {7,8,9,10,11} ℓ = 5, |C| = 2



- The number of domains that start or end at a specific value is at most $\ell |C| = c\,\ell$.

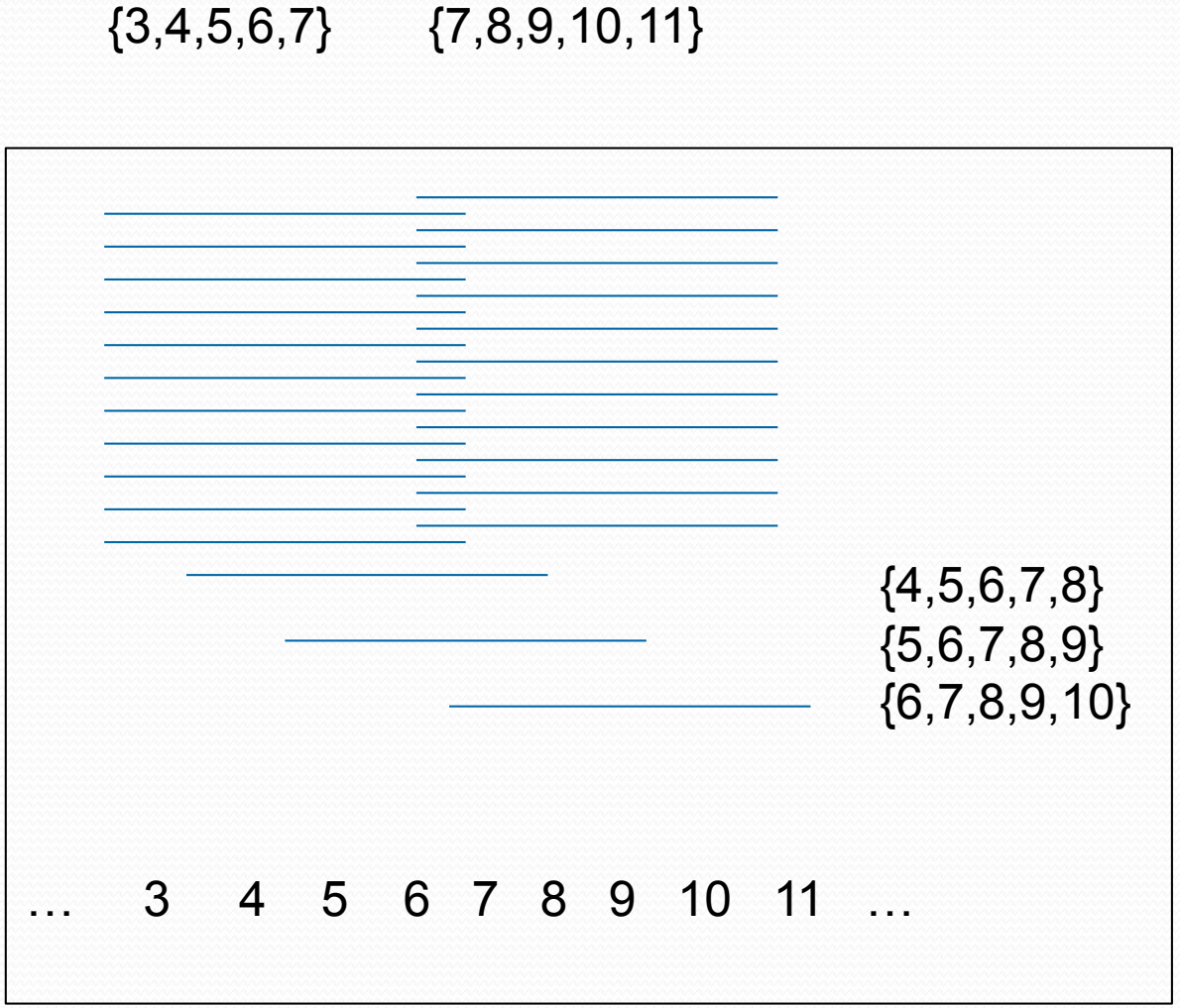
Any domain intersects less than $2\,c\,\ell^2$ other domains



- The number of domains that start or end at a specific value is at most $\ell |C| = c\,\ell$.

Any domain intersects less than $2\,c\,\ell^2$ other domains

Therefore G has maximum-clique-size less than $2\,c\,\ell^2$

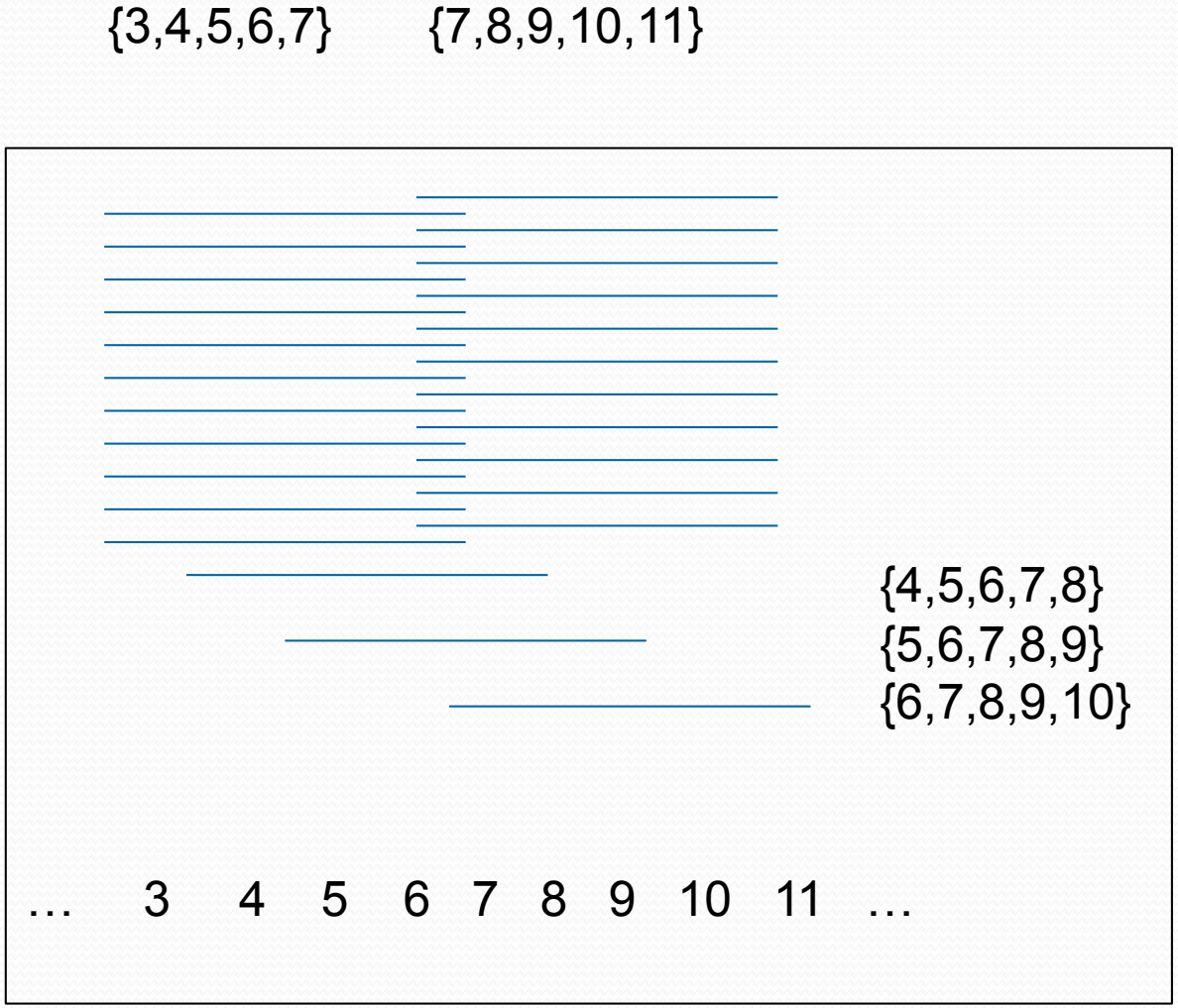


- The number of domains that start or end at a specific value is at most $\ell |C| = c\,\ell$.

Any domain intersects less than $2\,c\,\ell^2$ other domains

Therefore G has maximum-clique-size less than $2\,c\,\ell^2$

Therefore G has tw at most $2\,c\,\ell^2 - 1$.



Th 5.5 MAD-CSP with convex domains can be solved in $\ell^{2cd^2} n^{O(1)}$ time, where c is the number of constraints in the given instance.

Apply Lemma 5.3

Th 5.5 MAD-CSP with convex domains can be solved in $\ell^{2cd^2} n^{O(1)}$ time, where c is the number of constraints in the given instance.

We have shown that MAD-CSP is FPT parameterized by both the *maximum domain size* and the *number of constraints*, when the *domains are convex*.

The complexity of the bi-convex case for the single parameter ℓ = maximum domain size remains open.

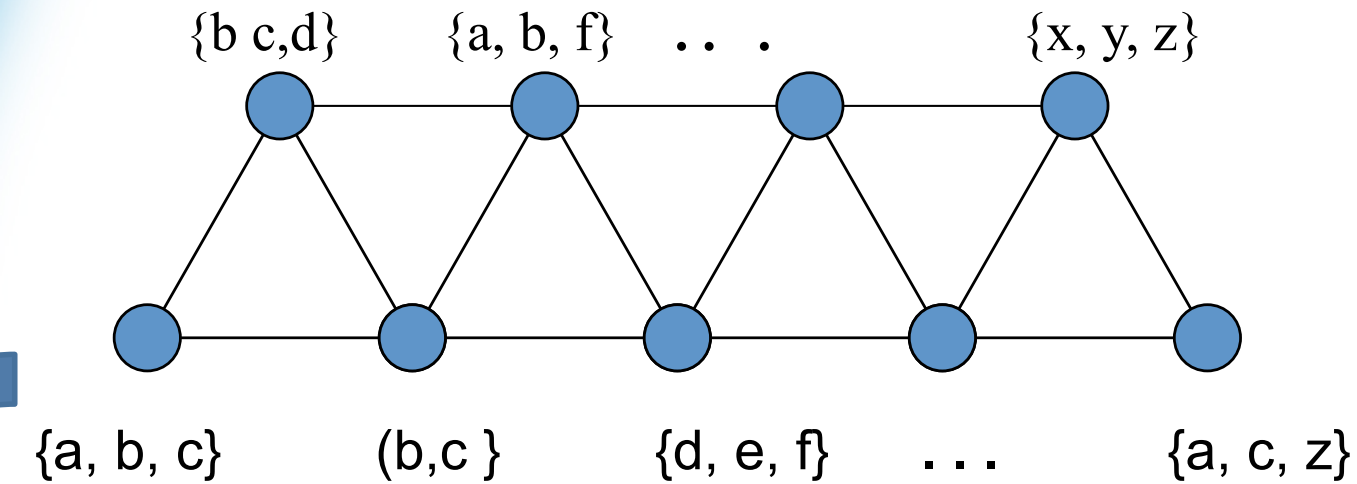
Thank You

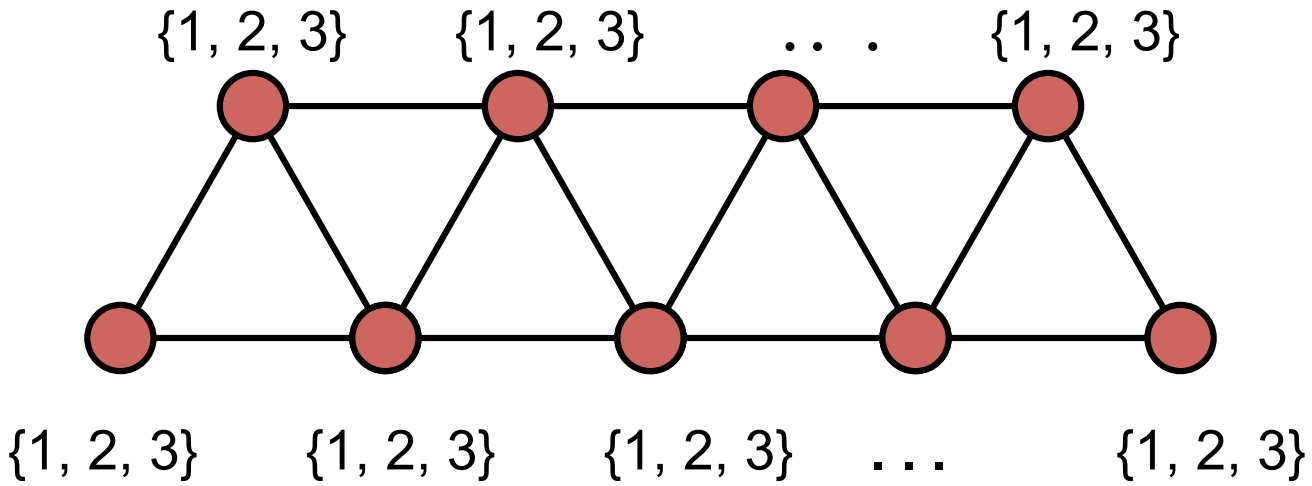
Parameter:

Maximum Domain Size $\ell := \max_{x \in \chi} |\mathcal{D}(x)|$

Lem 5.1 MAD-CSP with *convex constraints* is NP-complete for instances with $\ell \geq 3$.

