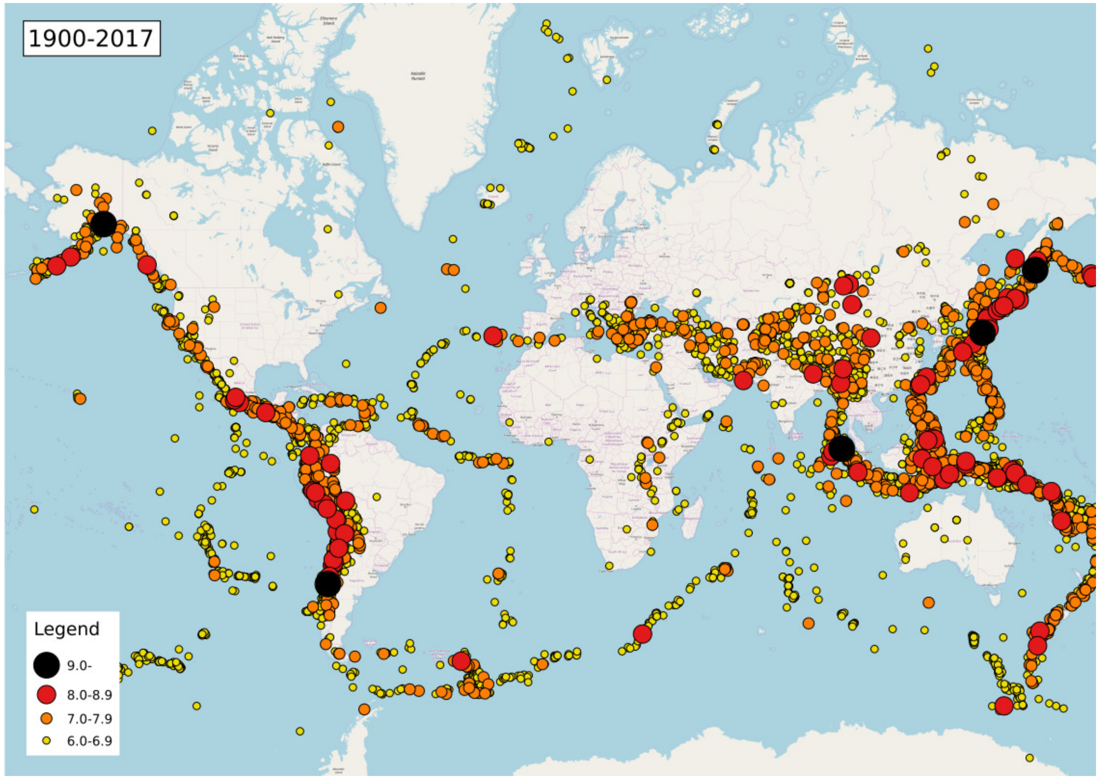
Computational Geometry
Hard Problems

Thomas Bläsius

Proportional Symbol Maps

Proportional Symbol Map (Example: Earthquakes)

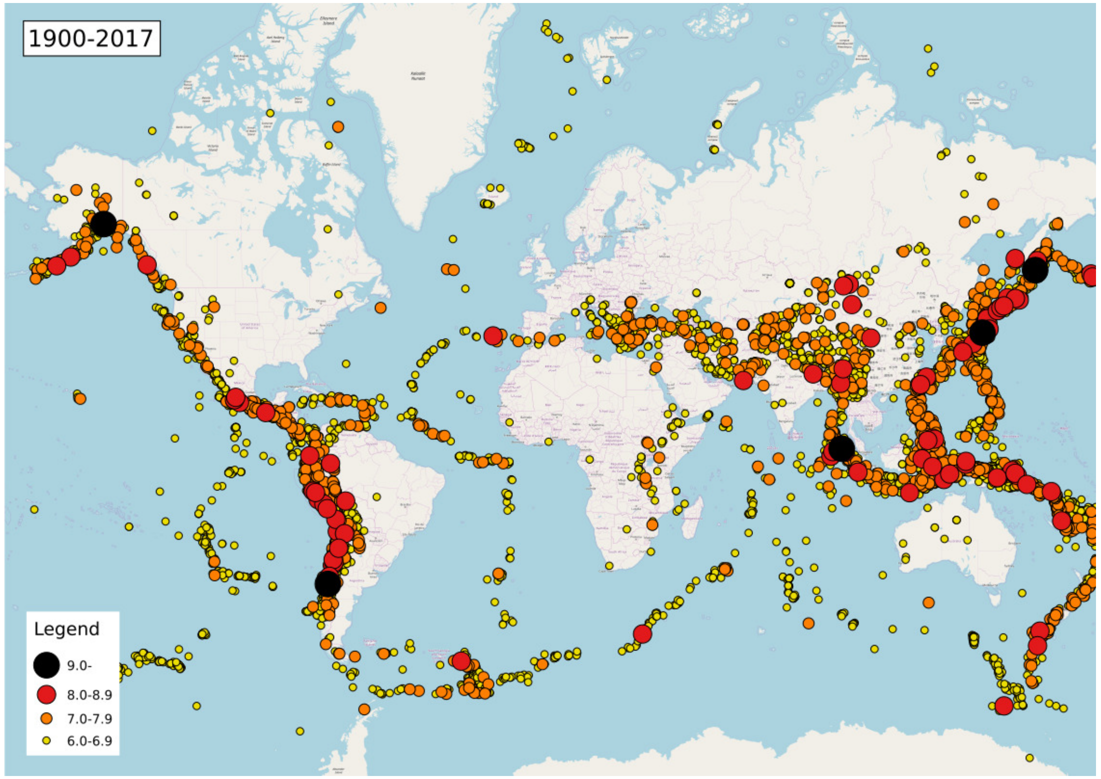
- visualizing weighted points on a map
- weight represented by disk size



Proportional Symbol Maps

Proportional Symbol Map (Example: Earthquakes)

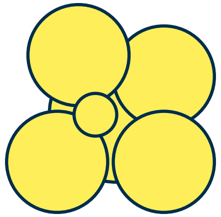
- visualizing weighted points on a map
- weight represented by disk size
- degree of freedom: z-order of overlapping disks



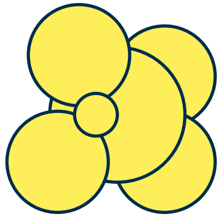
Proportional Symbol Maps

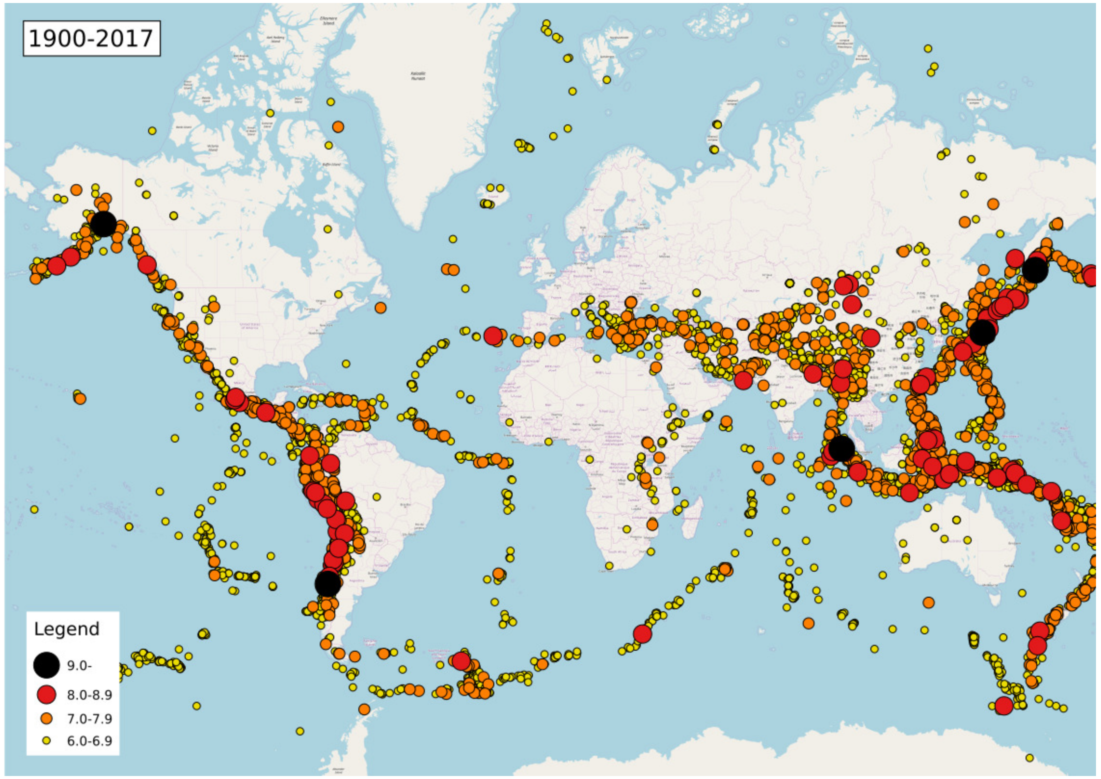
Proportional Symbol Map (Example: Earthquakes)

- visualizing weighted points on a map
- weight represented by disk size
- degree of freedom: z-order of overlapping disks
- readability depends on the order



vs.

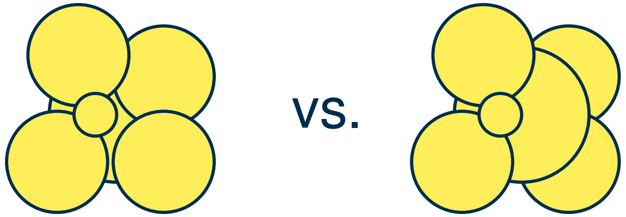


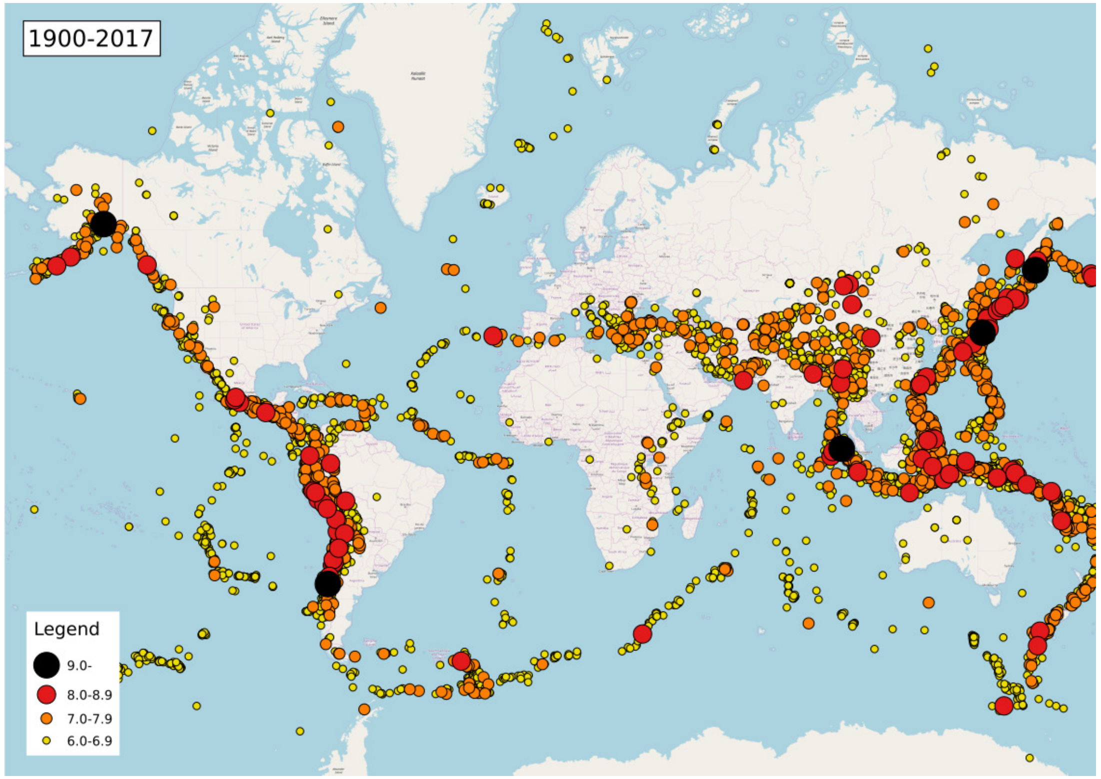


Proportional Symbol Maps

Proportional Symbol Map (Example: Earthquakes)

- visualizing weighted points on a map
- weight represented by disk size
- degree of freedom: z-order of overlapping disks
- readability depends on the order





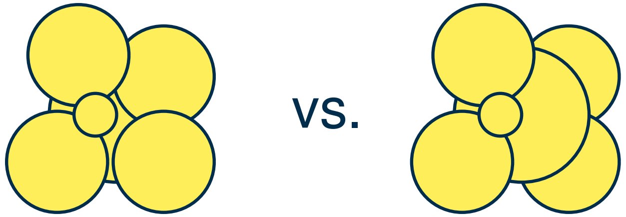
Problem

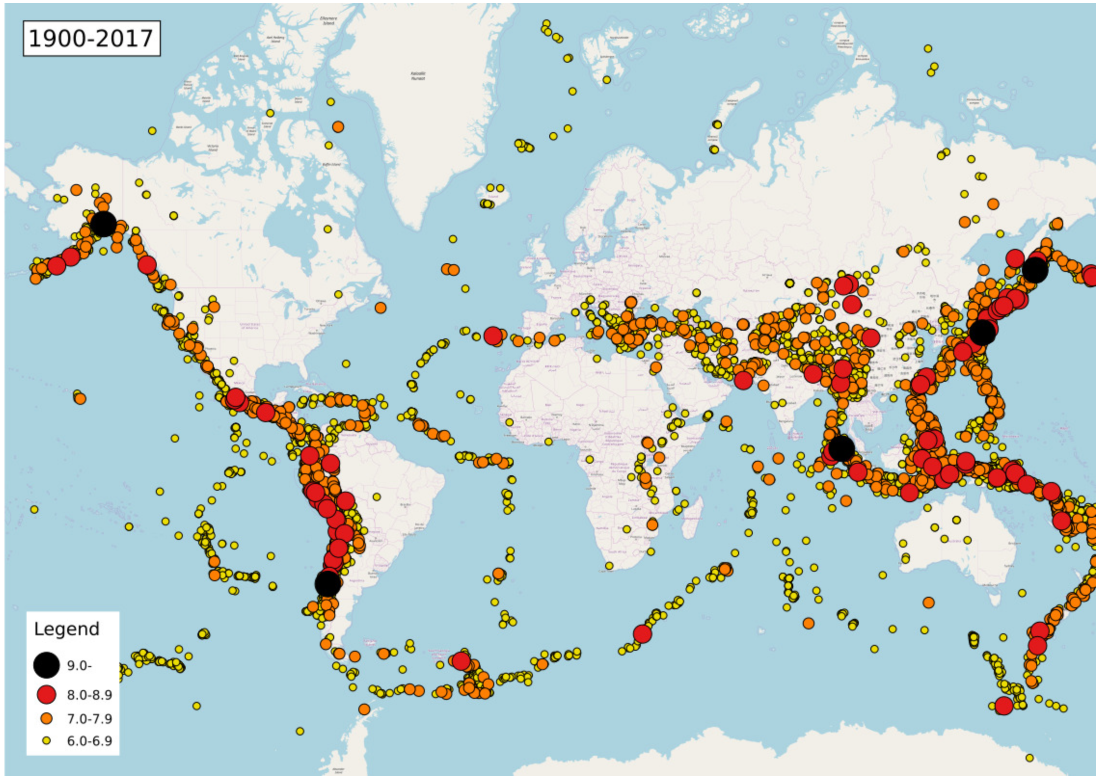
- given: set of disk with potentially different radii

Proportional Symbol Maps

Proportional Symbol Map (Example: Earthquakes)

- visualizing weighted points on a map
- weight represented by disk size
- degree of freedom: z-order of overlapping disks
- readability depends on the order





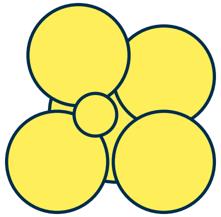
Problem

- given: set of disk with potentially different radii
- find: drawing that maximizes the visible border of each disk

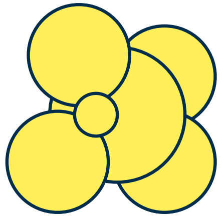
Proportional Symbol Maps

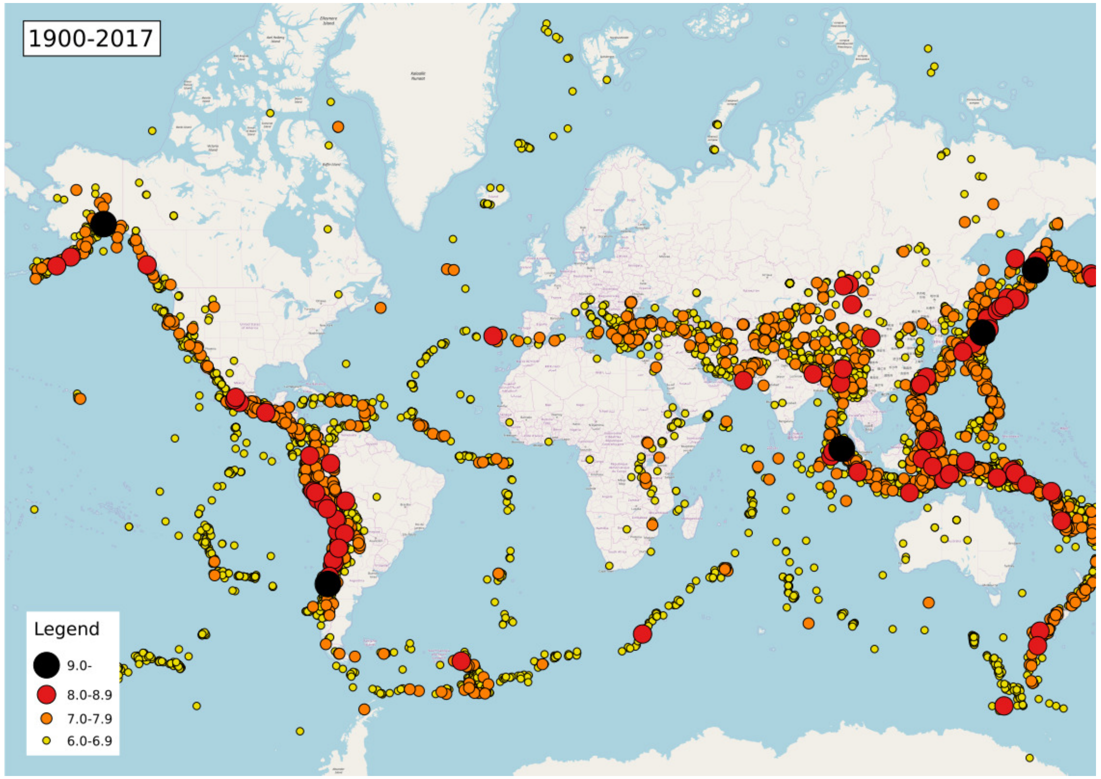
Proportional Symbol Map (Example: Earthquakes)

- visualizing weighted points on a map
- weight represented by disk size
- degree of freedom: z-order of overlapping disks
- readability depends on the order



vs.





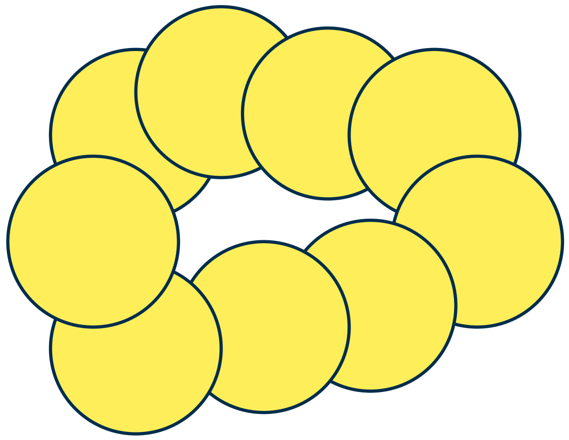
Problem

- given: set of disk with potentially different radii
- find: drawing that maximizes the visible border of each disk
- What is a valid drawing? What exactly does maximizing the visible border mean?

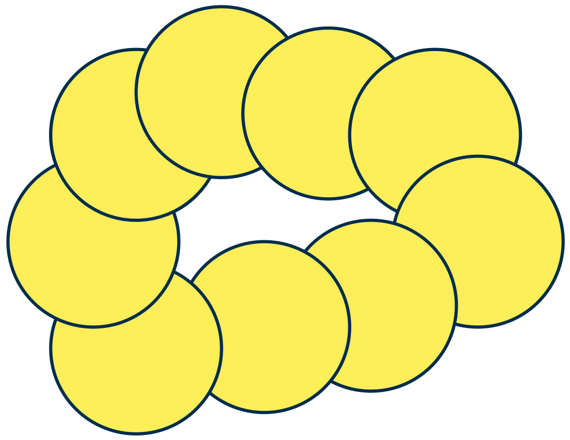
What Exactly Is The Problem?

Two Types Of Valid Drawings

- stacking: total z-order on all disks



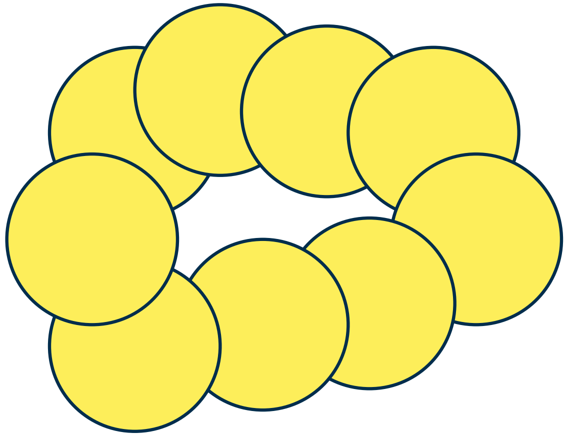
- physically realizable: buildabel with thin coins



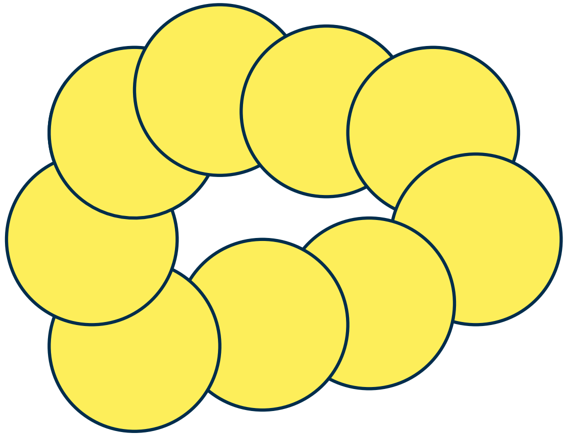
What Exactly Is The Problem?

Two Types Of Valid Drawings

- stacking: total z-order on all disks



- physically realizable: buildabel with thin coins

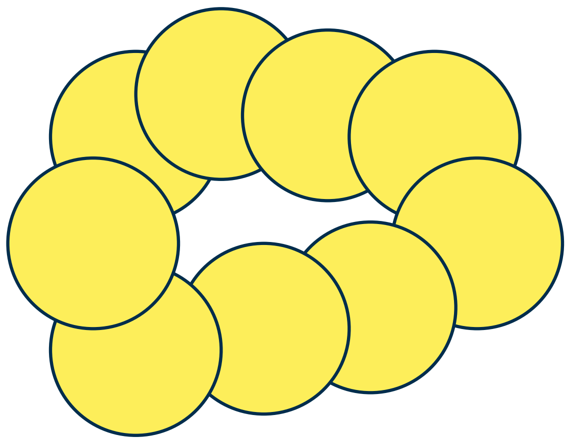


- every stacking is physically realizable, but not the other way round

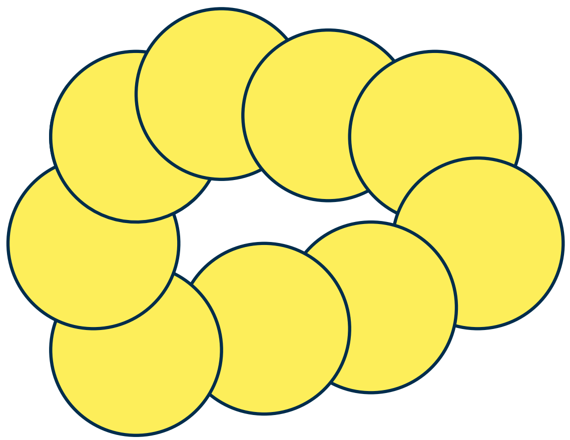
What Exactly Is The Problem?

Two Types Of Valid Drawings

- stacking: total z-order on all disks



- physically realizable: buildabel with thin coins



- every stacking is physically realizable, but not the other way round

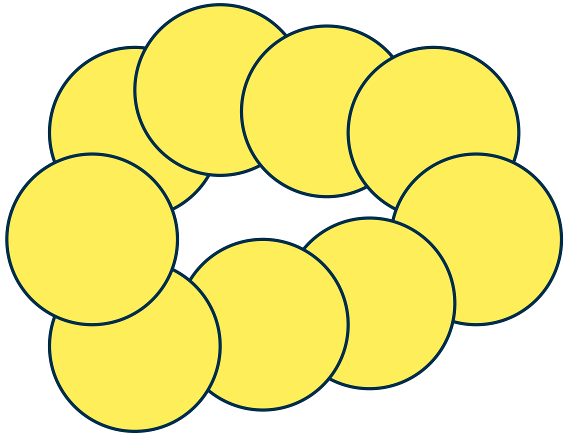
Two Optimization Problems

- Max-Min: maximize minimally visible border over all disks
- Max-Total: maximize the total visible border

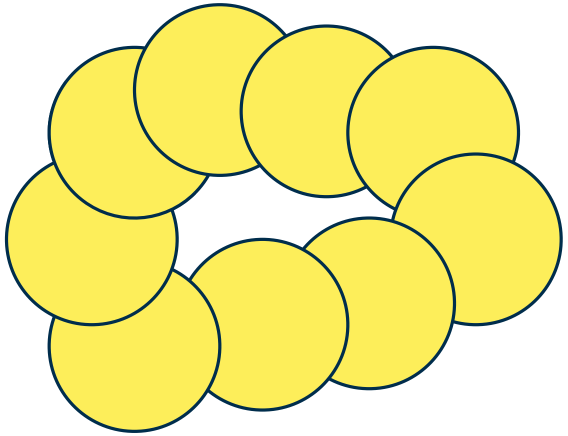
What Exactly Is The Problem?

Two Types Of Valid Drawings

- stacking: total z-order on all disks



- physically realizable: buildabel with thin coins



- every stacking is physically realizable, but not the other way round

Two Optimization Problems

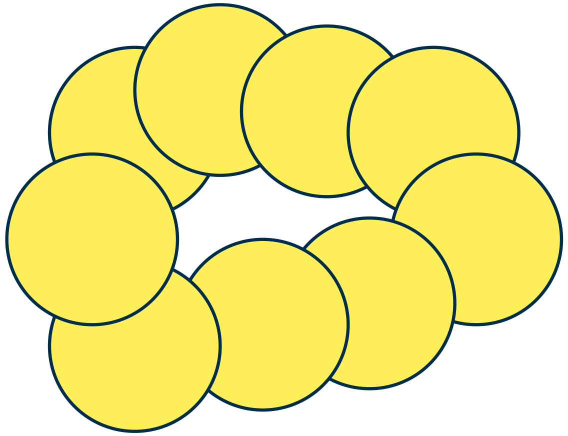
- Max-Min: maximize minimally visible border over all disks
- Max-Total: maximize the total visible border

	Max-Total	Max-Min
stacking	?	P
physically realizable	NP-hard	NP-hard

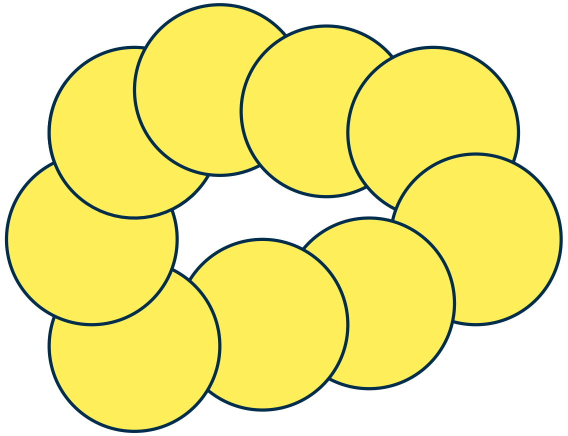
What Exactly Is The Problem?

Two Types Of Valid Drawings

- stacking: total z-order on all disks



- physically realizable: buildabel with thin coins



- every stacking is physically realizable, but not the other way round

Two Optimization Problems

- Max-Min: maximize minimally visible border over all disks
- Max-Total: maximize the total visible border

	Max-Total	Max-Min
stacking	?	P
physically realizable	NP-hard	NP-hard

← today

Useful NP-Hard SAT-Varints

Problem: 3-SAT
Boolean formula Φ in CNF, with ≤ 3 literals per clause. Is Φ satisfiable?