- To show NP-hardness, the list coloring problem for graphs that are the union of two cliques is NP-complete, even when the list size of each vertex is at most 3. [Klaus Jansen. *The optimum cost chromatic partition problem*. (CIAC) 1997]
- A graph which is the union of two cliques is an interval graph with two maximal cliques, and by the above proven equivalence to the LCIG

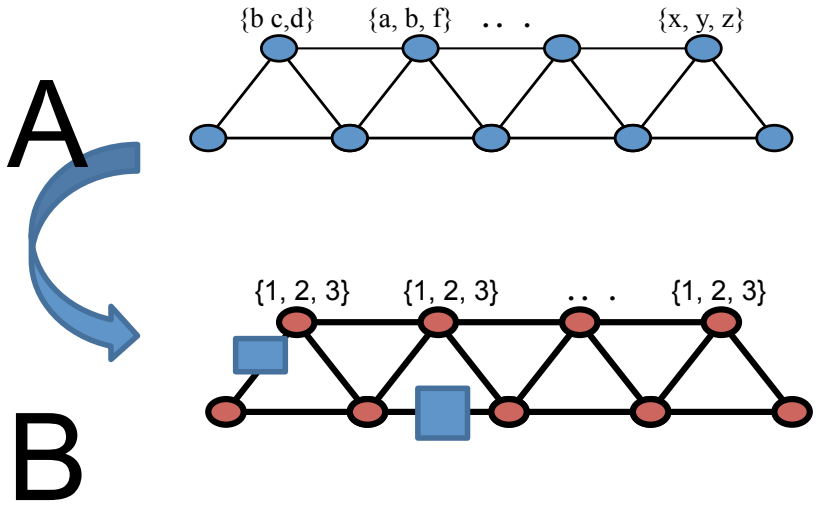

{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}


{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

Def 5.1 The *domain graph* of a MAD-CSP instance (X, U, D, C) is a graph with vertex set x where two variables x, y X are adjacent iff D(x) D(y)



MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

1. No empty constraints
2. No constraint is a proper subset of another
3. No value which is not used by any variable
4. No variable with an empty domain

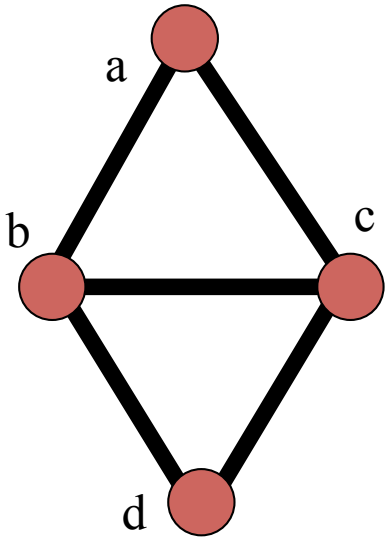
- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}$, $\tau(x) \in \mathcal{D}(x)$

(ii) $\forall C \in \mathcal{C}$ and $\forall x, y \in C$: $x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP models GRAPH COLORING

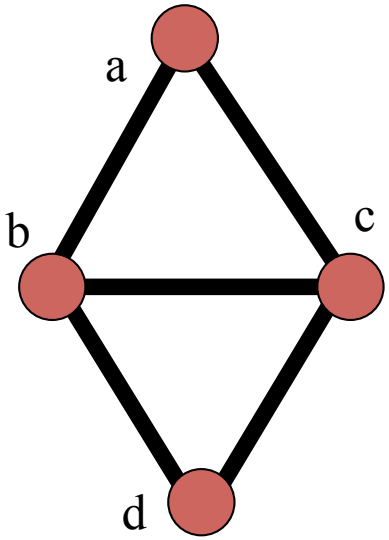


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

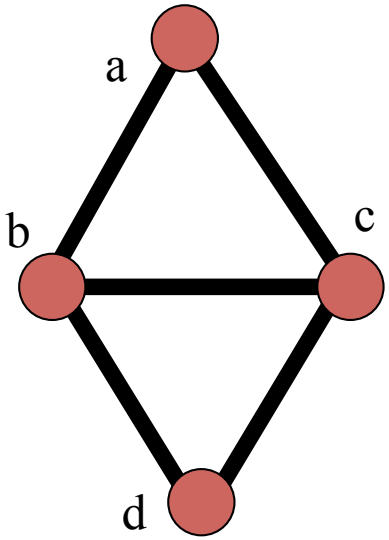


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}

D(b) = {1, 2, 3}

D(c) = {1, 2, 3}

D(d) = {1, 2, 3}

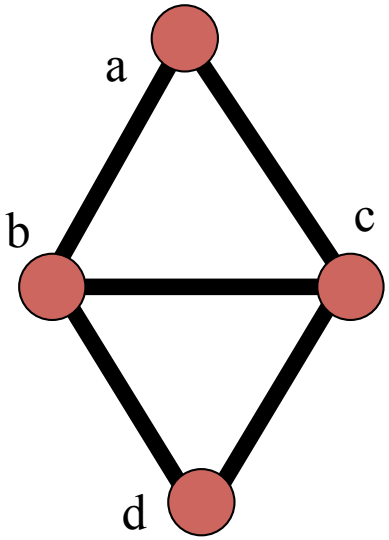
C_1 = {a, b}

C_2 = {a, c}

…

C_5 = {c, d}

MAD-CSP models GRAPH COLORING



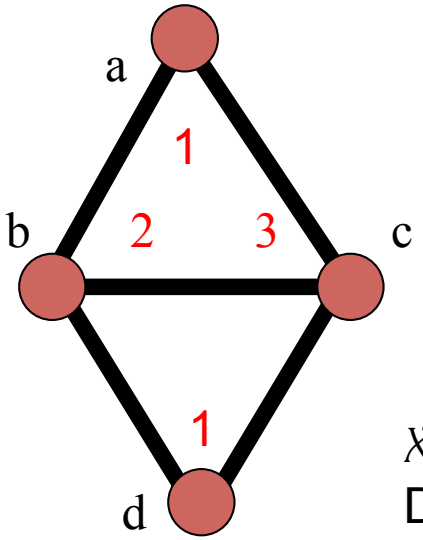
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

MAD-CSP models GRAPH COLORING



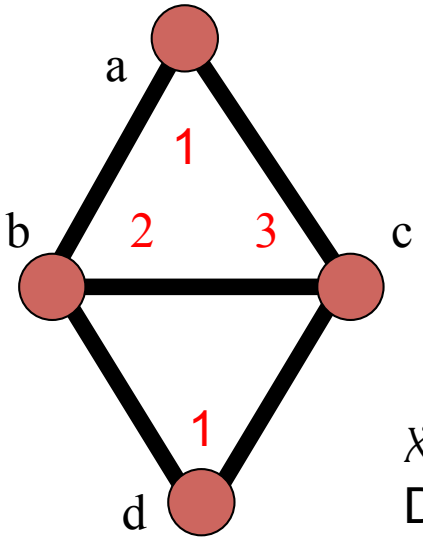
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

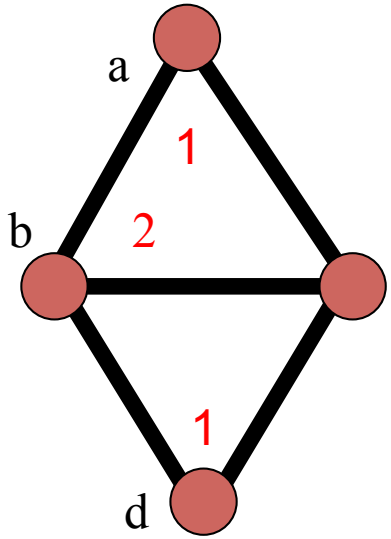
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible value
- l = maximum domain size
- k = maximum constraint size

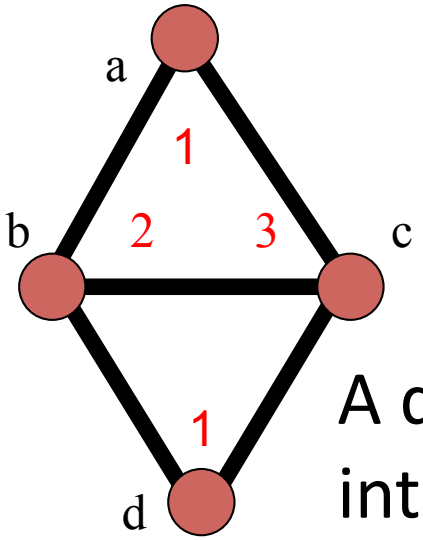


Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible value
- l = maximum domain size
- k = maximum constraint size

GRAPH COLORING is NP-c for instances with number of vertices ≥ 3.

MAD-CSP is NP-c even when restricted to convex domain instances with $m = 3$, $k = 2$, and $l = 3$.



CONVEXITY

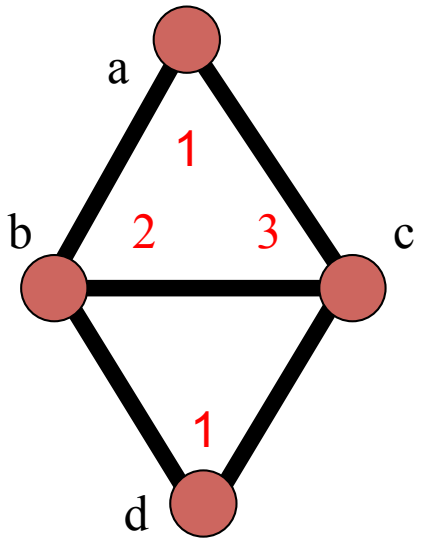
A domain D(x) of a variable is *convex* if it is an interval.

A constraint $C \in \mathcal{C}$ is *convex* if C is an interval.

A MAD-CSP instance has convex domains (constraints) if all domains (constraints) are convex.

A MAD-CSP is *bi-convex* if both domains and constraints are convex simultaneously.

Constraint Graph



Constraint Graph is the underlying hypergraph $(\mathcal{X}, \mathcal{C})$. Two variables are connected by an edge iff there is a constraint containing both of them.

MAD-CSP instance:

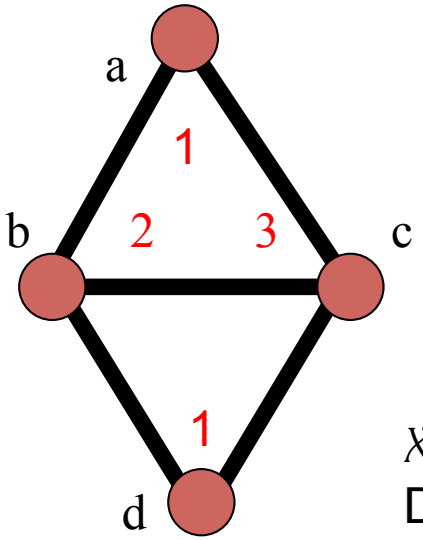
1. Variables
2. Domains

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

3. Constraints

C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

MAD-CSP models GRAPH COLORING

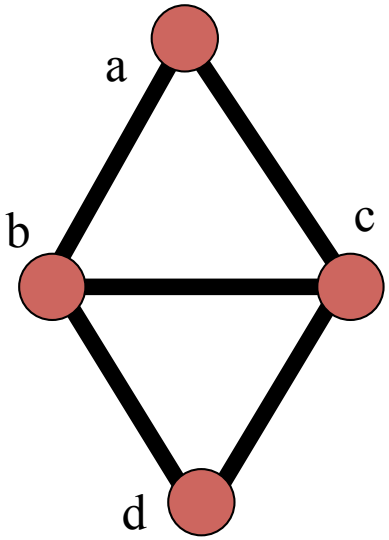


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING

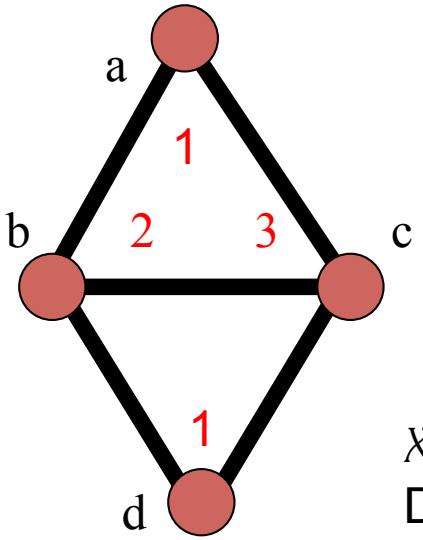


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

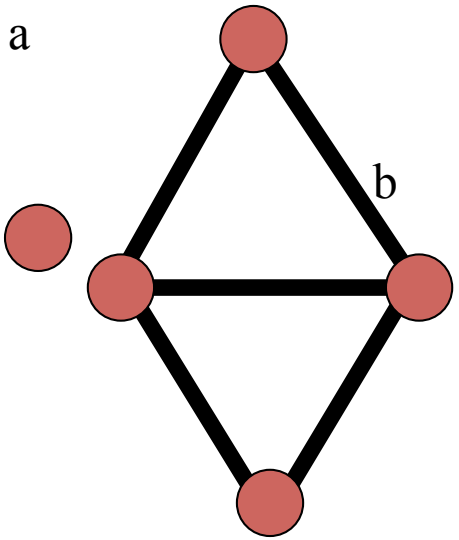


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: **G (V, E), k**

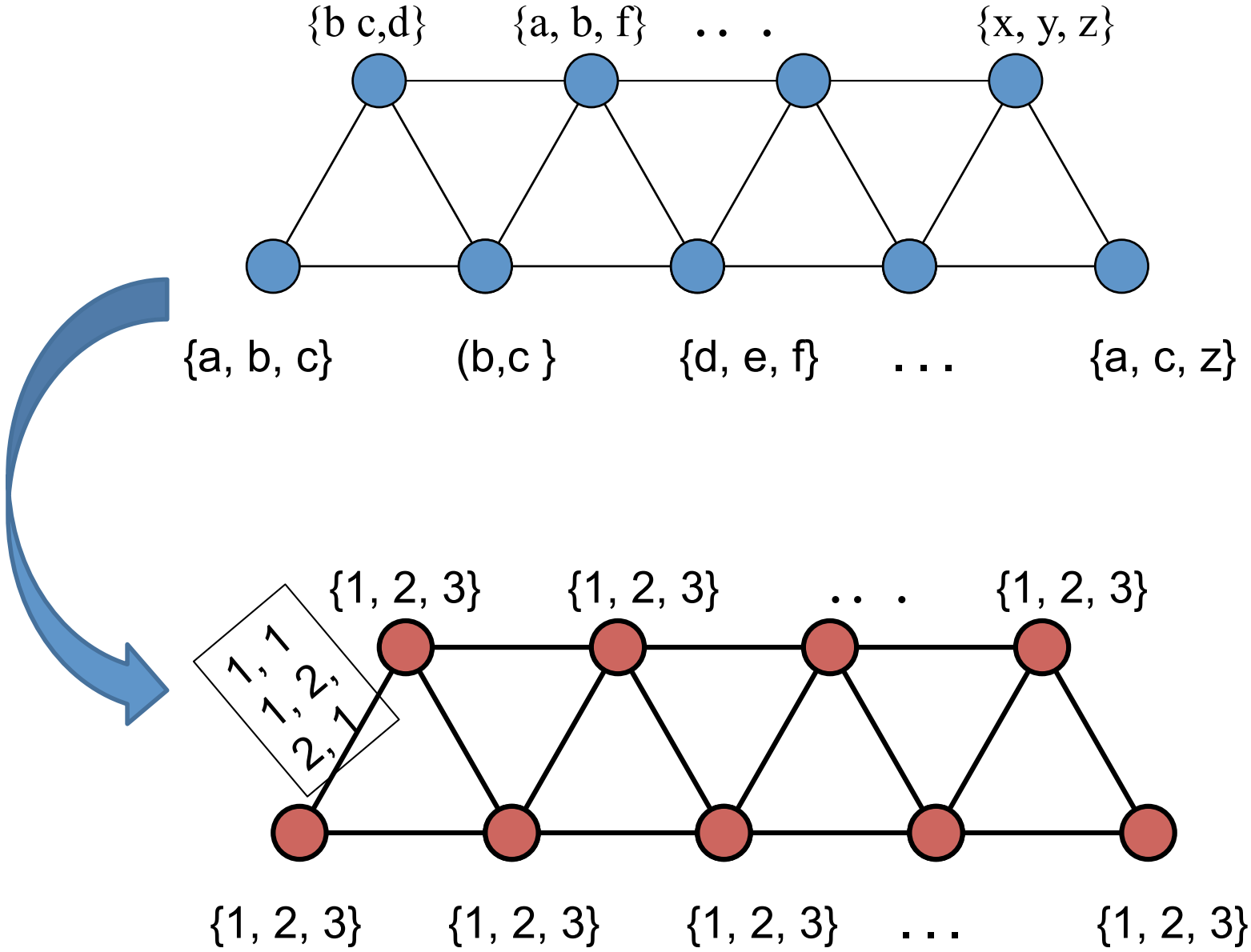
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E,