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_4)$$

Useful NP-Hard SAT-Varints

Problem: 3-SAT
Boolean formula Φ in CNF, with ≤ 3 literals per clause. Is Φ satisfiable?

$$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_4)$$

Problem: Monotone 3-SAT
Each clause has only positive or only negative literals.

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee x_4 \vee x_5) \wedge (\neg x_1 \vee \neg x_4)$$

Useful NP-Hard SAT-Varints

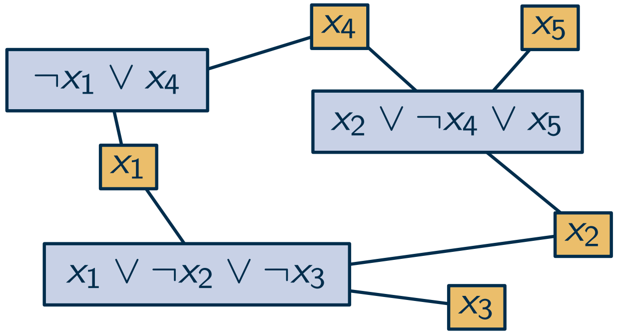
Problem: 3-SAT
Boolean formula Φ in CNF, with ≤ 3 literals per clause. Is Φ satisfiable?

$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_4)$

Problem: Monotone 3-SAT
Each clause has only positive or only negative literals.

$(\neg x_1 \vee \neg x_2 \vee \neg x_3)$
$\wedge (x_2 \vee x_4 \vee x_5)$
$\wedge (\neg x_1 \vee \neg x_4)$

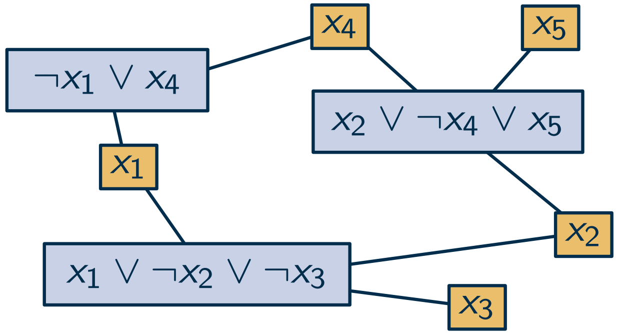
Problem: Planar 3-SAT
The clause–variable graph is planar.



Useful NP-Hard SAT-Varints

Problem: 3-SAT
Boolean formula Φ in CNF, with ≤ 3 literals per clause. Is Φ satisfiable?

$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_4)$

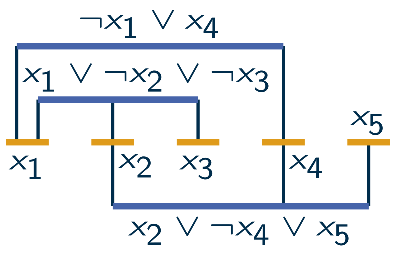


Problem: Monotone 3-SAT
Each clause has only positive or only negative literals.

$(\neg x_1 \vee \neg x_2 \vee \neg x_3)$
$\wedge (x_2 \vee x_4 \vee x_5)$
$\wedge (\neg x_1 \vee \neg x_4)$

Problem: Planar 3-SAT
The clause–variable graph is planar.

Problem: Rectilinear Planar 3-SAT
The clause–variable graph has a *rectilinear planar* drawing.



Rectilinear Planar Drawing

- vertices: horizontal segments
- edges: vertical segments
- all variable vertices on one line

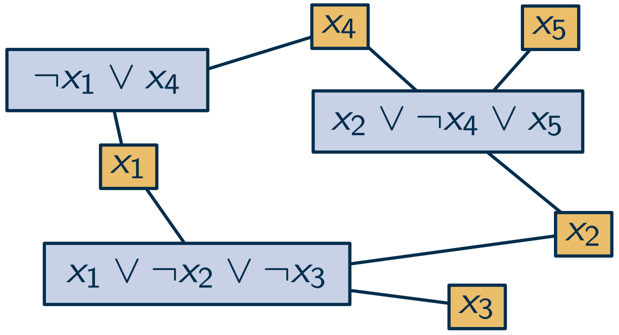
Useful NP-Hard SAT-Varints

Problem: 3-SAT
Boolean formula Φ in CNF, with ≤ 3 literals per clause. Is Φ satisfiable?

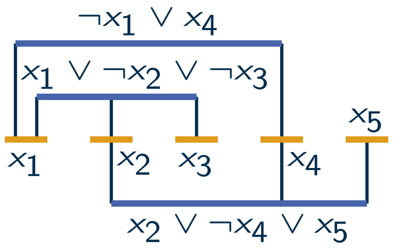
$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_4)$

Problem: Monotone 3-SAT
Each clause has only positive or only negative literals.

$(\neg x_1 \vee \neg x_2 \vee \neg x_3)$
$\wedge(x_2 \vee x_4 \vee x_5)$
$\wedge(\neg x_1 \vee \neg x_4)$



Problem: Planar 3-SAT
The clause–variable graph is planar.

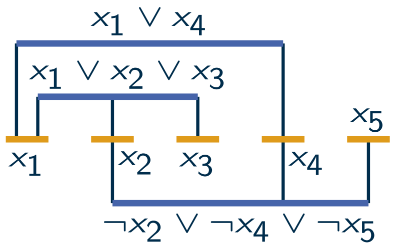


Rectilinear Planar Drawing

- vertices: horizontal segments
- edges: vertical segments
- all variable vertices on one line

Problem: Rectilinear Planar 3-SAT
The clause–variable graph has a *rectilinear planar* drawing.

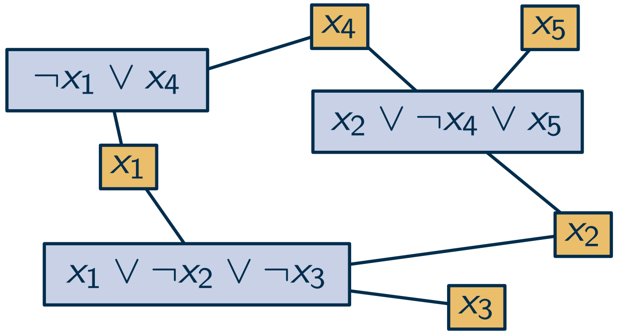
Problem: Planar Monotone 3-SAT
Clauses over/under the variables have only positive/negative literals.



Useful NP-Hard SAT-Varints

Problem: 3-SAT
Boolean formula Φ in CNF, with ≤ 3 literals per clause. Is Φ satisfiable?

$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_4)$

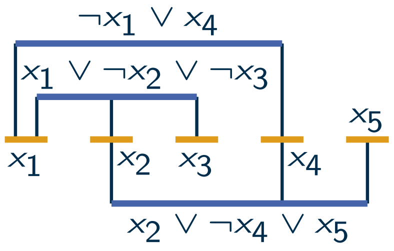


Problem: Monotone 3-SAT
Each clause has only positive or only negative literals.

$(\neg x_1 \vee \neg x_2 \vee \neg x_3)$
$\wedge (x_2 \vee x_4 \vee x_5)$
$\wedge (\neg x_1 \vee \neg x_4)$

Problem: Planar 3-SAT
The clause–variable graph is planar.

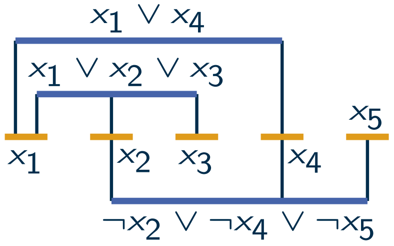
Problem: Rectilinear Planar 3-SAT
The clause–variable graph has a *rectilinear planar* drawing.



Rectilinear Planar Drawing

- vertices: horizontal segments
- edges: vertical segments
- all variable vertices on one line

Problem: Planar Monotone 3-SAT
Clauses over/under the variables have only positive/negative literals.



Note: allowing clauses < 2 is important here

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\widehat{=}$ decision how disks overlap

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\hat{=}$ decision how disks overlap
- satisfying all clauses $\hat{=}$ for each disk, a big part of its border is visible

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\widehat{=}$ decision how disks overlap
- satisfying all clauses $\widehat{=}$ for each disk, a big part of its border is visible

Needed Building Blocks

- variables: n independent decisions, everything else is forced

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\hat{=}$ decision how disks overlap
- satisfying all clauses $\hat{=}$ for each disk, a big part of its border is visible

Needed Building Blocks

- variables: n independent decisions, everything else is forced
- clauses: problematic $\Leftrightarrow$ three specific decisions are wrong

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\hat{=}$ decision how disks overlap
- satisfying all clauses $\hat{=}$ for each disk, a big part of its border is visible

Needed Building Blocks

- variables: n independent decisions, everything else is forced
- clauses: problematic $\Leftrightarrow$ three specific decisions are wrong
- information transport
  - propagate decisions made at the variables to the clauses

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\widehat{=}$ decision how disks overlap
- satisfying all clauses $\widehat{=}$ for each disk, a big part of its border is visible

Needed Building Blocks

- variables: n independent decisions, everything else is forced
- clauses: problematic $\Leftrightarrow$ three specific decisions are wrong
- information transport
  - propagate decisions made at the variables to the clauses
  - must be possible for positive and negative literals

Reductions From Planar Monotone 3-SAT

General Mindset

- we want to model a given 3-SAT instance
- our modeling language are overlapping disks
- variable assignment $\hat{=}$ decision how disks overlap
- satisfying all clauses $\hat{=}$ for each disk, a big part of its border is visible

Needed Building Blocks

- variables: n independent decisions, everything else is forced
- clauses: problematic $\Leftrightarrow$ three specific decisions are wrong
- information transport
  - propagate decisions made at the variables to the clauses
  - must be possible for positive and negative literals
  - transport channel can be faulty in one direction: flip from satisfied to unsatisfied literal is ok

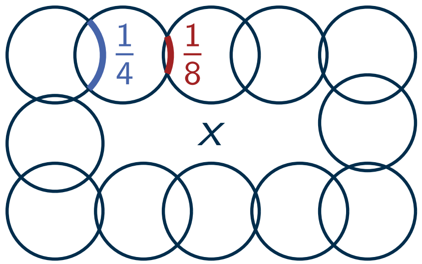
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

Variable Gadget

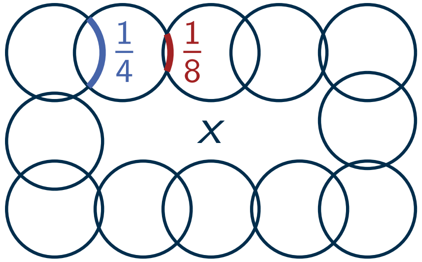


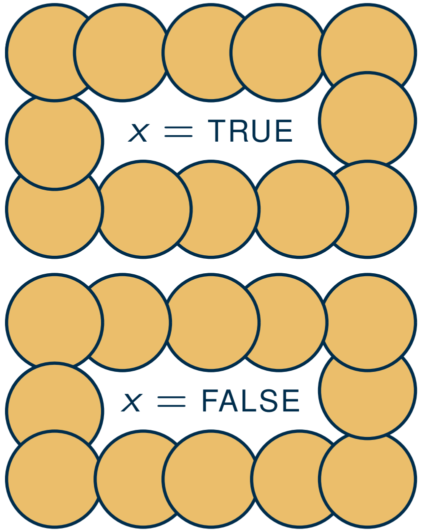
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk



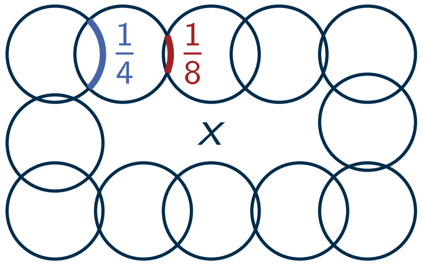


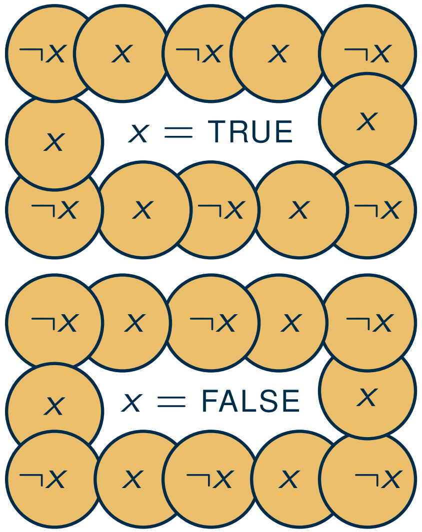
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk



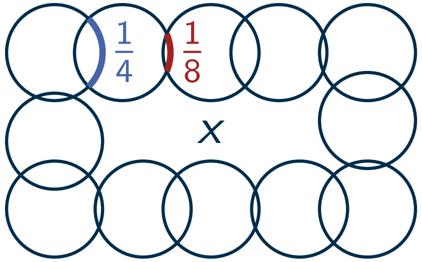


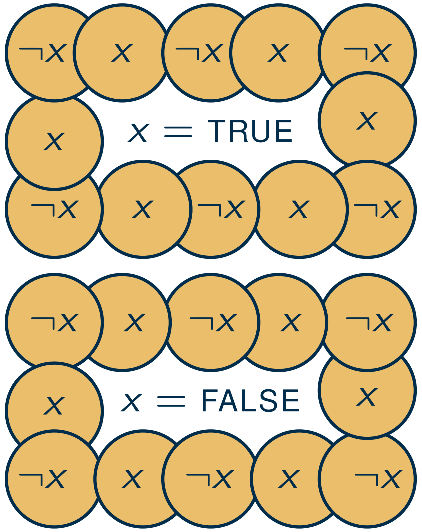
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

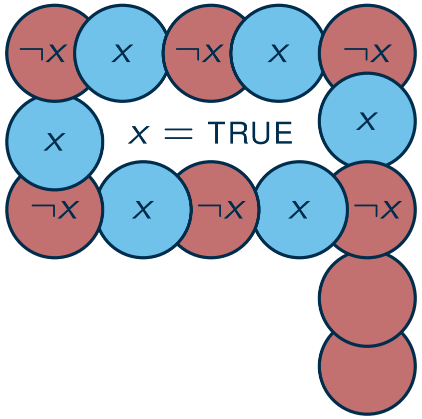
Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk





Transport Gadget



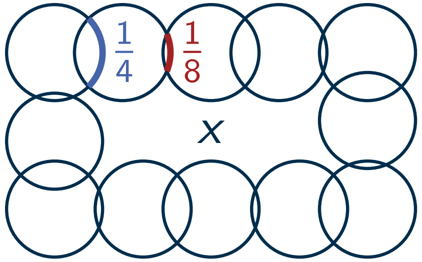
- chain starting at a $\neg x$ (with $\neg x =$ FALSE)

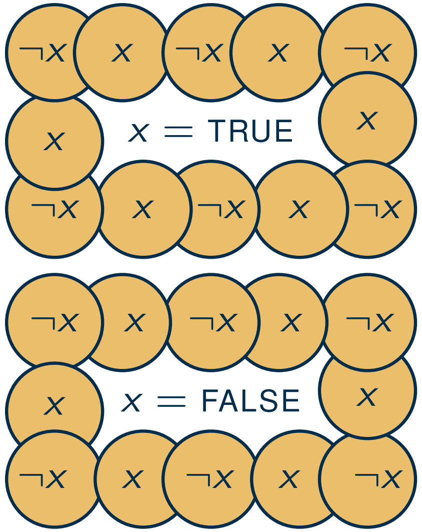
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

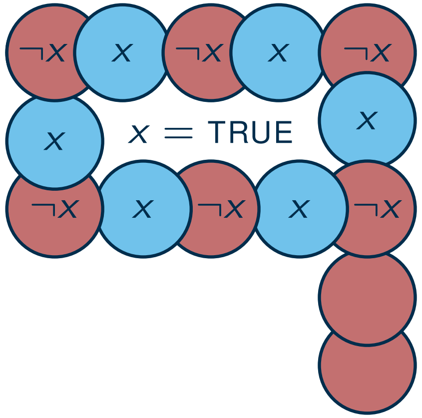
Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk





Transport Gadget



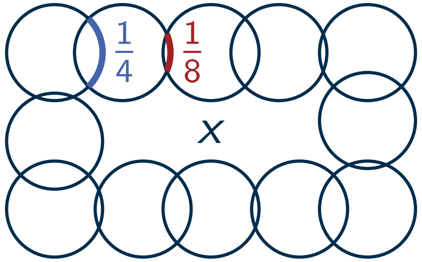
- chain starting at a $\neg x$ (with $\neg x =$ FALSE)
  - every decision is forced $\rightarrow \frac{1}{4}$ of last disk covered
  - transports information $\neg x =$ FALSE to (almost) arbitrary position

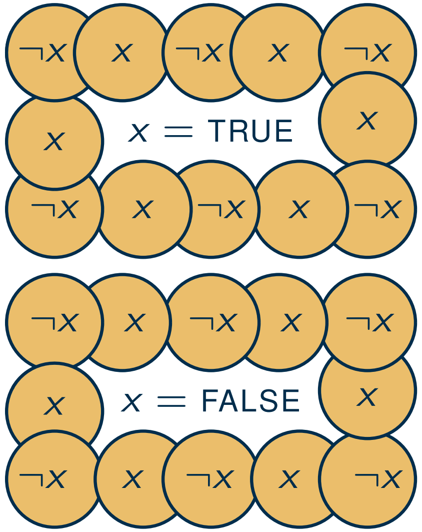
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

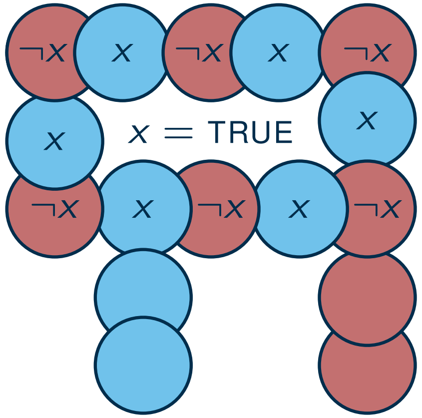
Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk





Transport Gadget



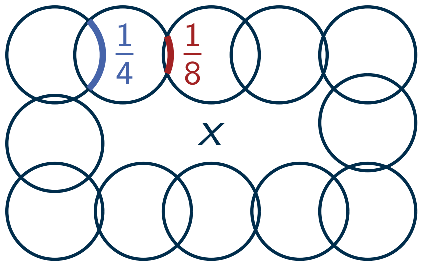
- chain starting at a $\neg x$ (with $\neg x = \text{FALSE}$)
  - every decision is forced $\rightarrow \frac{1}{4}$ of last disk covered
  - transports information $\neg x = \text{FALSE}$ to (almost) arbitrary position
- chain starts at an x (instead of $\neg x$) $\rightarrow$ last disk can be completely visible

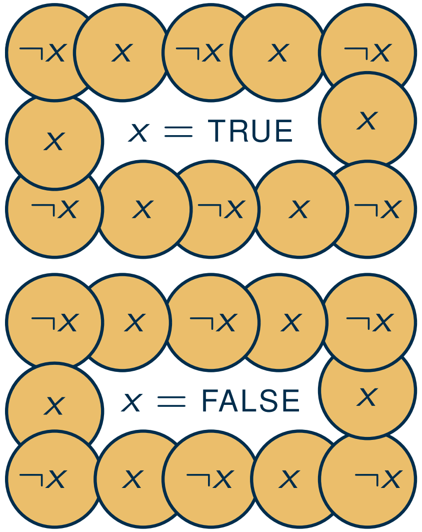
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

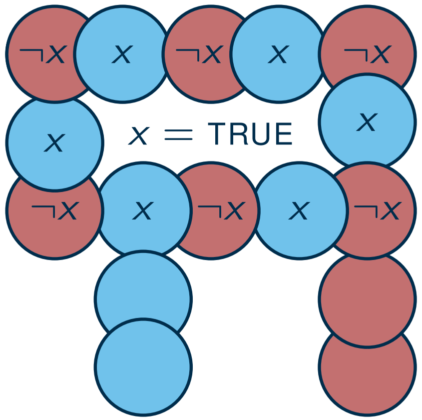
Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk





Transport Gadget



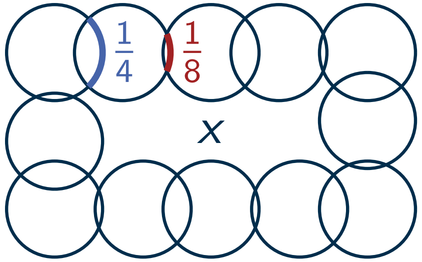
- chain starting at a $\neg x$ (with $\neg x =$ FALSE)
  - every decision is forced $\rightarrow \frac{1}{4}$ of last disk covered
  - transports information $\neg x =$ FALSE to (almost) arbitrary position
- chain starts at an x (instead of $\neg x$) $\rightarrow$ last disk can be completely visible
- inverted behavior for the $x =$ FALSE configuration

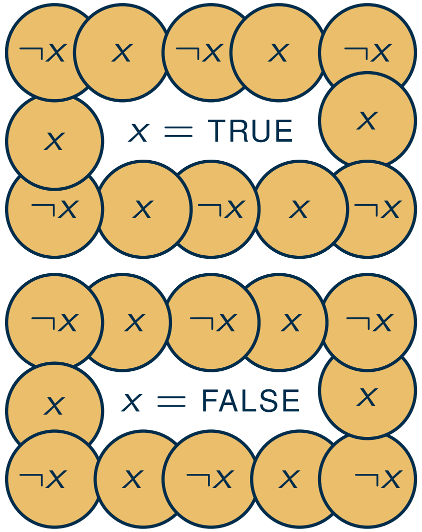
Gadgets

YES-instance: for every disk, $\geq 3/4$ of its border is visible

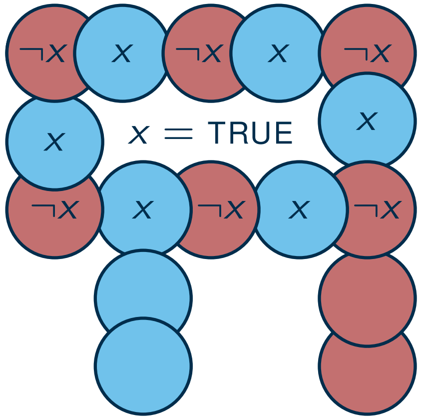
Variable Gadget

- only two configurations possible
- every different configuration covers $> 1/4$ of a disk





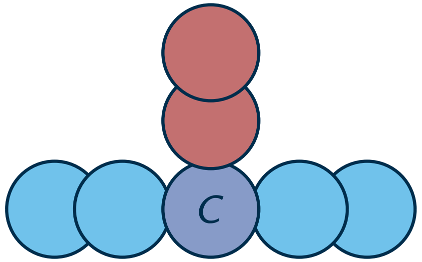
Transport Gadget



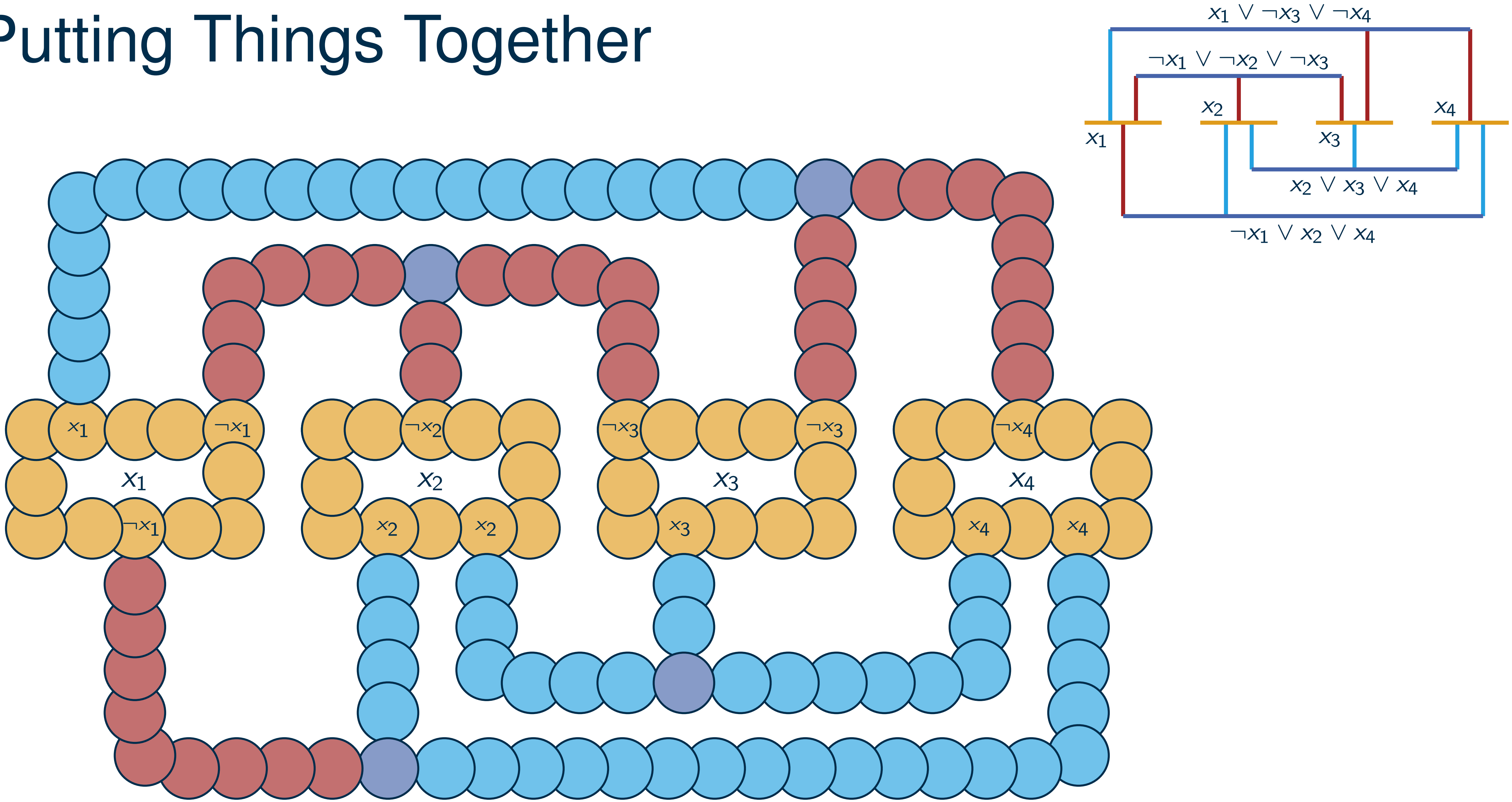
- chain starting at a $\neg x$ (with $\neg x =$ FALSE)
  - every decision is forced $\rightarrow \frac{1}{4}$ of last disk covered
  - transports information $\neg x =$ FALSE to (almost) arbitrary position
- chain starts at an x (instead of $\neg x$) $\rightarrow$ last disk can be completely visible
- inverted behavior for the $x =$ FALSE configuration

Clause Gadget

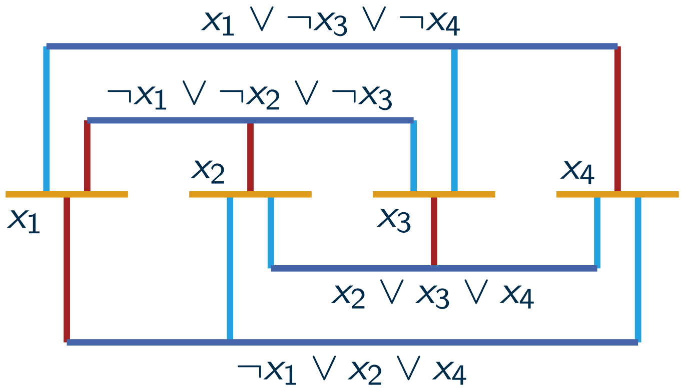
- C must overlap ≥ 1 of its neighbors $\rightarrow \geq 1$ chain comes from a `true` literal

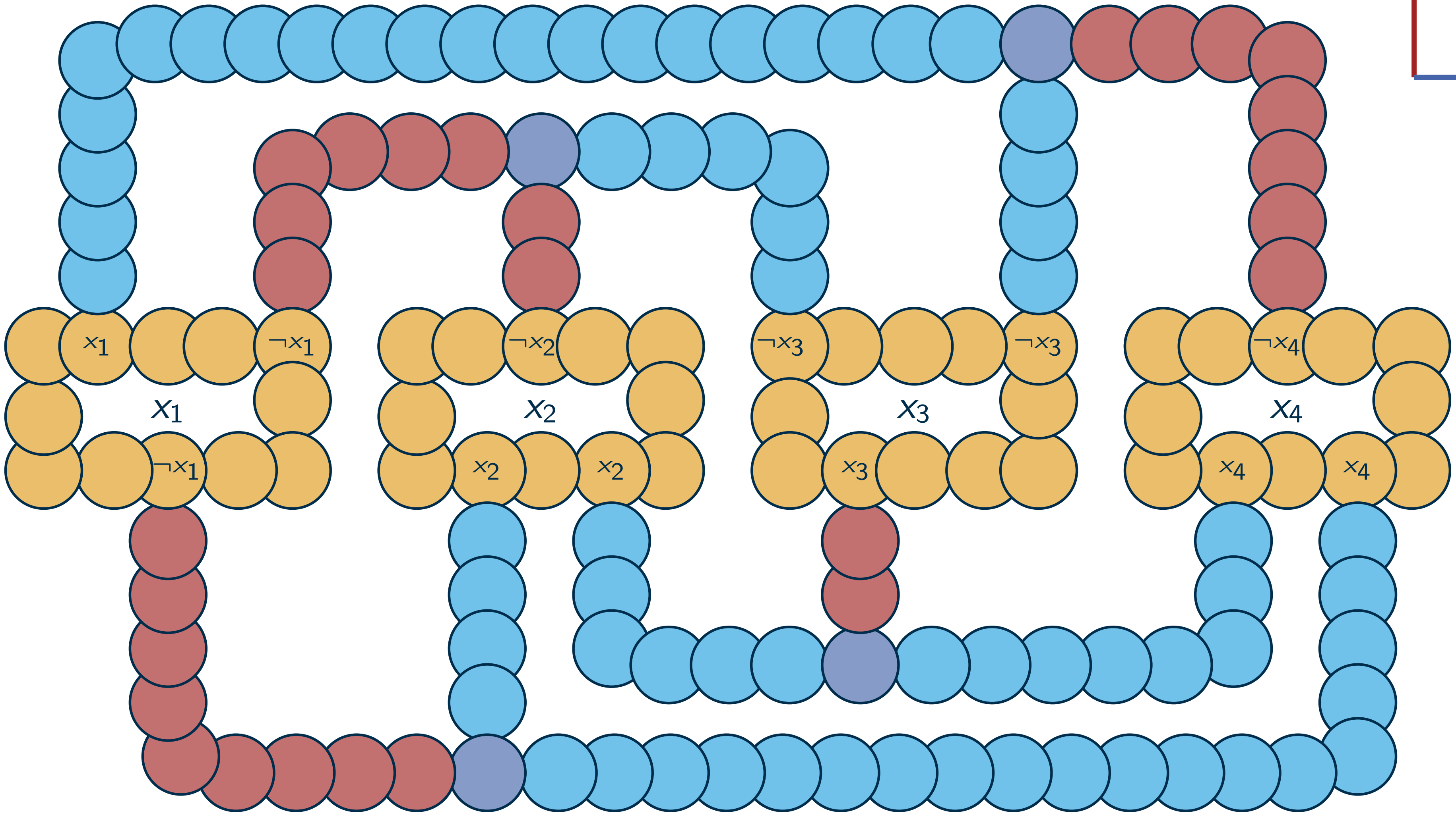


Putting Things Together



Putting Things Together





What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

Details Of The Reduction

(the big picture should be more or less clear already)

What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

Details Of The Reduction

(the big picture should be more or less clear already)

- size of the variable gadget: dependent on number of appearances in clauses

What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

Details Of The Reduction

(the big picture should be more or less clear already)

- size of the variable gadget: dependent on number of appearances in clauses
- length and shape of the transport gadget
  - follows the rectilinear drawing of the 3-SAT instance in the input

What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

Details Of The Reduction

(the big picture should be more or less clear already)

- size of the variable gadget: dependent on number of appearances in clauses
- length and shape of the transport gadget
  - follows the rectilinear drawing of the 3-SAT instance in the input
  - we can assume: drawing on the grid with polynomially bounded coordinates

What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

Details Of The Reduction

(the big picture should be more or less clear already)

- size of the variable gadget: dependent on number of appearances in clauses
- length and shape of the transport gadget
  - follows the rectilinear drawing of the 3-SAT instance in the input
  - we can assume: drawing on the grid with polynomially bounded coordinates
- resulting instance has polynomial size and reduction runs in polynomial time

What Is Left To Show?

Theorem
Deciding whether there is a physically realizable configuration that shows $3/4$ of the border of each disk is NP-hard.

Details Of The Reduction

(the big picture should be more or less clear already)

- size of the variable gadget: dependent on number of appearances in clauses
- length and shape of the transport gadget
  - follows the rectilinear drawing of the 3-SAT instance in the input
  - we can assume: drawing on the grid with polynomially bounded coordinates
- resulting instance has polynomial size and reduction runs in polynomial time

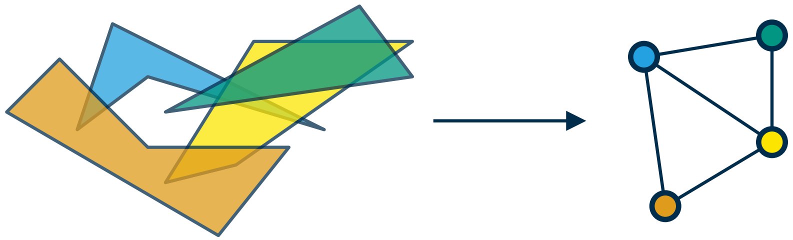
Correctness

- $3/4$ of the border of each disk visible $\Rightarrow$ formula satisfiable
- formula satisfiable $\Rightarrow$ $3/4$ of the border of each disk visible

Why?

Unit Disk Graphs

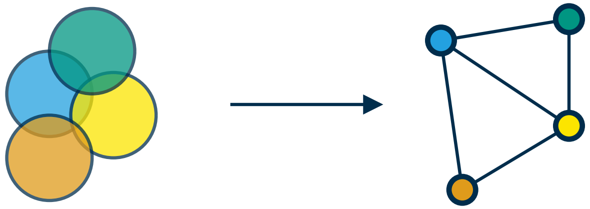
Definition
Set of geometric objects V defines **intersection graph** $G = (V, E)$ with $uv \in E \Leftrightarrow u \cap v \neq \emptyset$.



Unit Disk Graphs

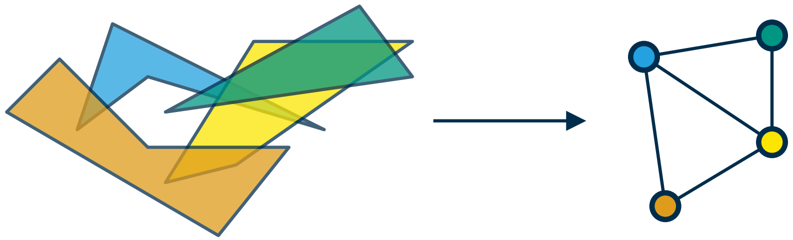
Definition
Set of geometric objects V defines **intersection graph** $G = (V, E)$ with $uv \in E \Leftrightarrow u \cap v \neq \emptyset$.

Definition
A graph is a **unit disk graph** if it is the intersection graph of disks of radius 1.

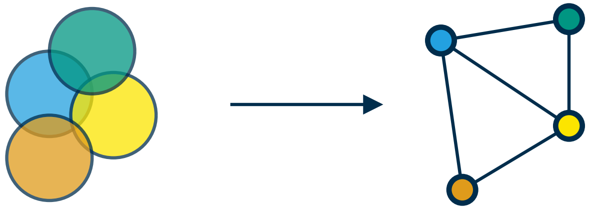


Unit Disk Graphs

Definition
Set of geometric objects V defines **intersection graph** $G = (V, E)$ with $uv \in E \Leftrightarrow u \cap v \neq \emptyset$.

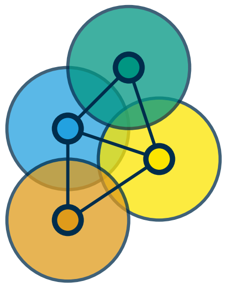


Definition
A graph is a **unit disk graph** if it is the intersection graph of disks of radius 1.



Recognition Problem
Is a given graph a unit disk graph?

Basic Observations

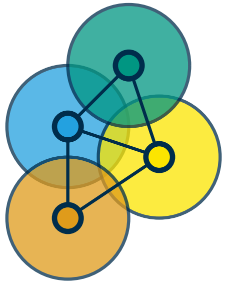


Goal: reduce planar monotone 3-SAT to unit disk graphs recognition

Useful Basic Observations

- equivalent: are there vertex positions such that $\text{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$?

Basic Observations

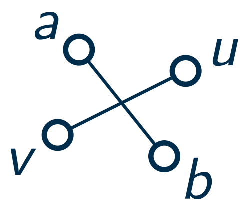


Goal: reduce planar monotone 3-SAT to unit disk graphs recognition

Useful Basic Observations

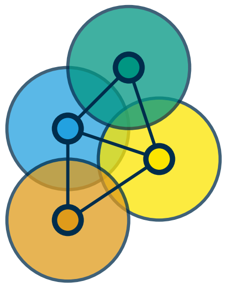
- equivalent: are there vertex positions such that $\mathrm{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$?
- two edges ab and uv cross in this representation $\Rightarrow$ three of the vertices a, b, u, v form a triangle

Why?



Basic Observations

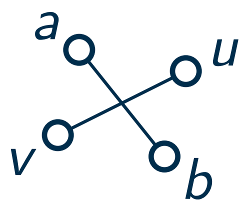
Goal: reduce planar monotone 3-SAT to unit disk graphs recognition



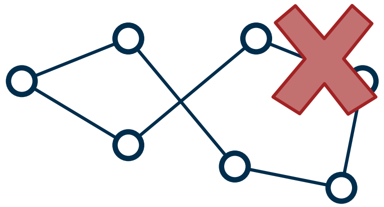
Useful Basic Observations

- equivalent: are there vertex positions such that $\text{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$?
- two edges ab and uv cross in this representation $\Rightarrow$ three of the vertices a, b, u, v form a triangle

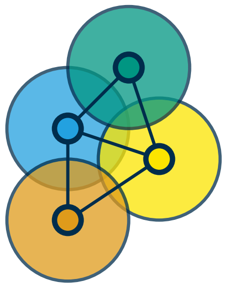
Why?



- induced cycles are planar



Basic Observations

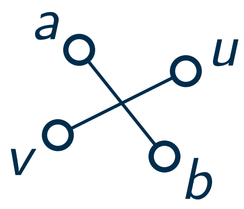


Goal: reduce planar monotone 3-SAT to unit disk graphs recognition

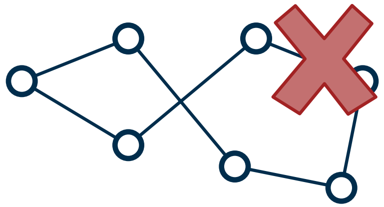
Useful Basic Observations

- equivalent: are there vertex positions such that $\mathrm{dist}(u, v) \leq 2 \Leftrightarrow uv \in E$?
- two edges ab and uv cross in this representation $\Rightarrow$ three of the vertices a, b, u, v form a triangle

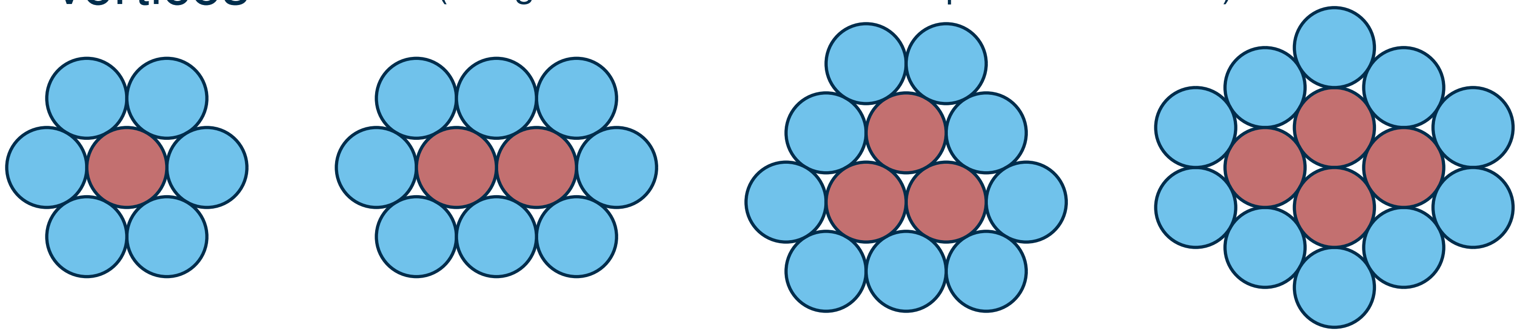
Why?