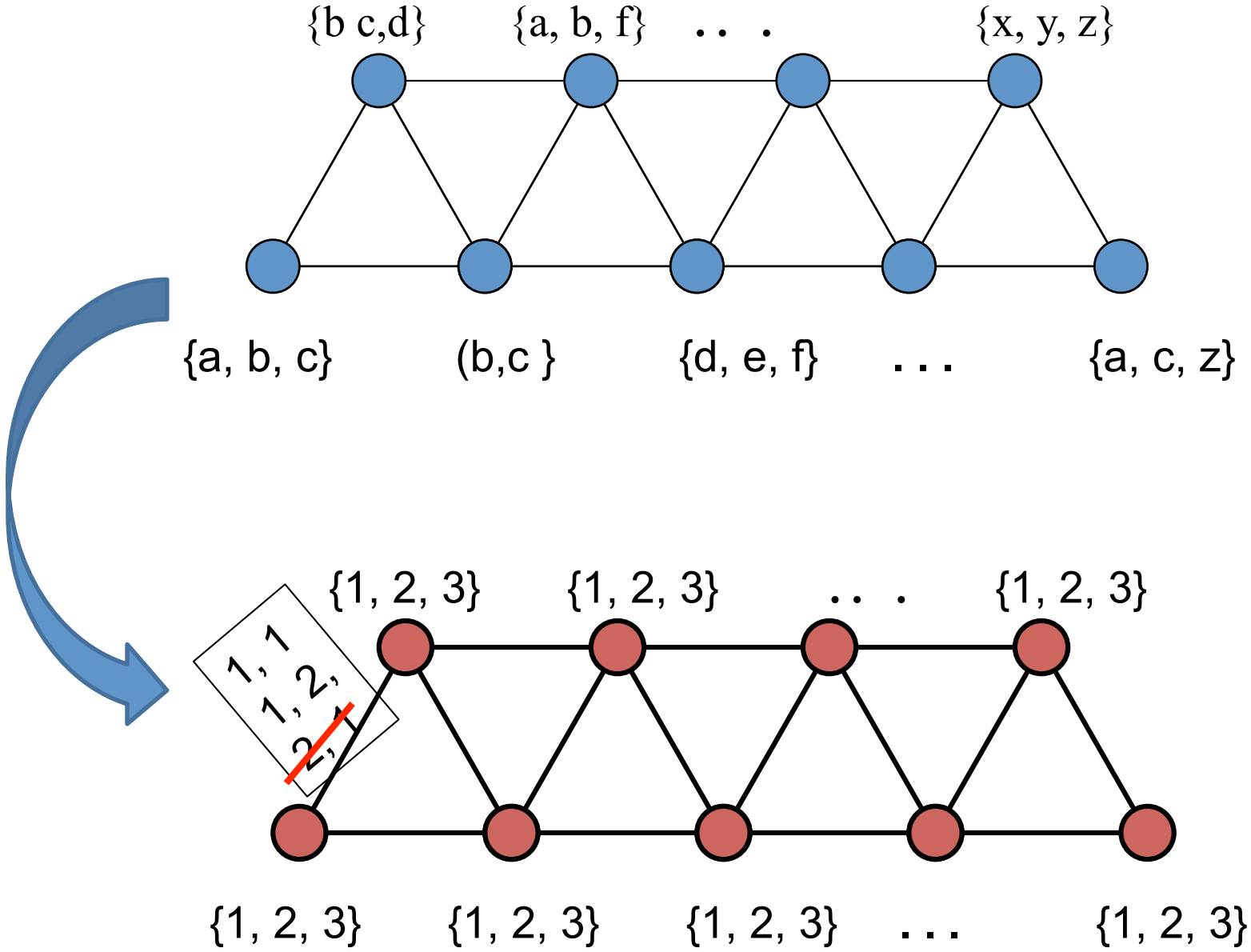
A f: V cover (of size ≤ k, e.g. a subset *S* of the vertices such that each edge has one of its vertices in *S*.

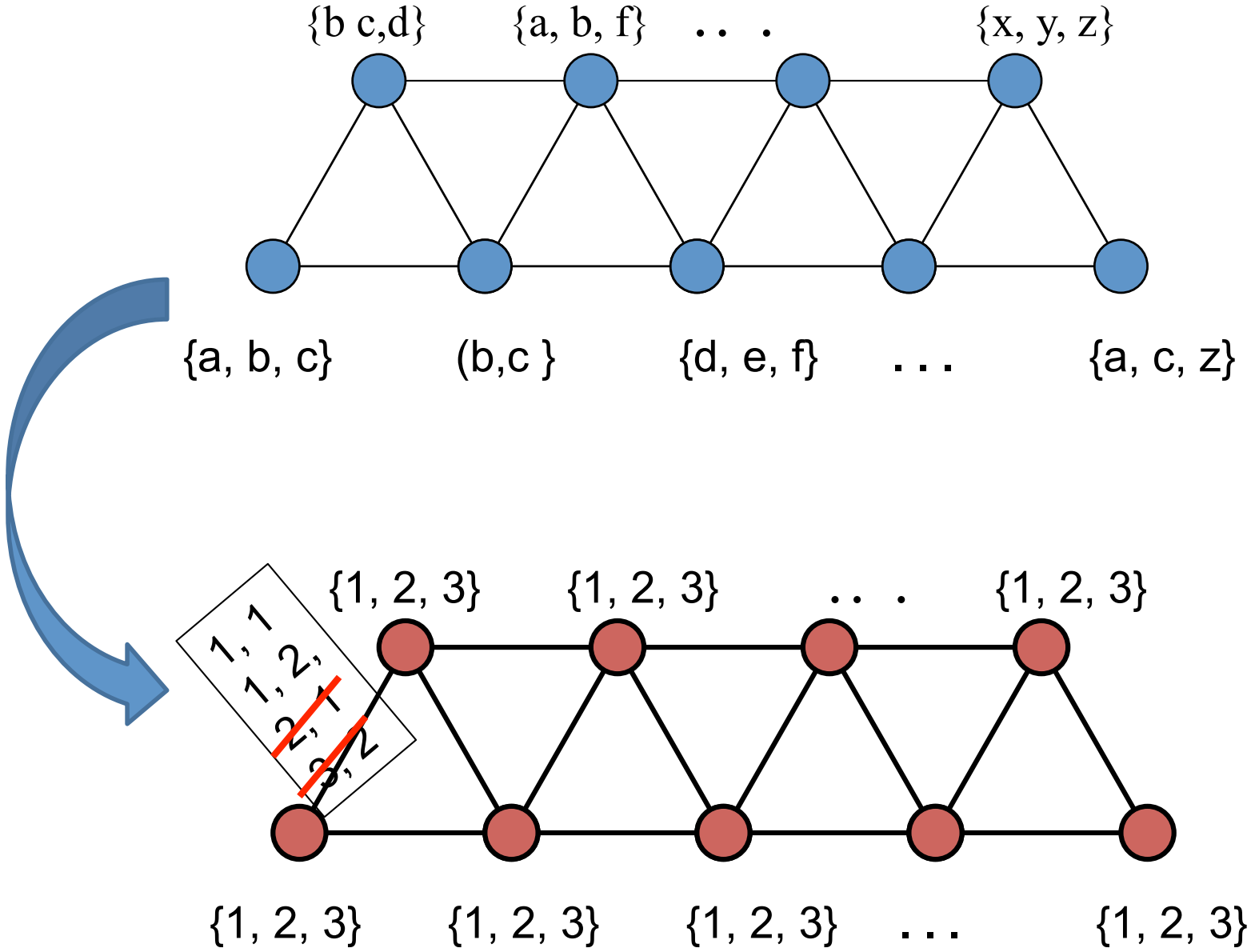
a

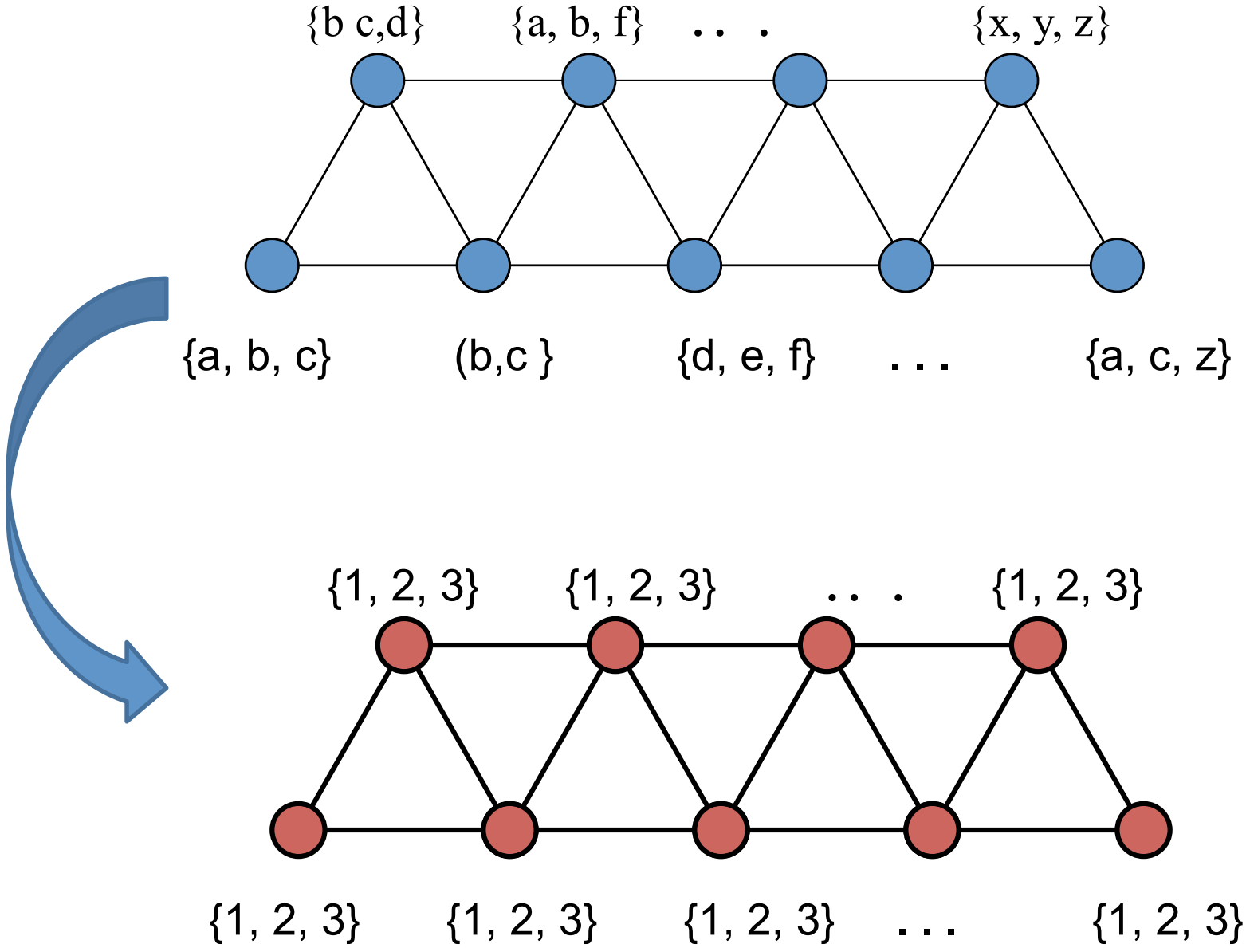
a

MAD-CSP definition


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2,
2, 1
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2,
2, 1
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2
2, 1
2, 2
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}$, $\tau(x) \in \mathcal{D}(x)$

(ii) $\forall (C \in \mathcal{C}$ and $\forall x,y \in C$: $x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

1. No empty constraints
2. No constraint is a proper subset of another
3. No value which is not used by any variable
4. No variable with an empty domain

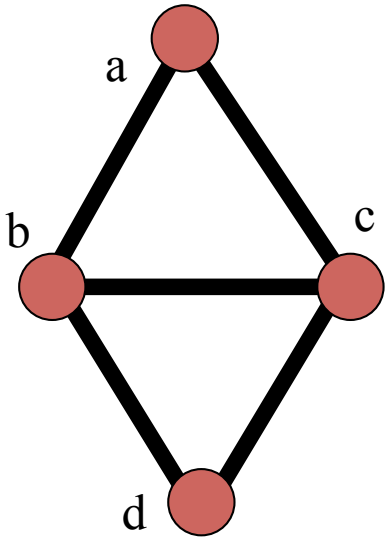
- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}$, $\tau(x) \in \mathcal{D}(x)$

(ii) $\forall C \in \mathcal{C}$ and $\forall x, y \in C$: $x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP models GRAPH COLORING