- induced cycles are planar

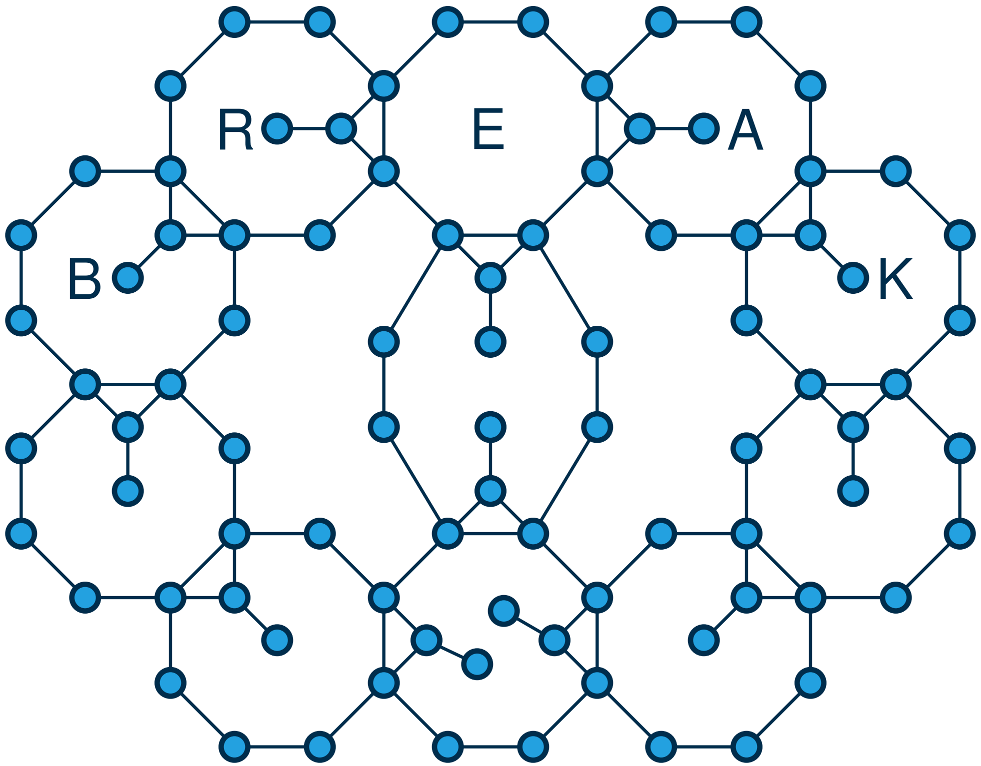


- cycles contain a limited number of independent vertices (*i*-cage contains at most *i* independent vertices)

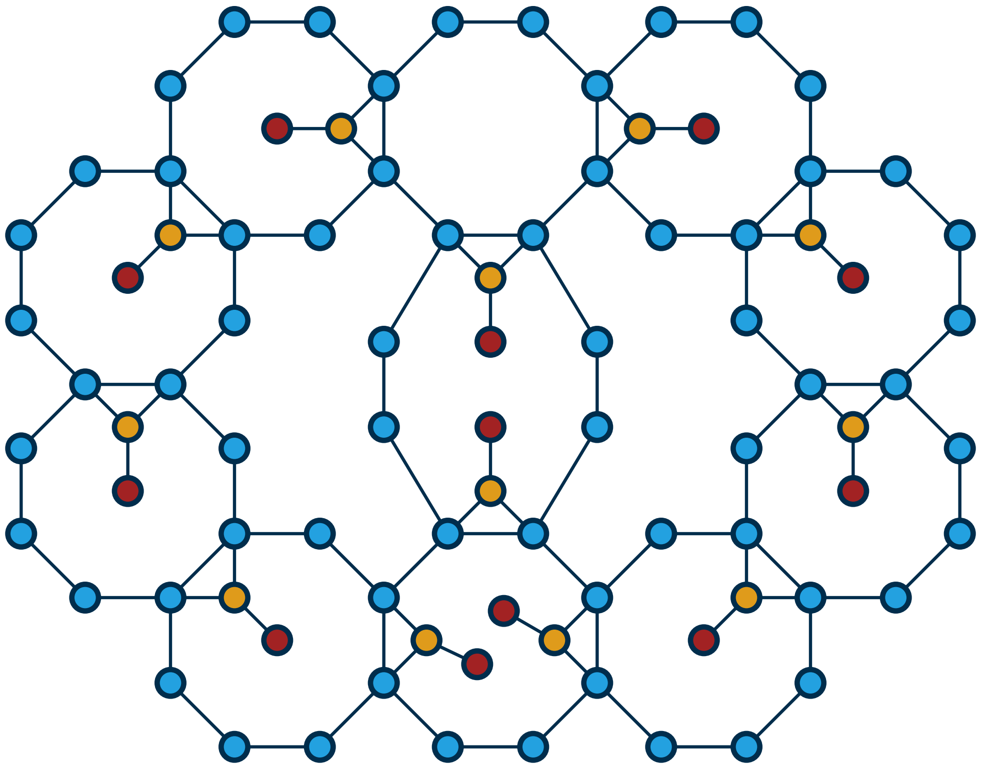


0-cage 1-cage 2-cage 3-cage

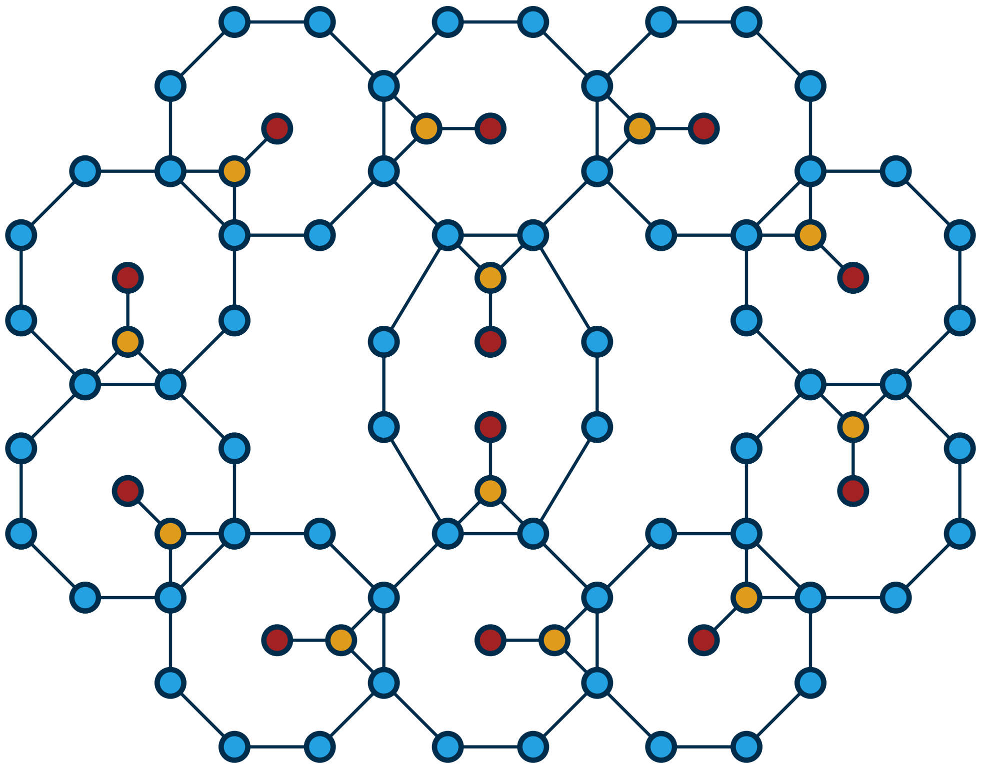
Unit Disk Graph Or Not?



Unit Disk Graph Or Not?



Unit Disk Graph Or Not?



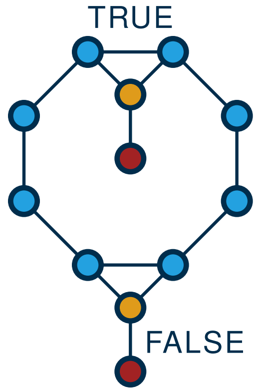
Gadgets

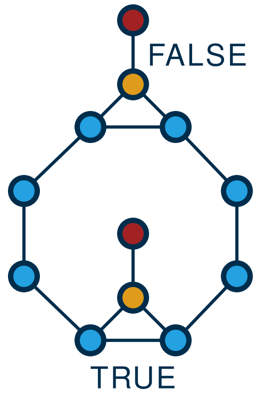
Gadgets We Need: variable, clause, transport

Gadgets

Gadgets We Need: variable, clause, transport

Variable Gadget

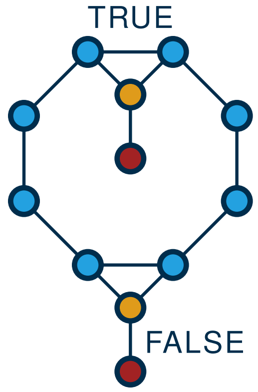


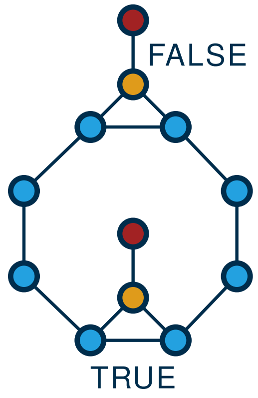


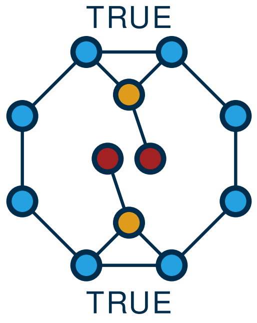
Gadgets

Gadgets We Need: variable, clause, transport

Variable Gadget





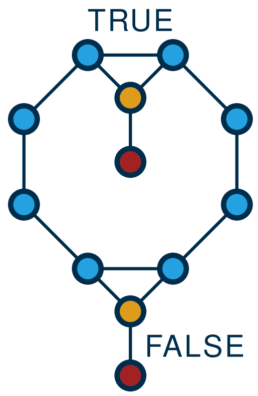


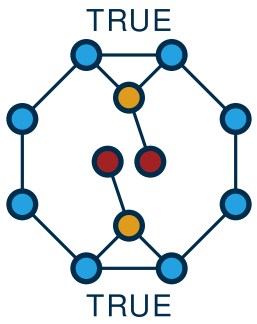
not possible (1-cage with two independent vertices)

Gadgets

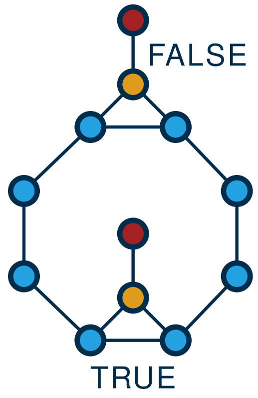
Gadgets We Need: variable, clause, transport

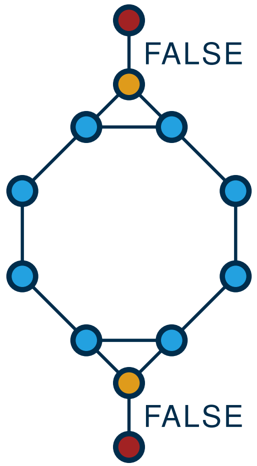
Variable Gadget





not possible (1-cage with two independent vertices)



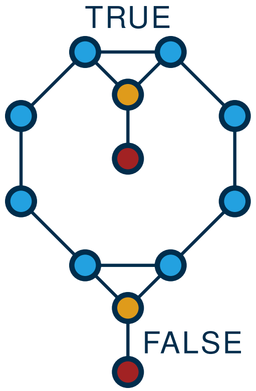


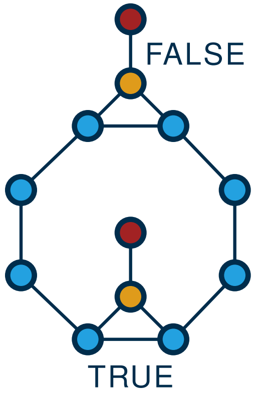
possible but never helpful (can basically be ignored)

Gadgets

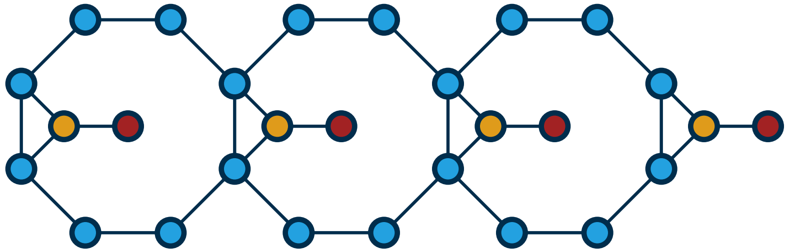
Gadgets We Need: variable, clause, transport

Variable Gadget





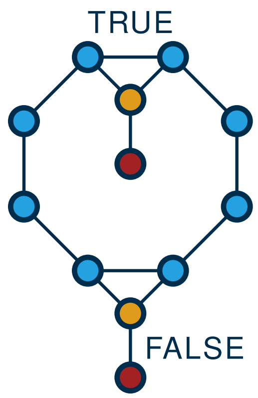
Transporting Information

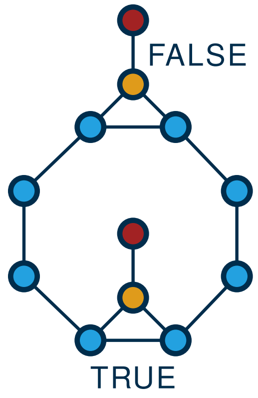


Gadgets

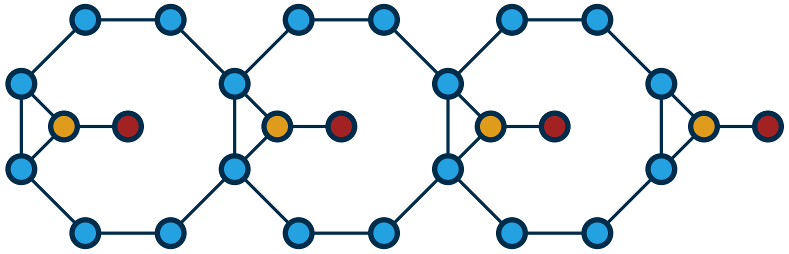
Gadgets We Need: variable, clause, transport

Variable Gadget

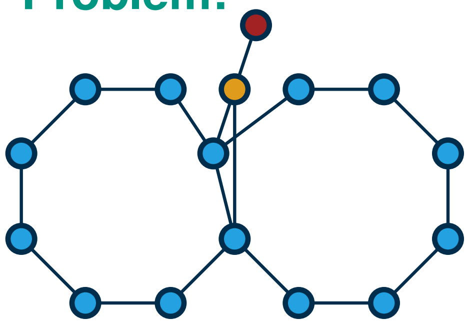




Transporting Information



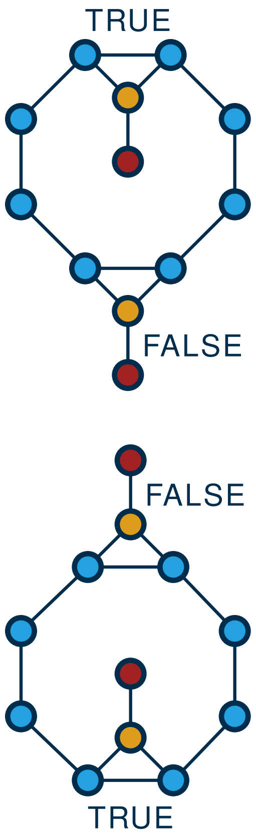
Problem:



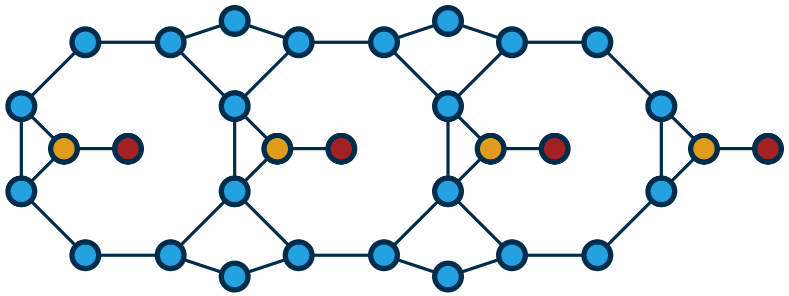
Gadgets

Gadgets We Need: variable, clause, transport

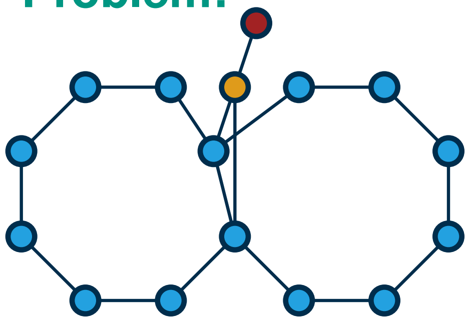
Variable Gadget



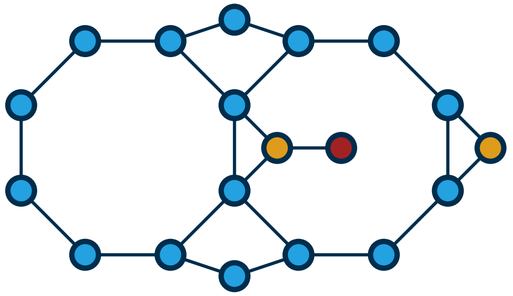
Transporting Information



Problem:



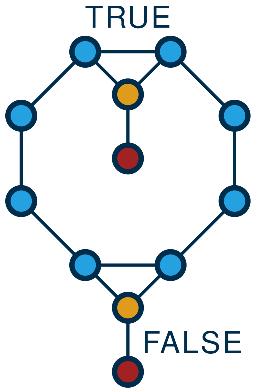
Solution:

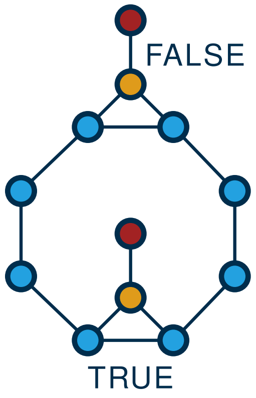


Gadgets

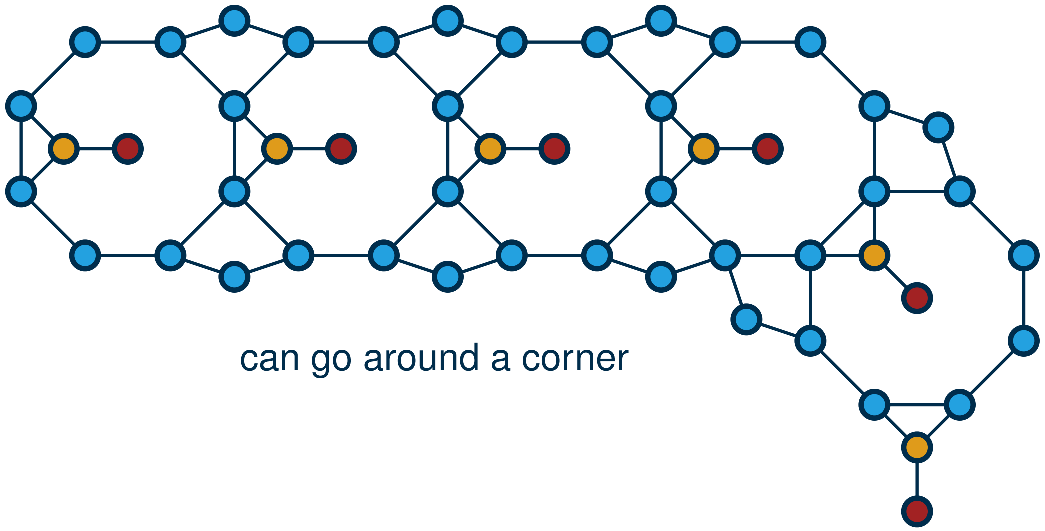
Gadgets We Need: variable, clause, transport

Variable Gadget





Transporting Information

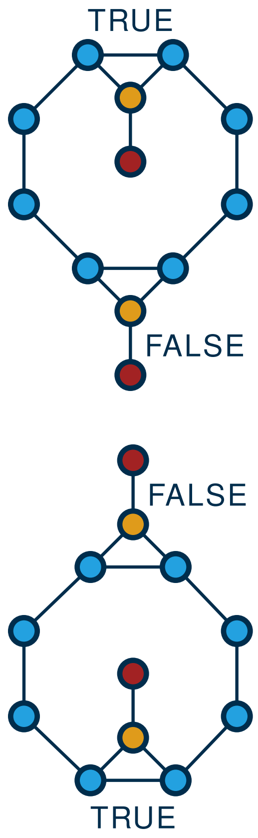


can go around a corner

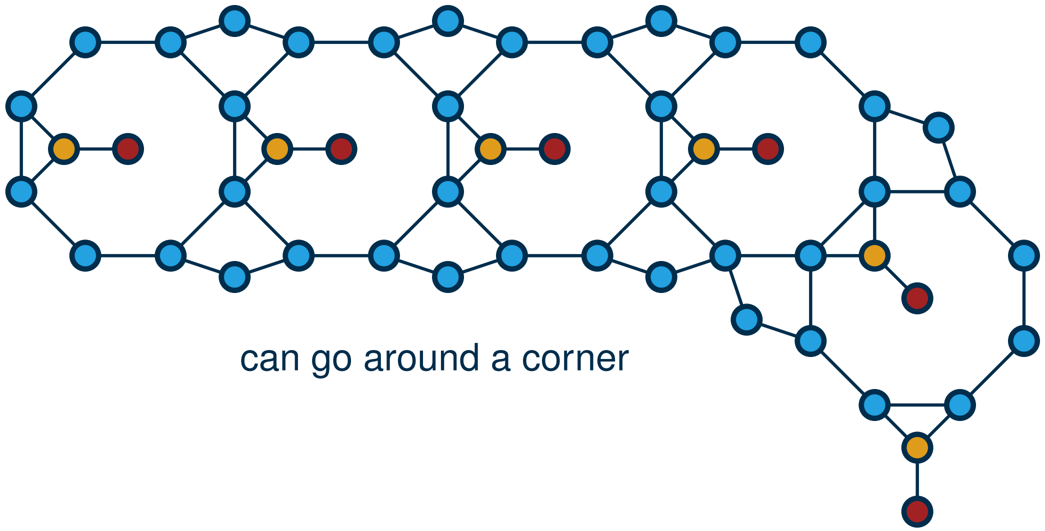
Gadgets

Gadgets We Need: variable, clause, transport

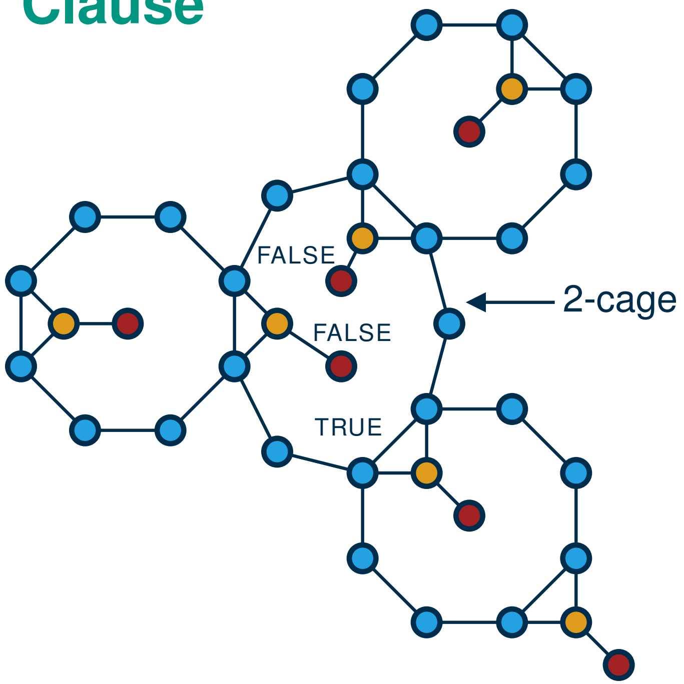
Variable Gadget



Transporting Information



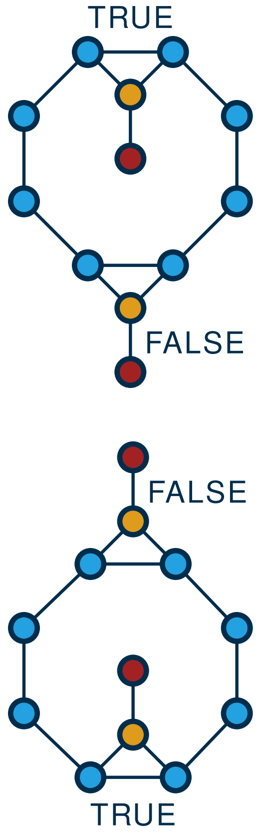
Clause



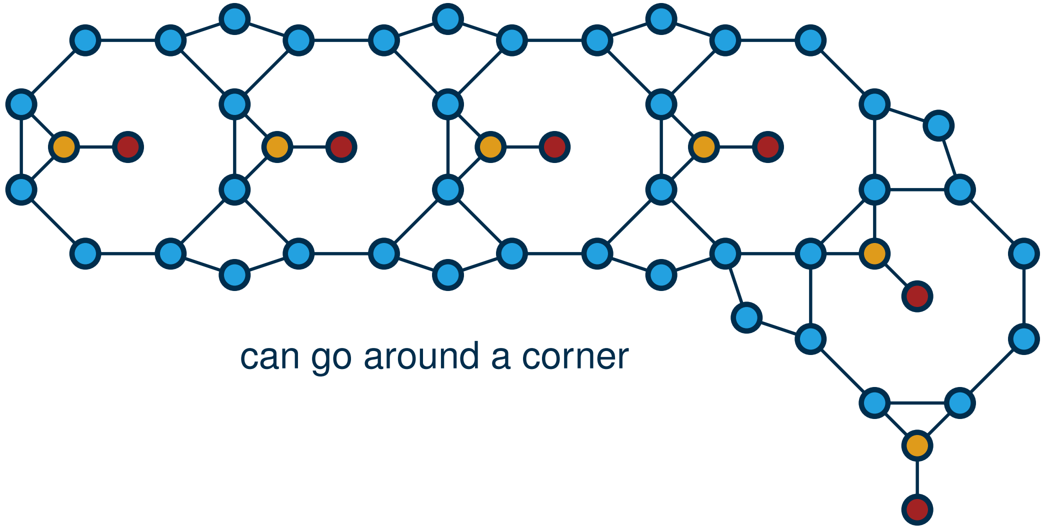
Gadgets

Gadgets We Need: variable, clause, transport

Variable Gadget

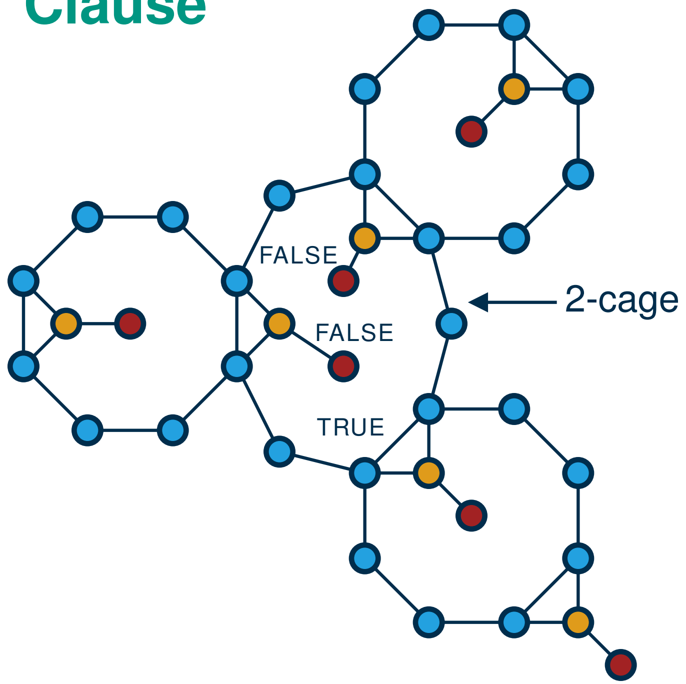


Transporting Information



What Is Missing?

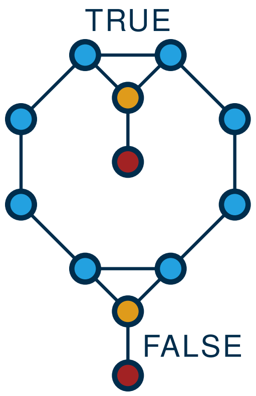
Clause

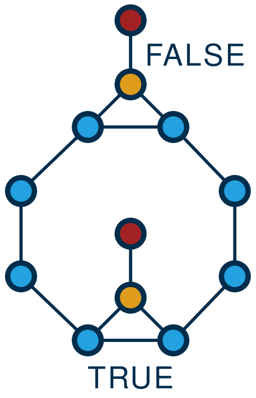


Gadgets

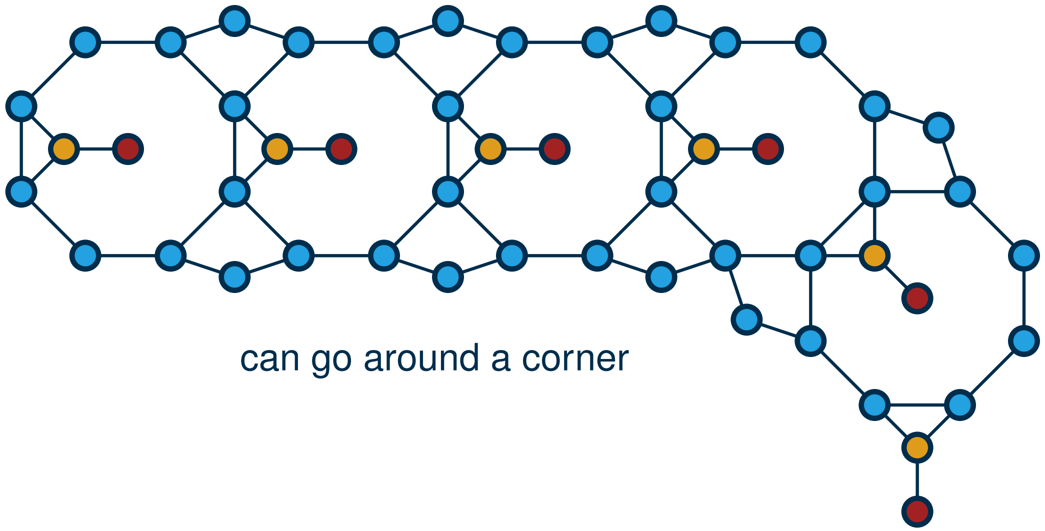
Gadgets We Need: variable, clause, transport

Variable Gadget

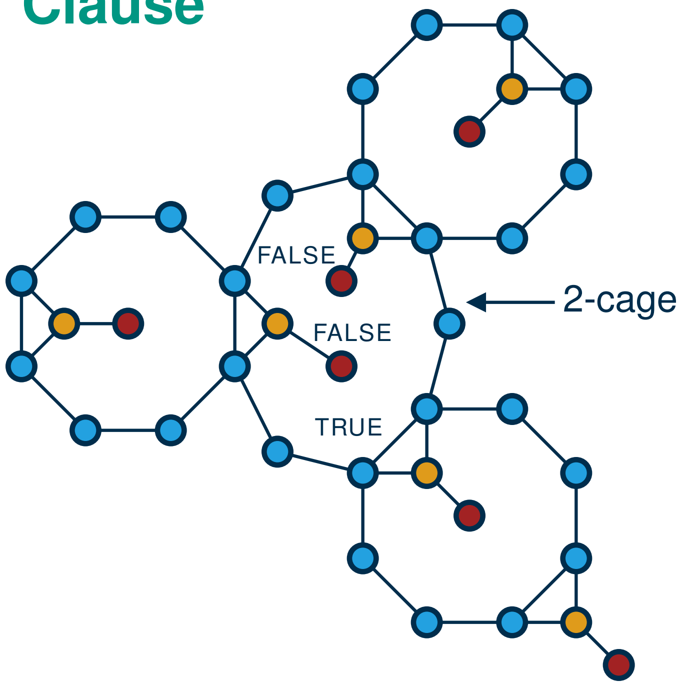




Transporting Information



Clause



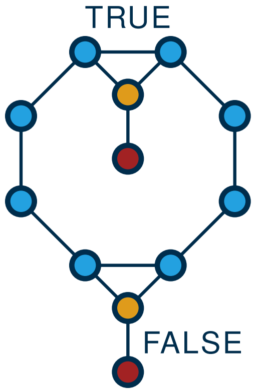
What Is Missing?