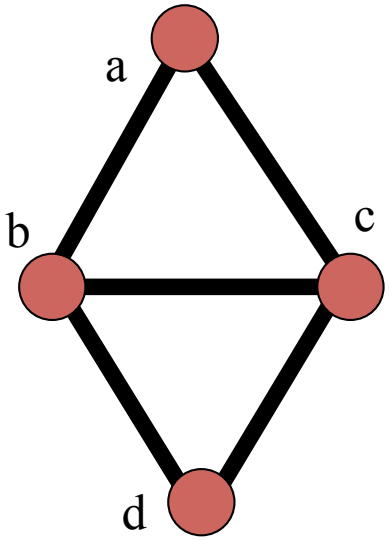


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

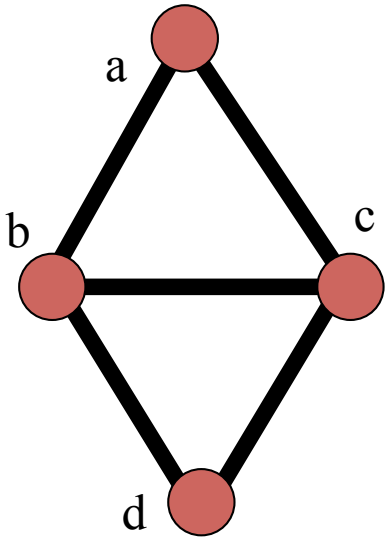


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}

D(b) = {1, 2, 3}

D(c) = {1, 2, 3}

D(d) = {1, 2, 3}

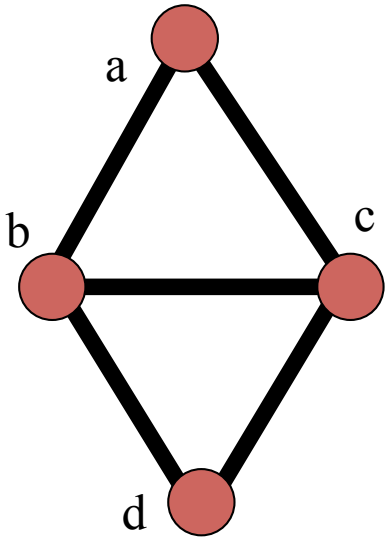
C_1 = {a, b}

C_2 = {a, c}

…

C_5 = {c, d}

MAD-CSP models GRAPH COLORING



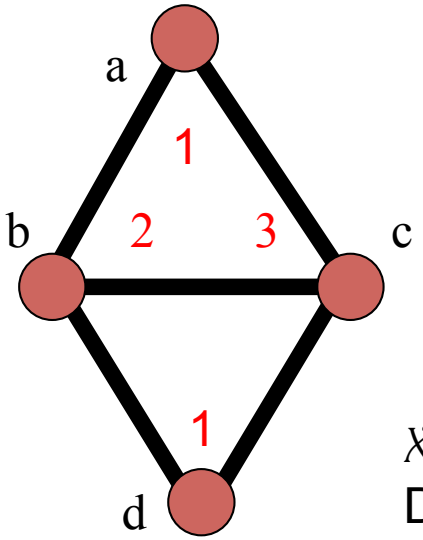
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

MAD-CSP models GRAPH COLORING



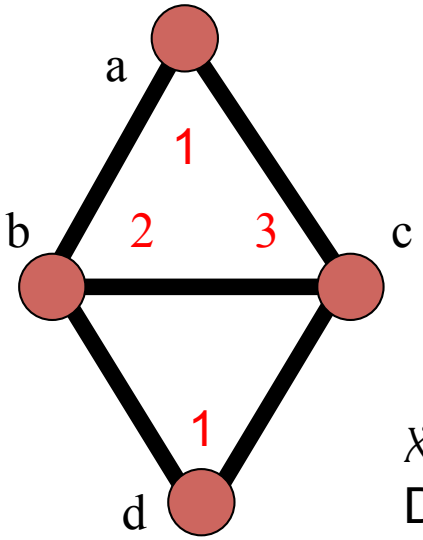
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

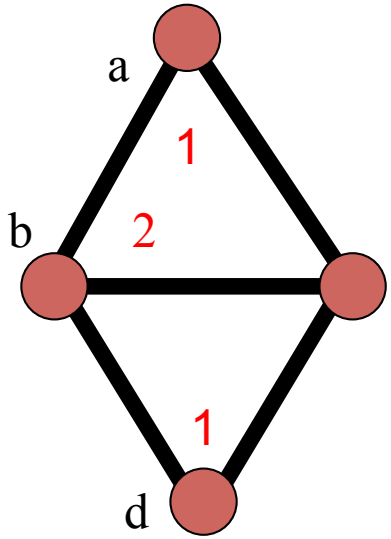
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible value
- l = maximum domain size
- k = maximum constraint size

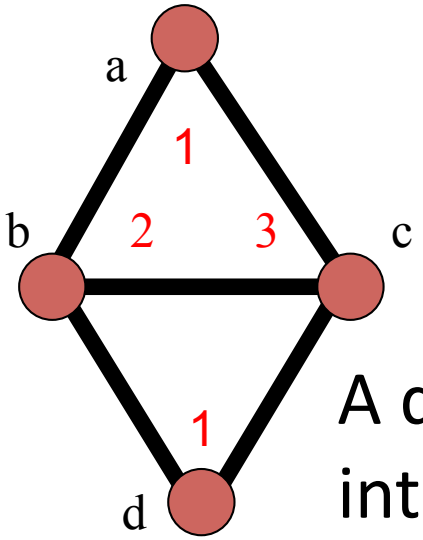


Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible value
- l = maximum domain size
- k = maximum constraint size

GRAPH COLORING is NP-c for instances with number of vertices ≥ 3.

MAD-CSP is NP-c even when restricted to convex domain instances with $m = 3$, $k = 2$, and $l = 3$.



CONVEXITY

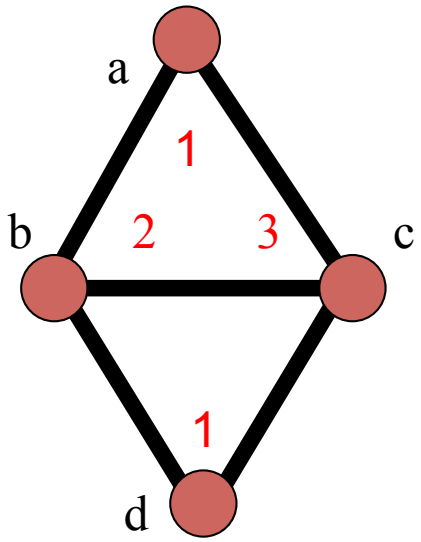
A domain D(x) of a variable is *convex* if it is an interval.

A constraint $C \in \mathcal{C}$ is *convex* if C is an interval.

A MAD-CSP instance has convex domains (constraints) if all domains (constraints) are convex.

A MAD-CSP is *bi-convex* if both domains and constraints are convex simultaneously.

Constraint Graph



Constraint Graph is the underlying hypergraph $(\mathcal{X}, \mathcal{C})$. Two variables are connected by an edge iff there is a constraint containing both of them.

MAD-CSP instance:

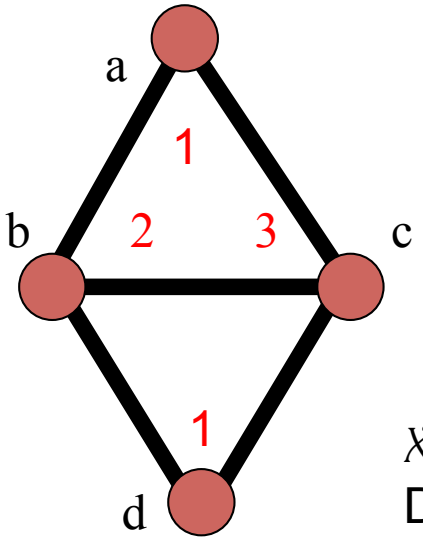
1. Variables
2. Domains

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

3. Constraints

C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

MAD-CSP models GRAPH COLORING

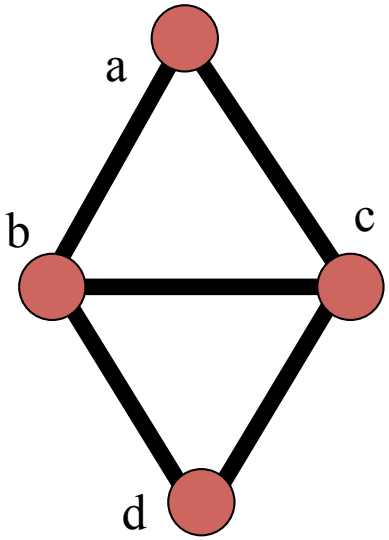


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING

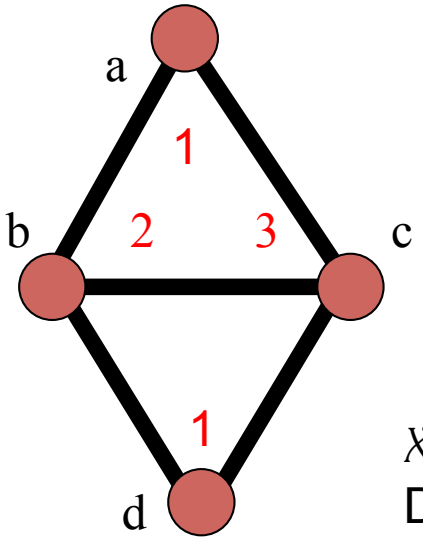


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

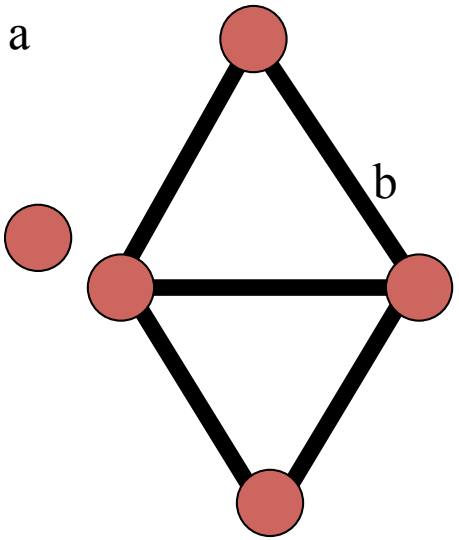


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: **G (V, E), k**

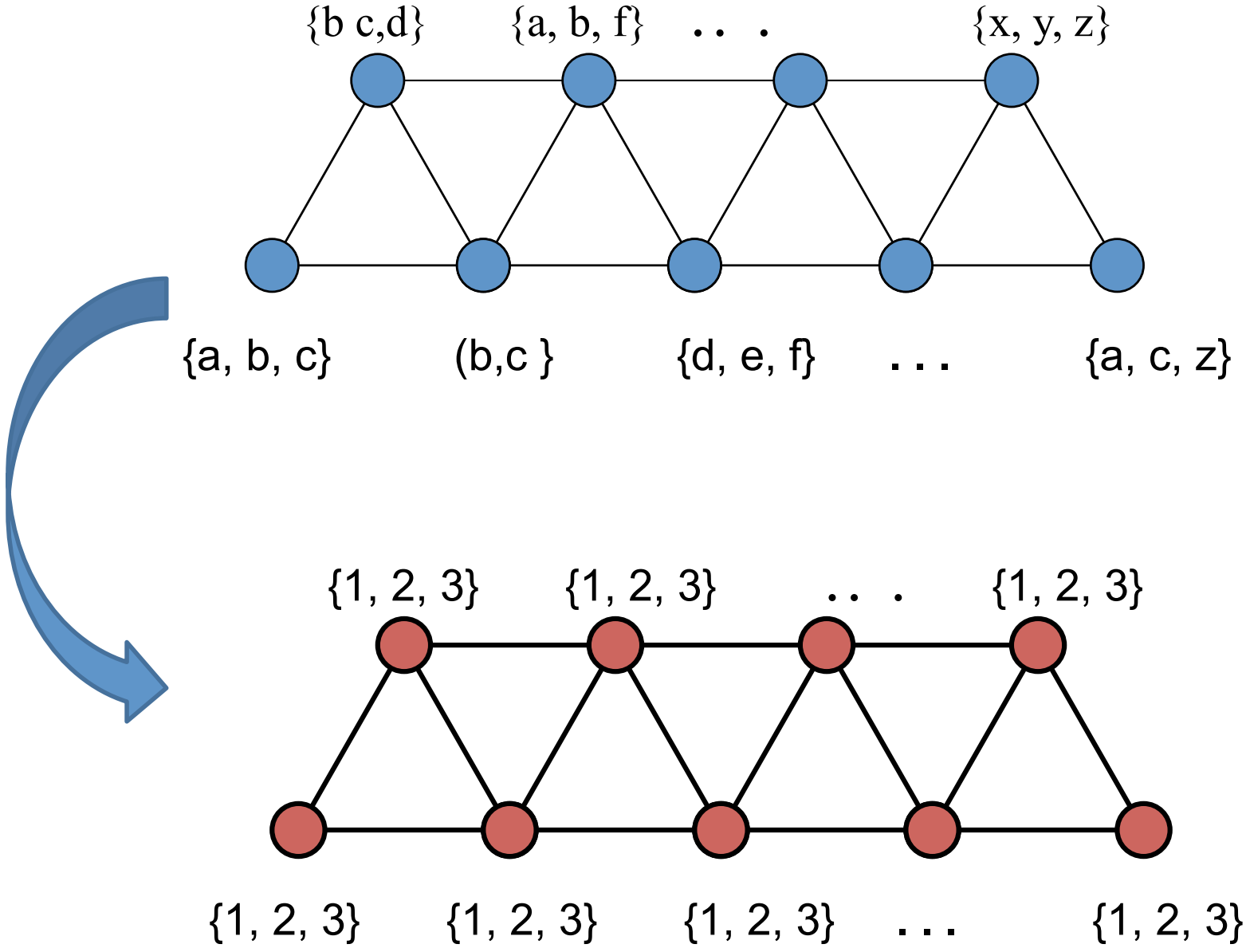
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E,

A f: V cover (of size ≤ k, e.g. a subset *S* of the vertices such that each edge has one of its vertices in *S*.

a

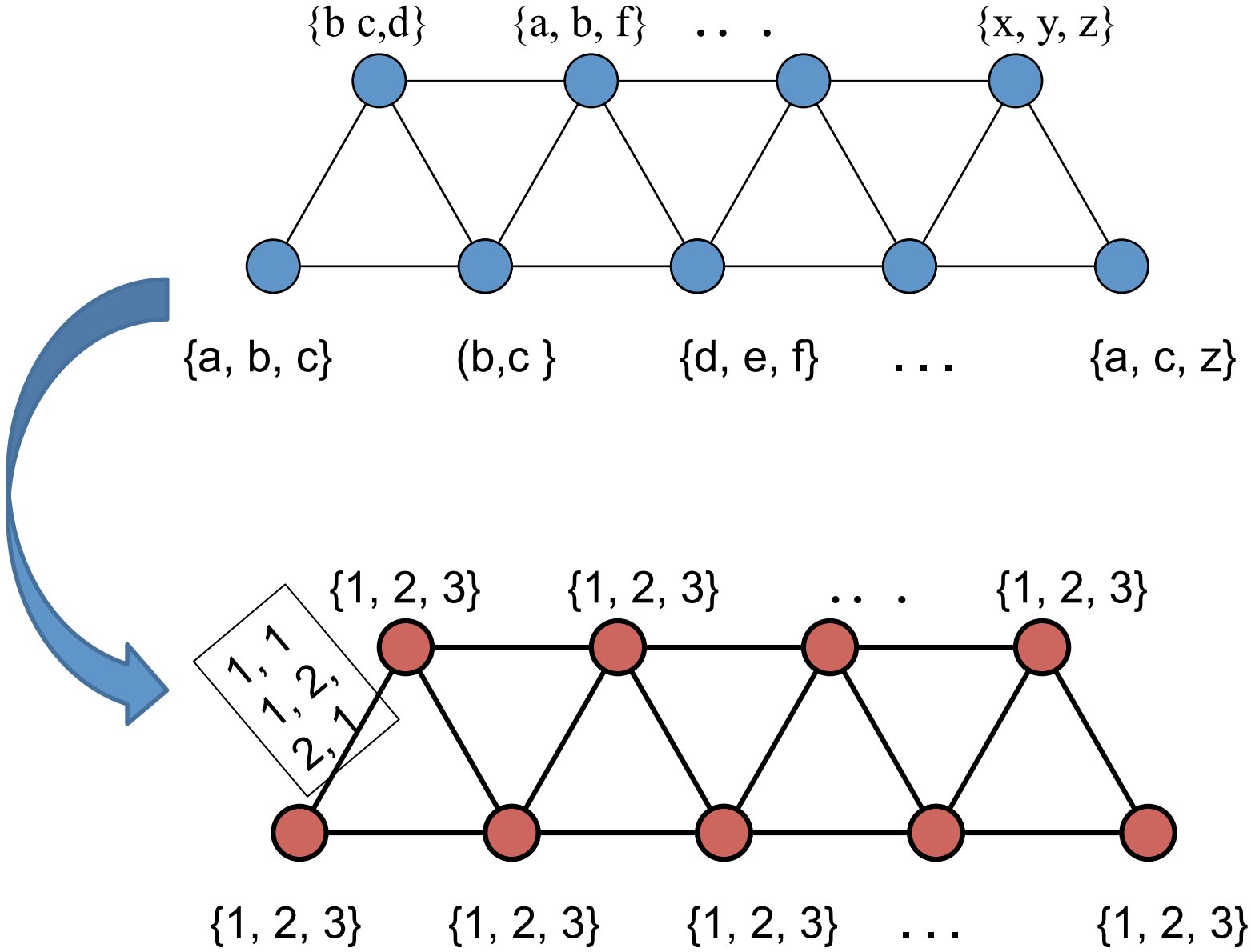
a

MAD-CSP definition


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

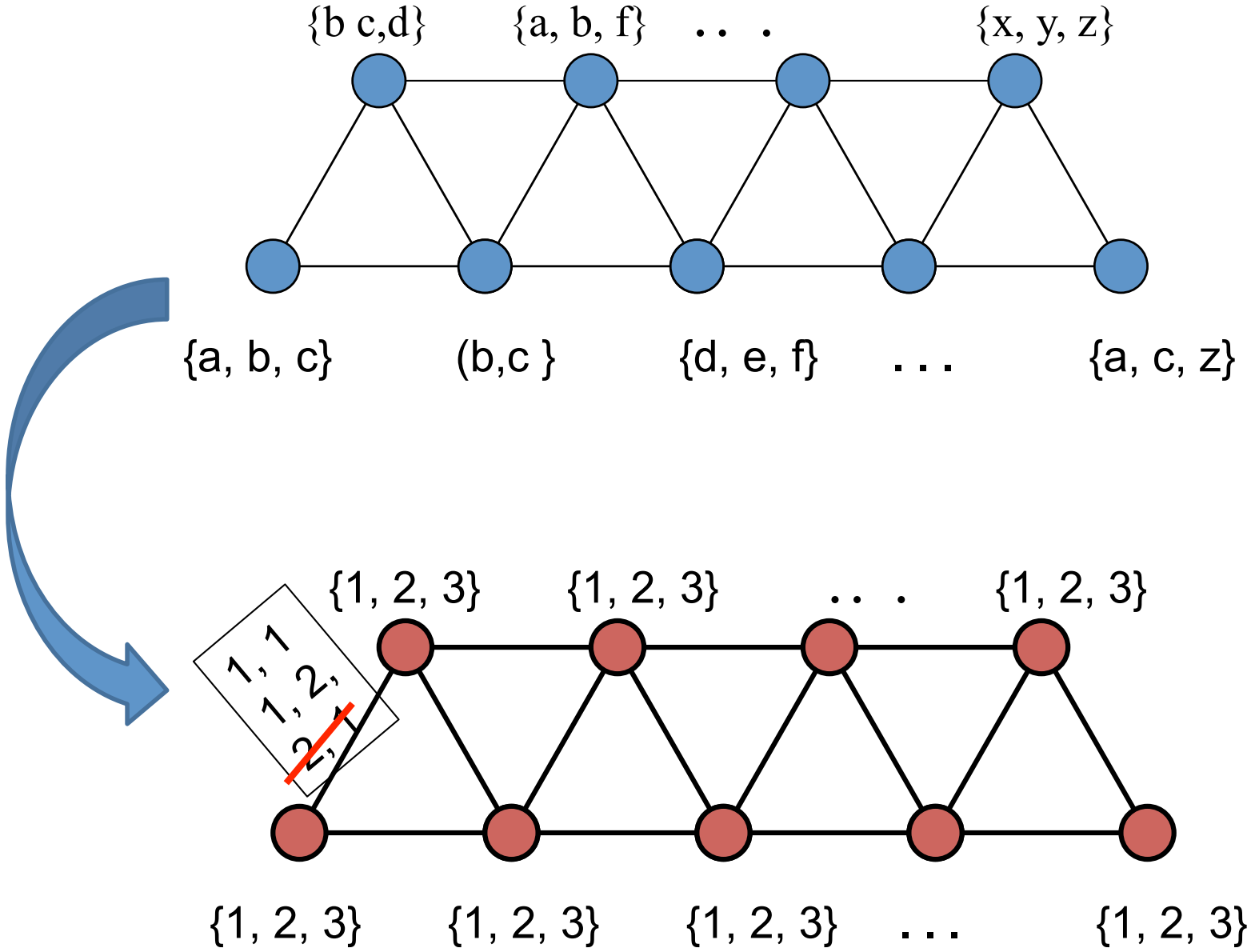
Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2,
2, 1
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

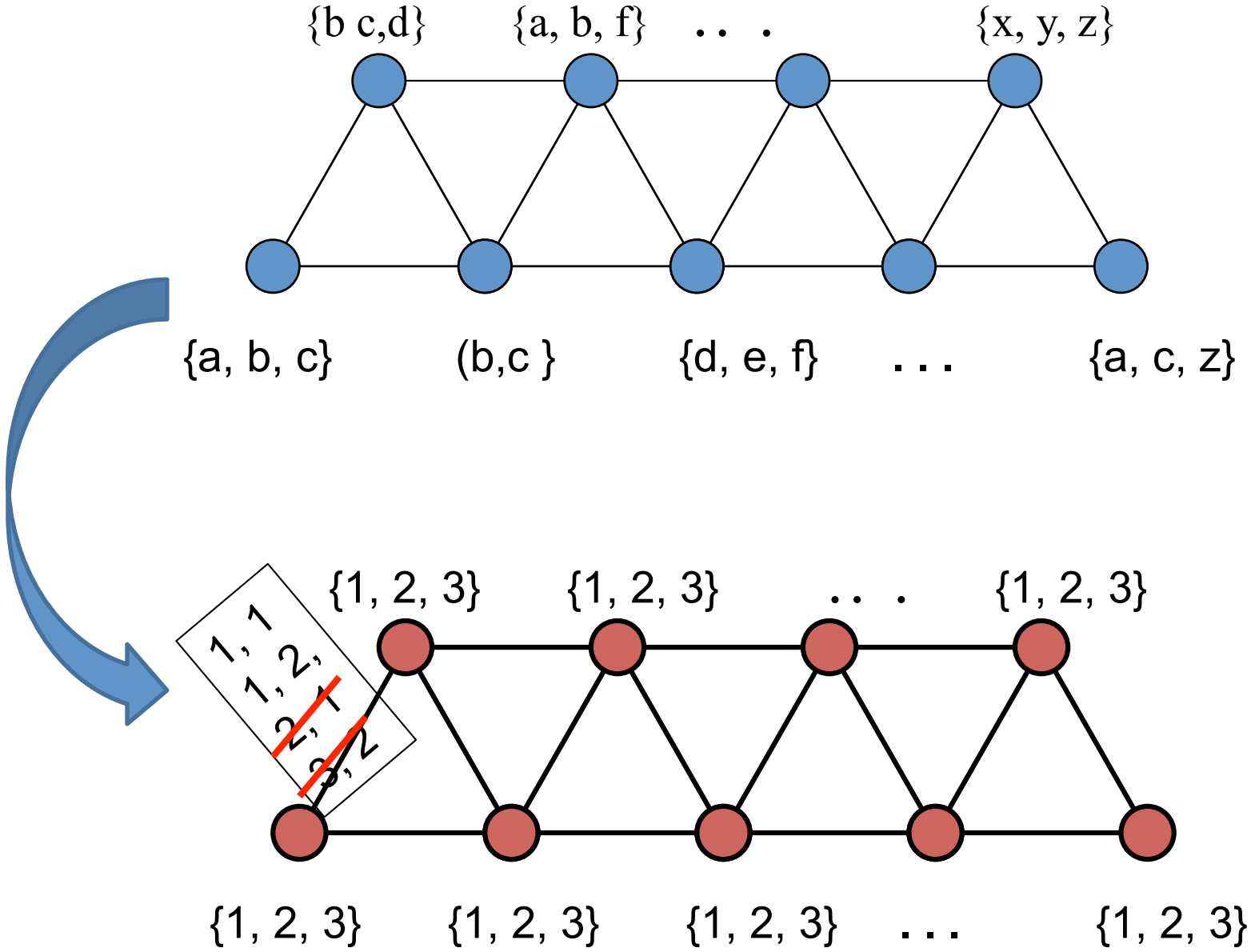
Title

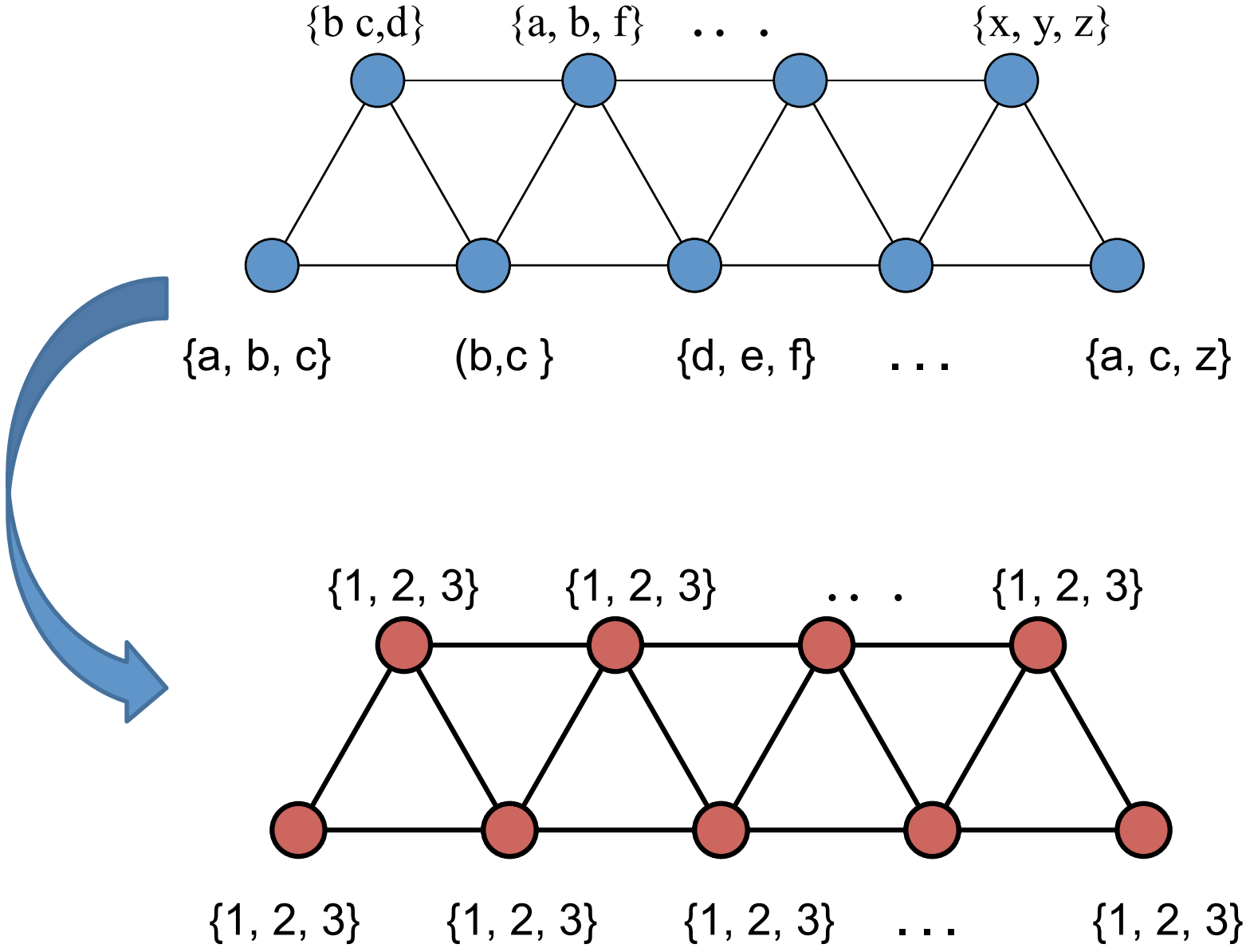
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2,
2, 1
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2
2, 1
2, 2
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}$, $\tau(x) \in \mathcal{D}(x)$

(ii) $\forall (C \in \mathcal{C}$ and $\forall x,y \in C$: $x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

1. No empty constraints
2. No constraint is a proper subset of another
3. No value which is not used by any variable
4. No variable with an empty domain

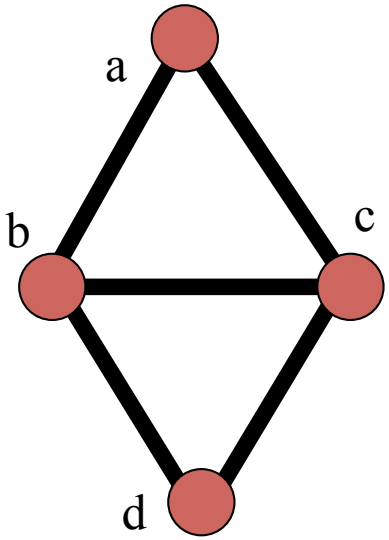
- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}$, $\tau(x) \in \mathcal{D}(x)$

(ii) $\forall C \in \mathcal{C}$ and $\forall x, y \in C$: $x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP models GRAPH COLORING

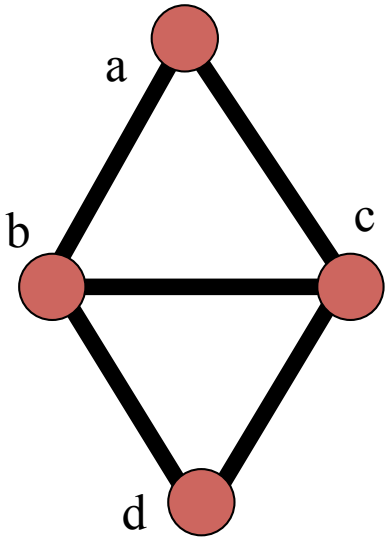


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

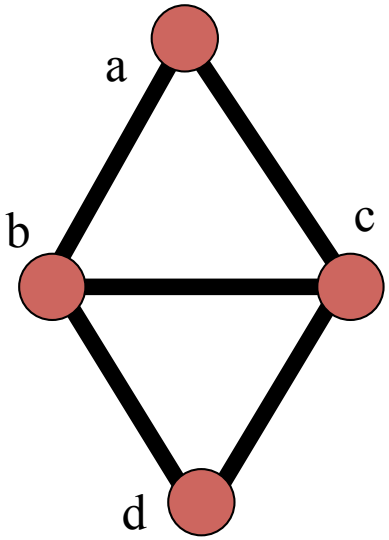


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}

D(b) = {1, 2, 3}

D(c) = {1, 2, 3}

D(d) = {1, 2, 3}

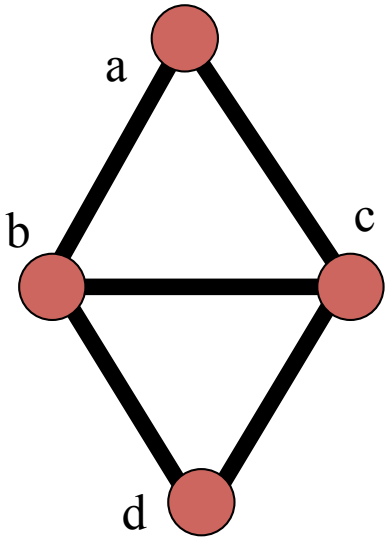
C_1 = {a, b}

C_2 = {a, c}

…

C_5 = {c, d}

MAD-CSP models GRAPH COLORING



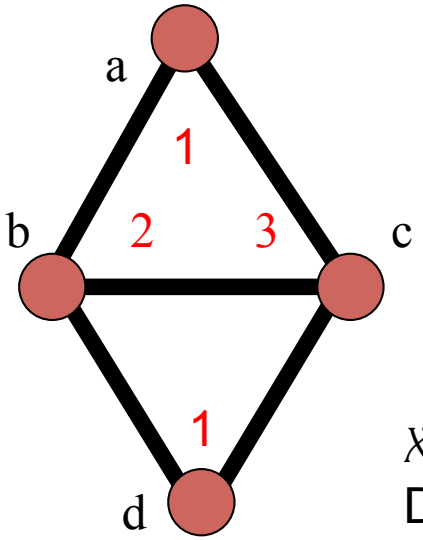
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

MAD-CSP models GRAPH COLORING



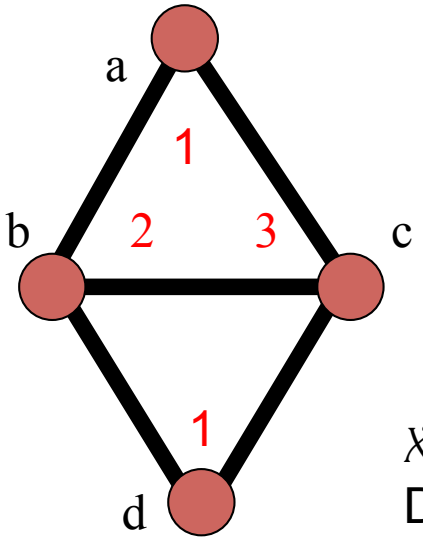
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

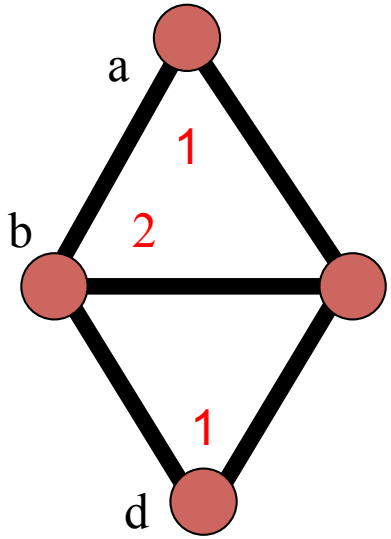
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f\big|_c$ is injective.

Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible value
- l = maximum domain size
- k = maximum constraint size

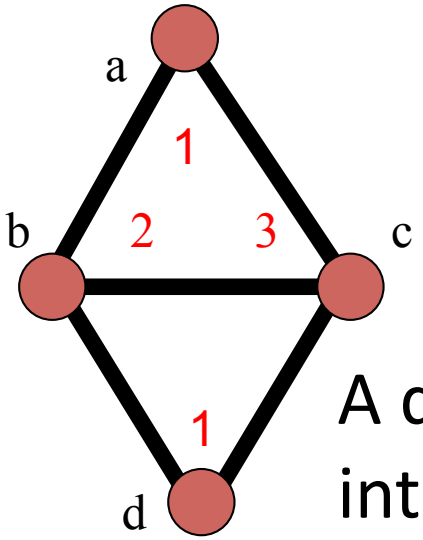


Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible value
- l = maximum domain size
- k = maximum constraint size

GRAPH COLORING is NP-c for instances with number of vertices ≥ 3.

MAD-CSP is NP-c even when restricted to convex domain instances with $m = 3$, $k = 2$, and $l = 3$.



CONVEXITY

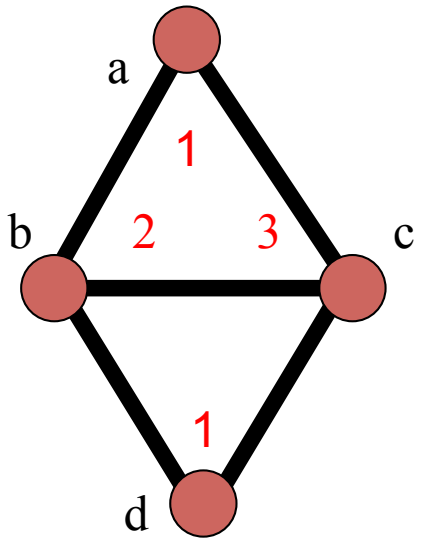
A domain D(x) of a variable is *convex* if it is an interval.

A constraint $C \in \mathcal{C}$ is *convex* if C is an interval.

A MAD-CSP instance has convex domains (constraints) if all domains (constraints) are convex.

A MAD-CSP is *bi-convex* if both domains and constraints are convex simultaneously.

Constraint Graph



Constraint Graph is the underlying hypergraph $(\mathcal{X}, \mathcal{C})$. Two variables are connected by an edge iff there is a constraint containing both of them.

MAD-CSP instance:

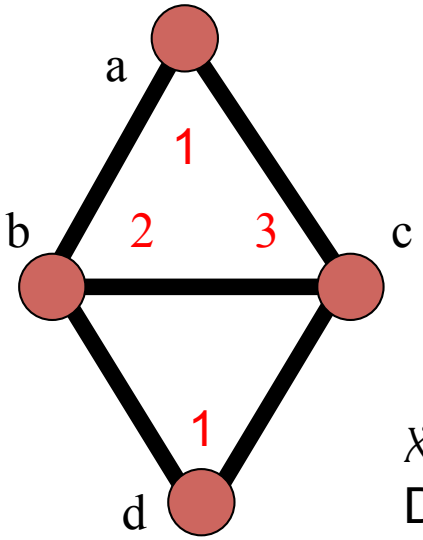
1. Variables
2. Domains

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

3. Constraints

C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

MAD-CSP models GRAPH COLORING

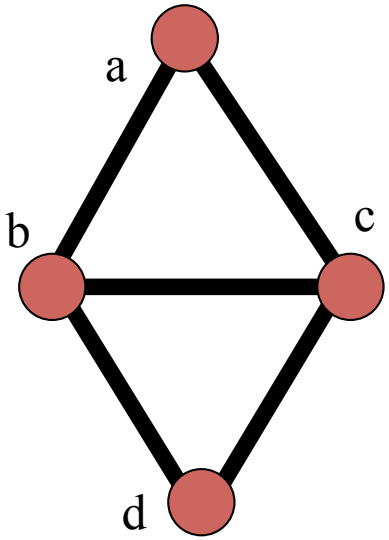


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING

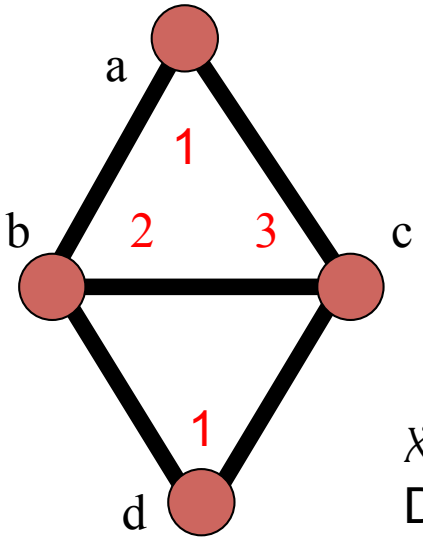


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

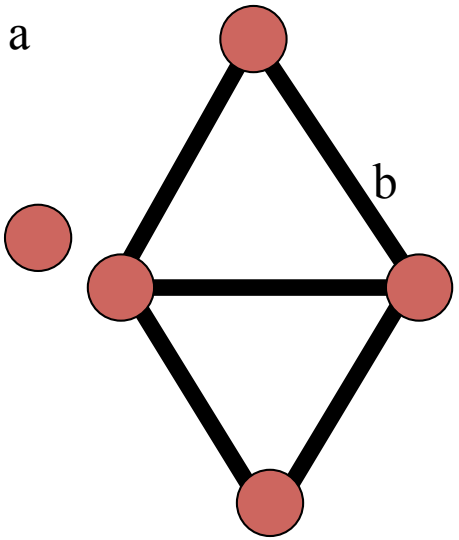


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: **G (V, E), k**

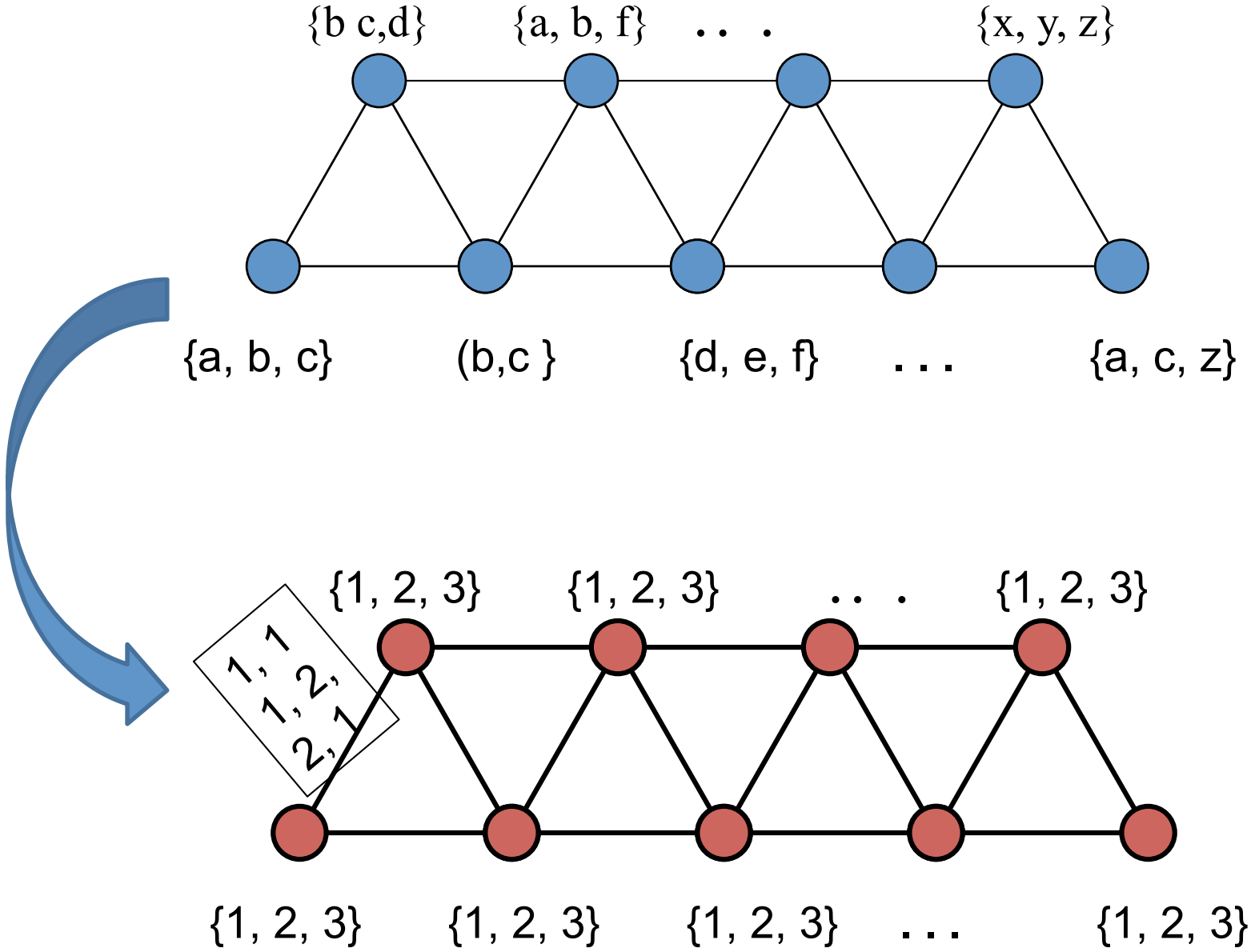
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E,

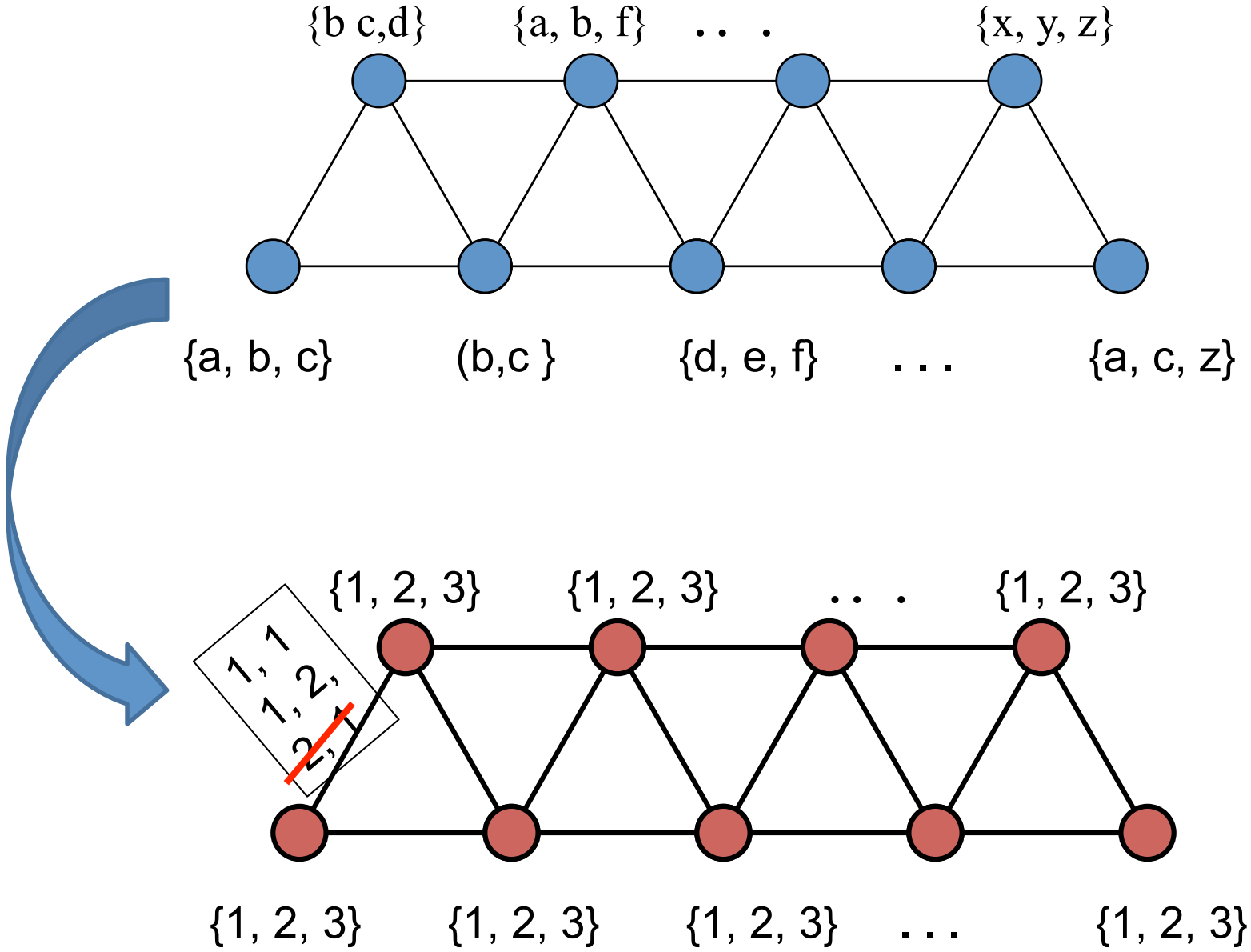
A f: V cover (of size ≤ k, e.g. a subset *S* of the vertices such that each edge has one of its vertices in *S*.

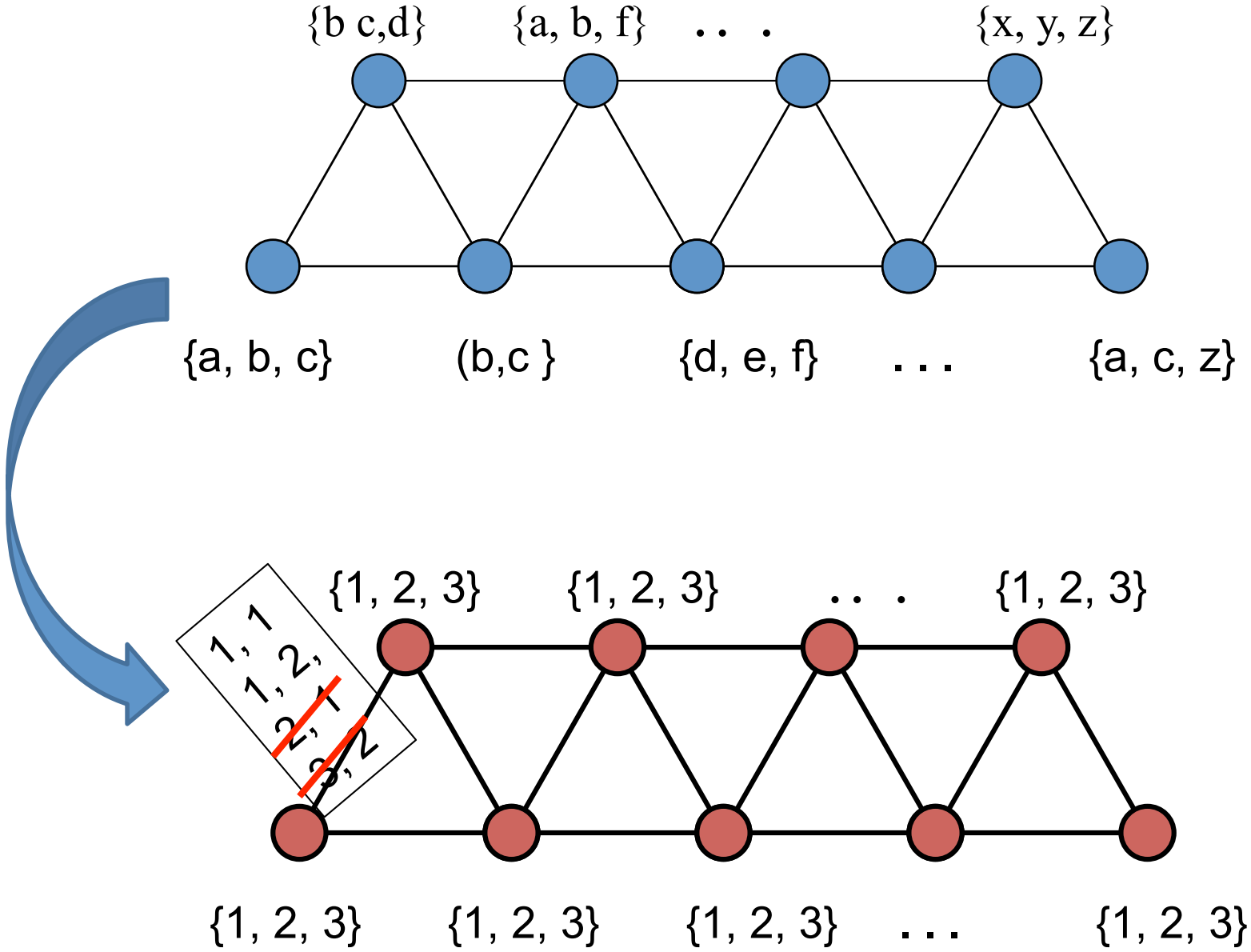
a

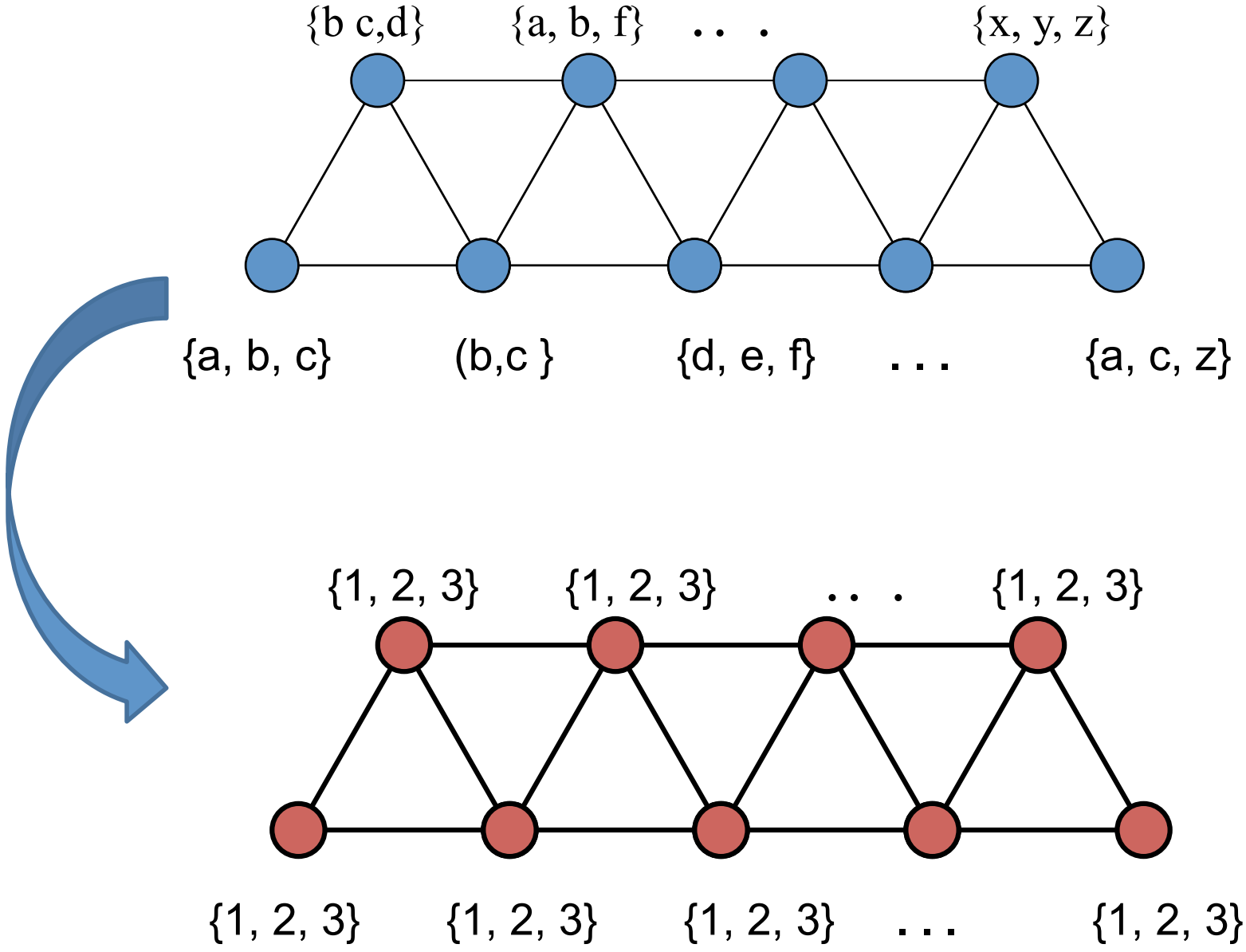
a

MAD-CSP definition


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2,
2, 1
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2,
2, 1
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
1, 1
1, 2
2, 1
2, 2
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}


{b c,d}
{a, b, f}
. . .
{x, y, z}
{a, b, c}
(b,c }
{d, e, f}
. . .
{a, c, z}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
{1, 2, 3}
. . .
{1, 2, 3}

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function τ: $\mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}, \tau(x) \in \mathcal{D}(x)$

(ii) $\forall (C \in \mathcal{C}$ and $\forall x,y \in C: x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP is a quadruple $(\mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{C})$

1. No empty constraints
2. No constraint is a proper subset of another
3. No value which is not used by any variable
4. No variable with an empty domain

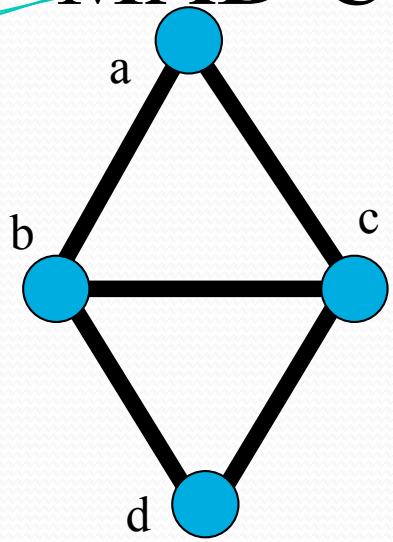
- $\mathcal{X}$, set of variables
- $\mathcal{U}$, universe of possible values
- $\mathcal{D}$, assigns domains to variables: function from $\mathcal{X}$ to $2^{\mathcal{U}}$
- $\mathcal{C}$, constraints: subset of $2^{\mathcal{X}}$, enforcing that all variables occurring together in one constraint must be assigned different values.

A solution to a MAD-CSP instance is a function $\tau: \mathcal{X} \rightarrow \mathcal{U}$,

(i) $\forall x \in \mathcal{X}, \tau(x) \in \mathcal{D}(x)$

(ii) $\forall C \in \mathcal{C}$ and $\forall x, y \in C: x \neq y \Rightarrow \tau(x) \neq \tau(y)$

MAD-CSP models GRAPH COLORING

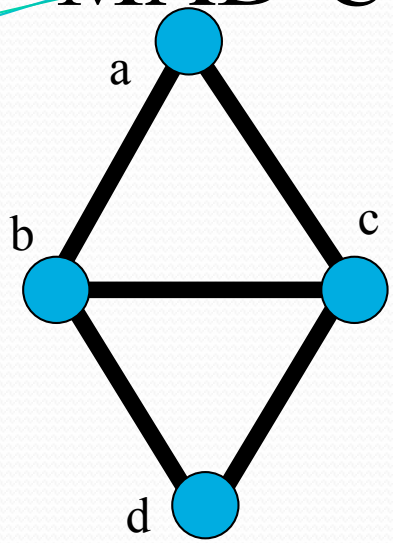


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING

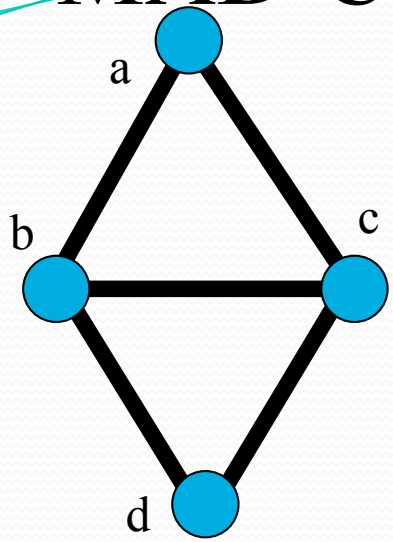


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

Variables are the vertices
D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



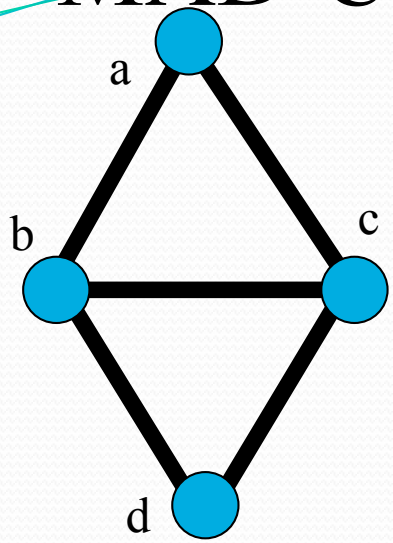
Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

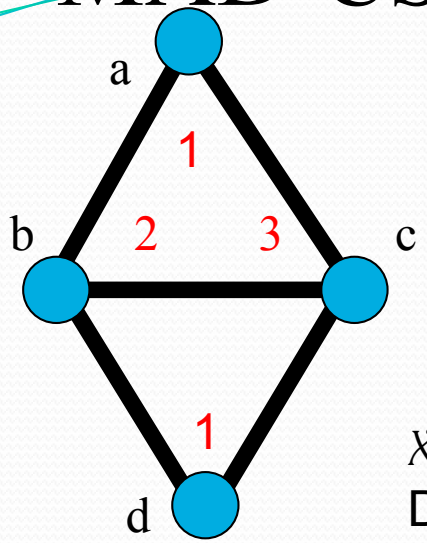
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f \mid_c$ is injective.

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

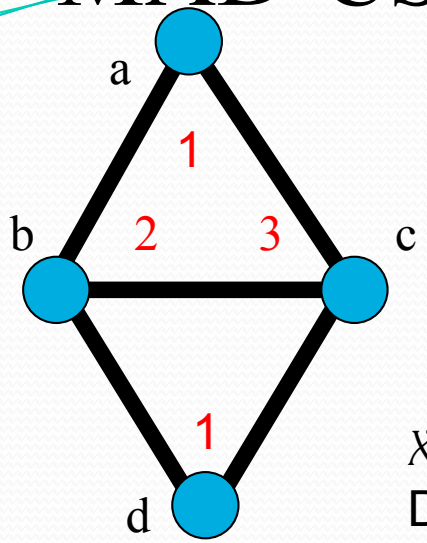
Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f \mid_c$ is injective.

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

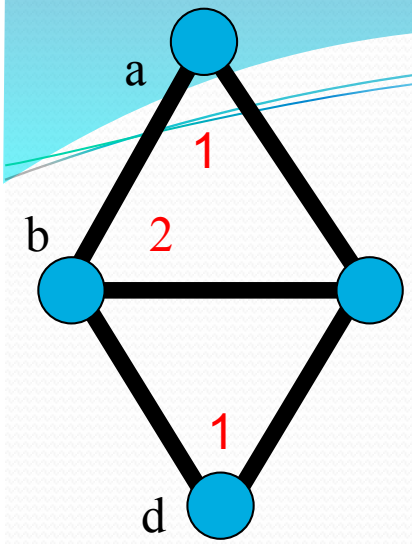
$\mathcal{X}$: Variables are the vertices

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}
C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

An all-different constraint C is satisfied by an assignment f: Vars → Values if $f \mid_c$ is injective.

Parameters for MAD-CSP

- $n = |\mathcal{X}|$, number of variables
- $m = |\mathcal{U}|$, size of the universe of possible values
- l = maximum domain size
- k = maximum constraint size

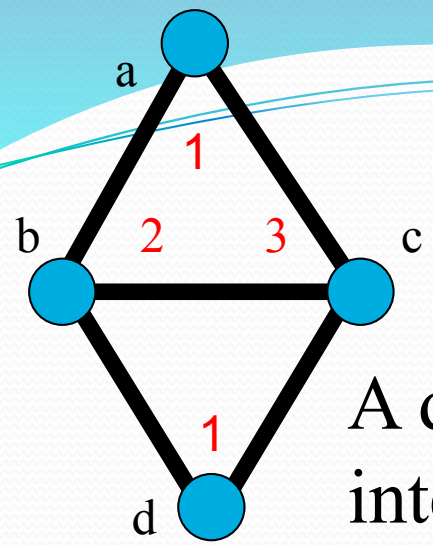


Parameters for MAD-CSP

- $n = |X|$, number of variables
- $m = |U|$, size of the universe of possible values
- l = maximum domain size
- k = maximum constraint size

GRAPH COLORING is NP-c for instances with number of vertices ≥ 3.

MAD-CSP is NP-c even when restricted to convex domain instances with $m = 3$, $k = 2$, and $l = 3$.



CONVEXITY

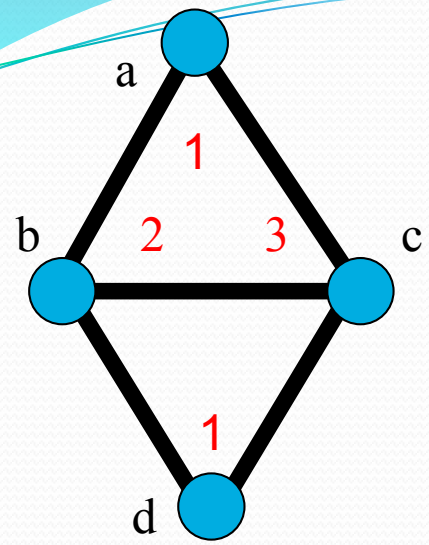
A domain D(x) of a variable is *convex* if it is an interval.

A constraint $C \in \mathcal{C}$ is *convex* if C is an interval.

A MAD-CSP instance has convex domains (constraints) if all domains (constraints) are convex.

A MAD-CSP is *bi-convex* if both domains and constraints are convex simultaneously.

Constraint Graph



Constraint Graph is the underlying hypergraph $(\mathcal{X}, \mathcal{C})$. Two variables are connected by an edge iff there is a constraint containing both of them.

MAD-CSP instance:

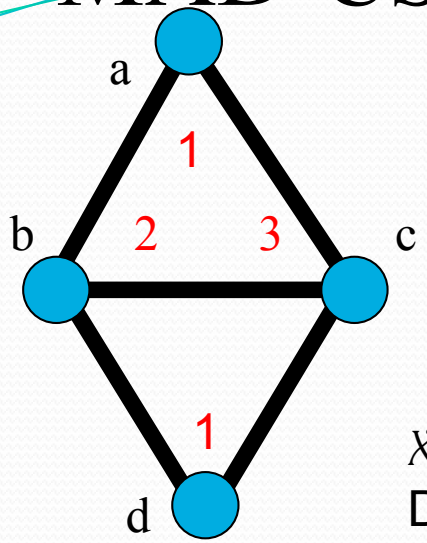
1. Variables
2. Domains

D(a) = {1, 2, 3}
D(b) = {1, 2, 3}
D(c) = {1, 2, 3}
D(d) = {1, 2, 3}

3. Constraints

C_1 = {a, b}
C_2 = {a, c}
…
C_5 = {c, d}

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

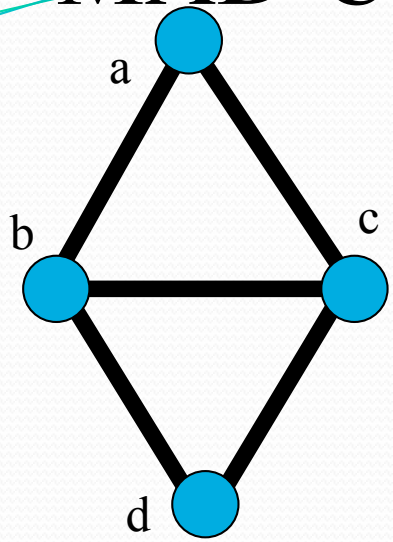
D(a) = {1, 2, 3}

D(b) = {1, 2, 3}

D(c) = {1, 2, 3}

D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING

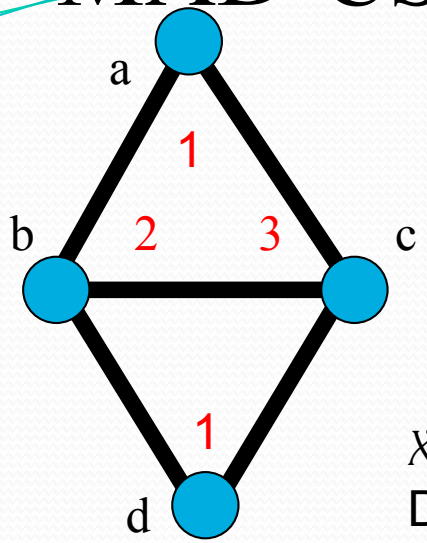


Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E.

X: Variables are the vertices

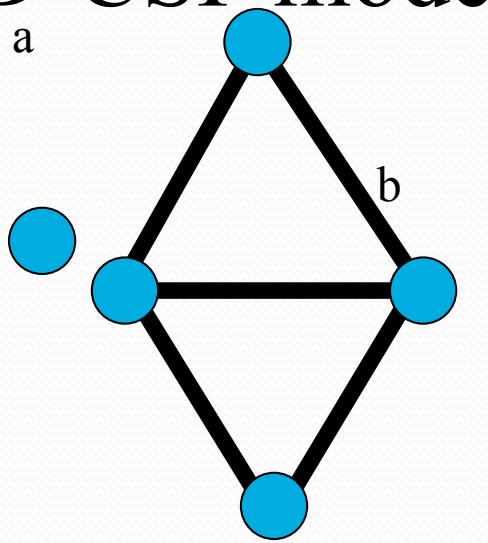
D(a) = {1, 2, 3}

D(b) = {1, 2, 3}

D(c) = {1, 2, 3}

D(d) = {1, 2, 3}

MAD-CSP models GRAPH COLORING



Input: G (V, E), k

Output: f: V → {1, 2, …k} with f (u) ≠ f (v), ∀{u, v} ∈ E,

A f: V cover (of size ≤ k, e.g. a subset *S* of the vertices such that each edge has one of its vertices in *S*.

a

a

MAD-CSP definition

Title

- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.
- Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus et magna. Fusce sed sem sed magna suscipit egestas.

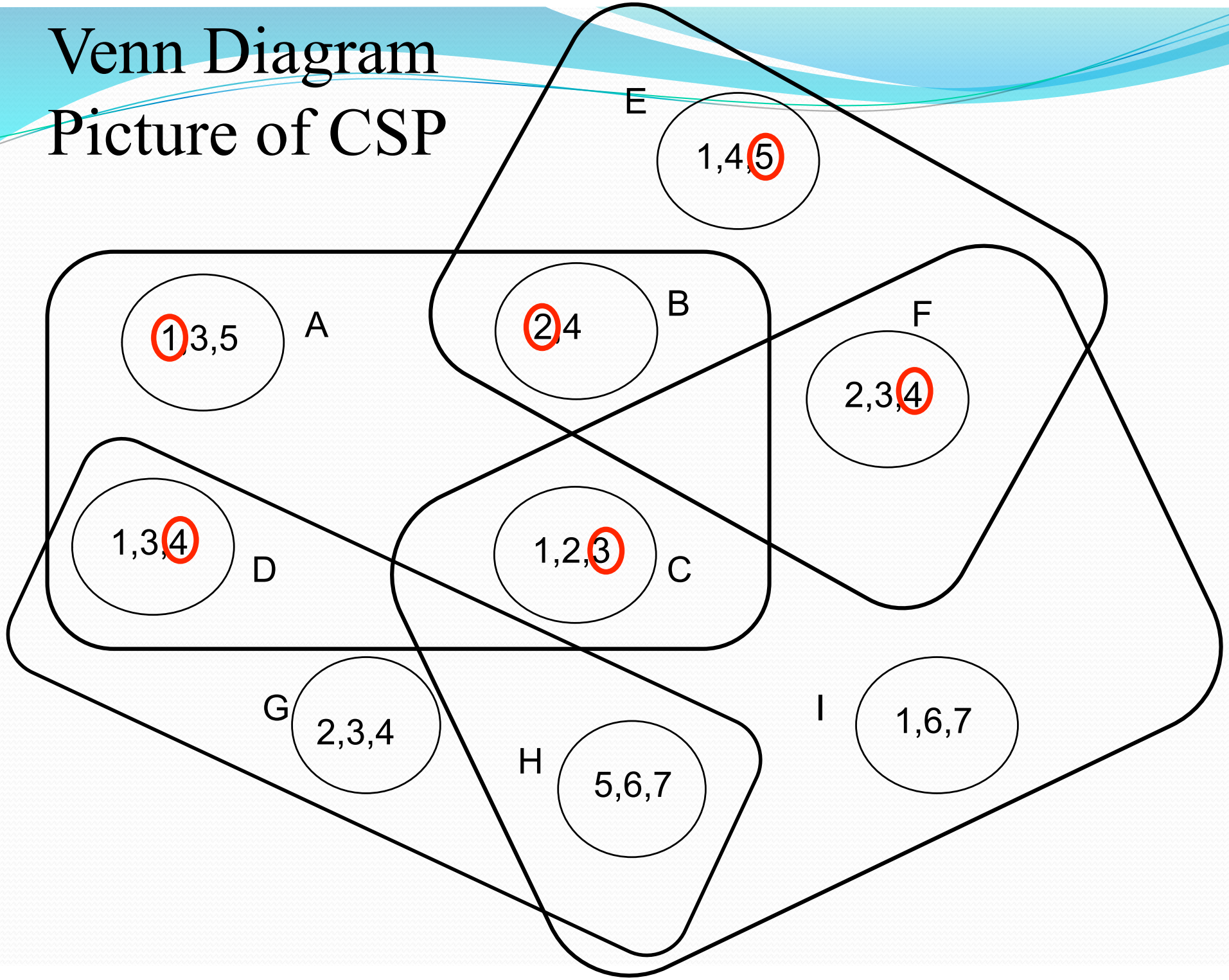
Thank You

PRESENTATION NAME

Company Name

© TemplatesWise.com

Parameter:

Maximum Constraint Size $k := \max_{C \in \mathcal{C}} |C|$

Venn Diagram Picture of CSP