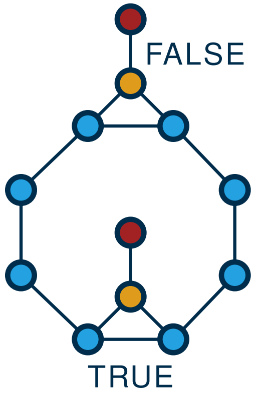
- we can transport the decision of a variable to only one clause (technically 2: one positive, one negative)
- variables are contained in multiple clauses

Gadgets

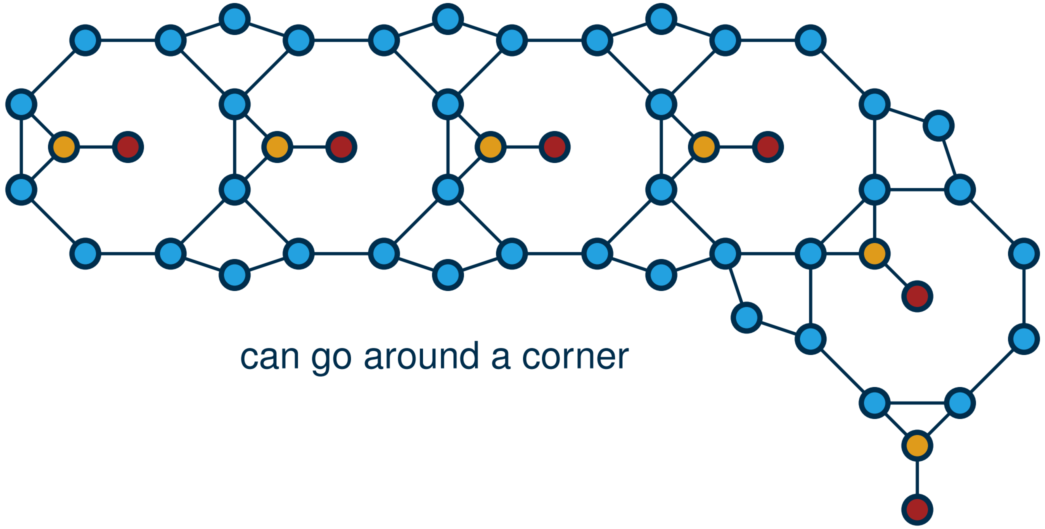
Gadgets We Need: variable, clause, transport

Variable Gadget

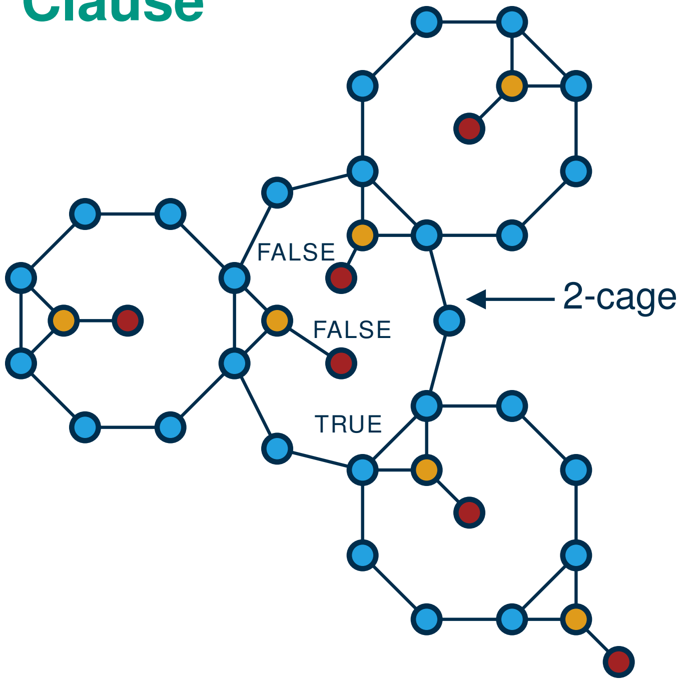




Transporting Information



Clause

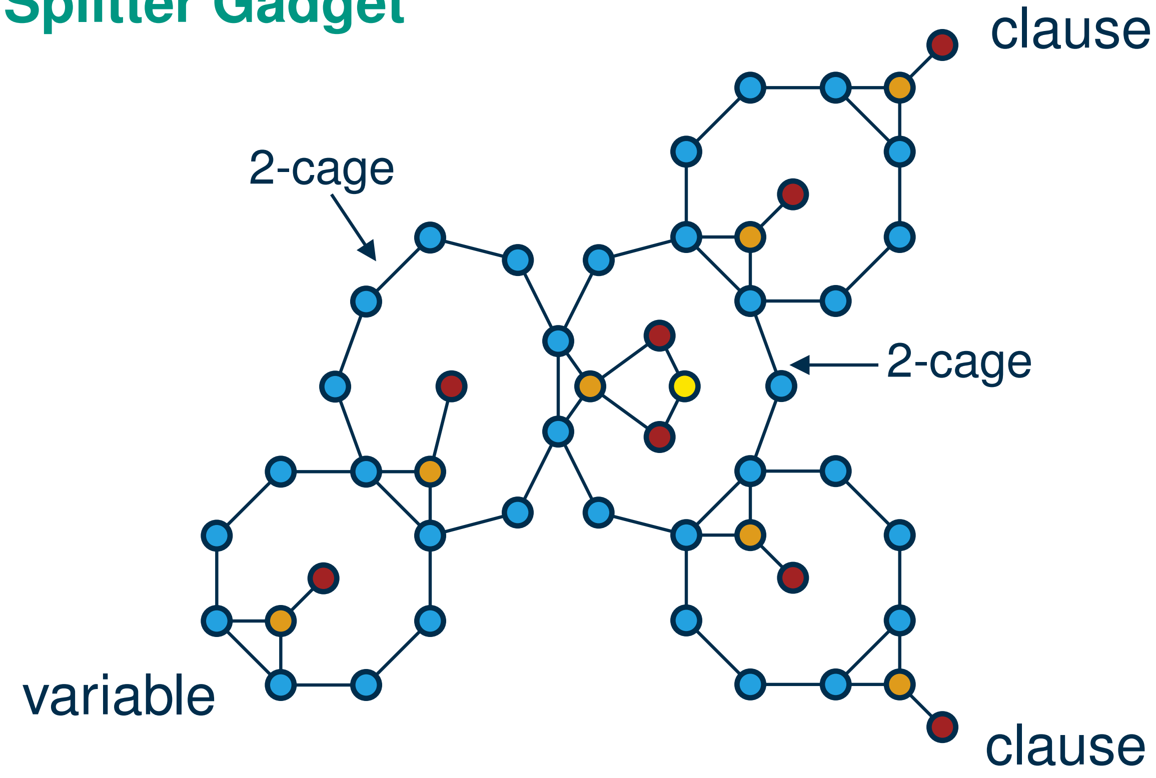


What Is Missing?

- we can transport the decision of a variable to only one clause
  (technically 2: one positive, one negative)
- variables are contained in multiple clauses
- we need a splitter gadget

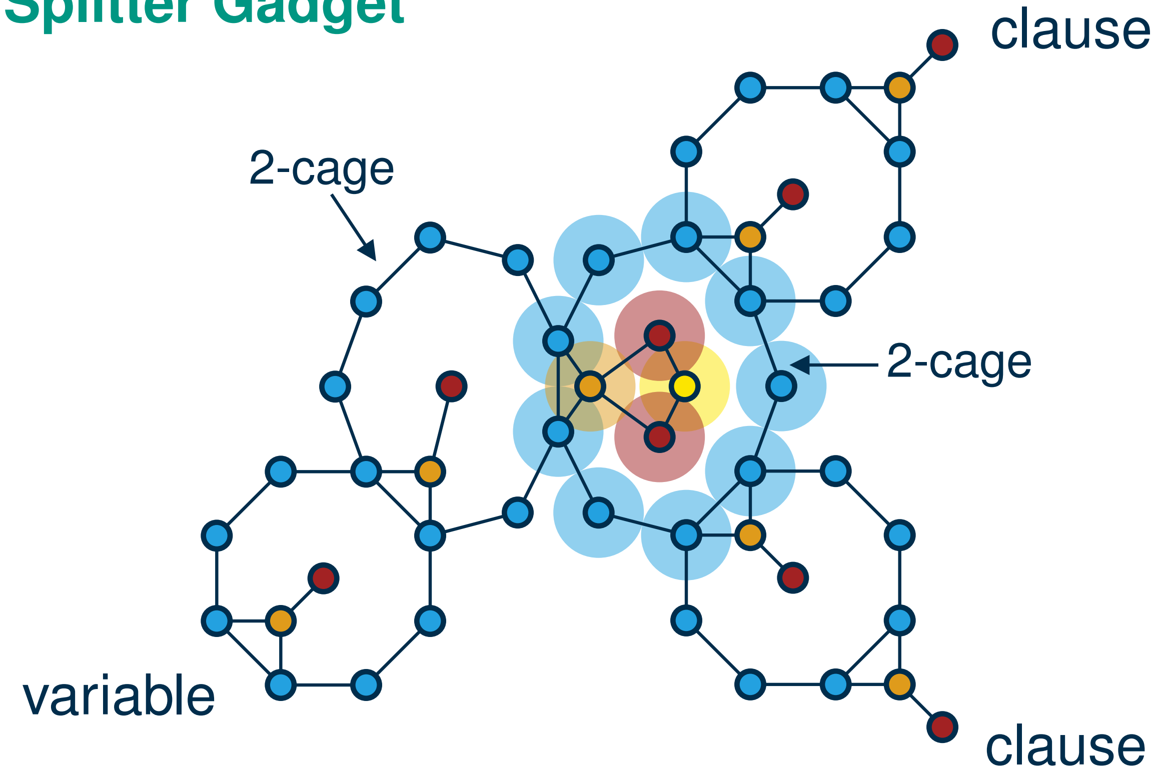
Multiplying Information

Splitter Gadget



Multiplying Information

Splitter Gadget

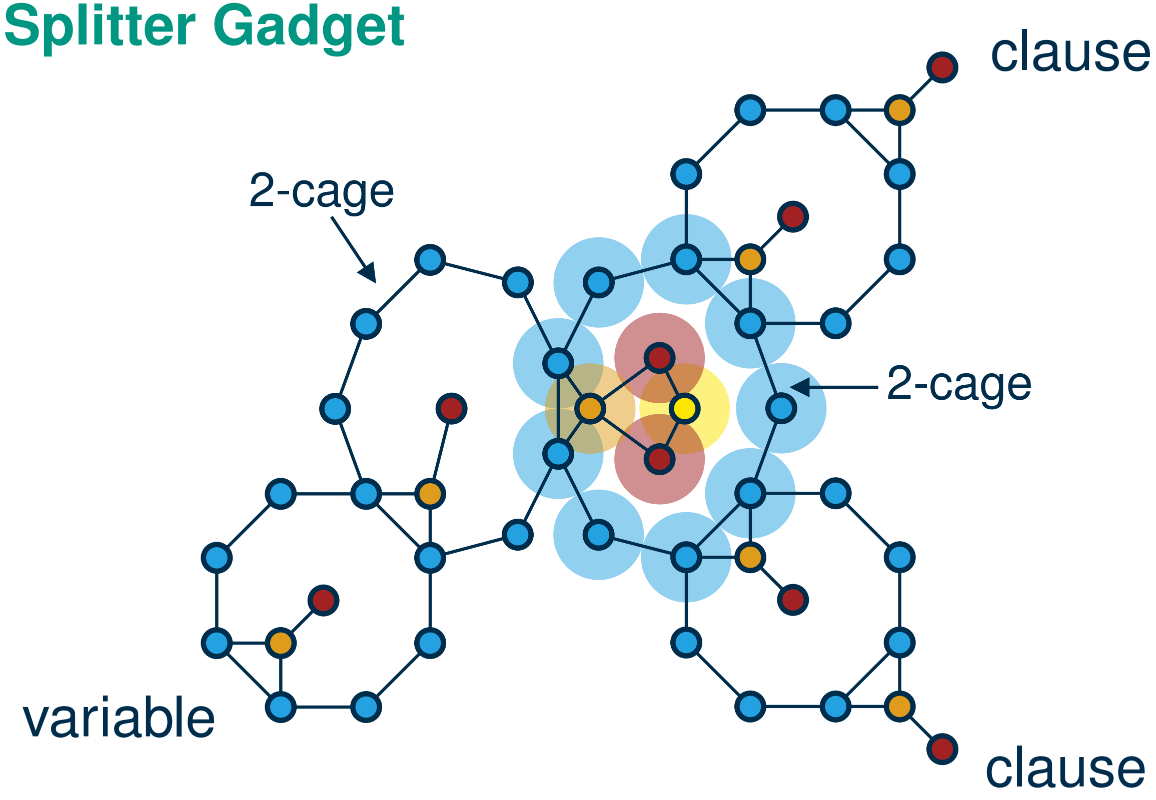


Note

- ○ forces both ● into the same 2-cage
- the 2-cage is realizable

Multiplying Information

Splitter Gadget

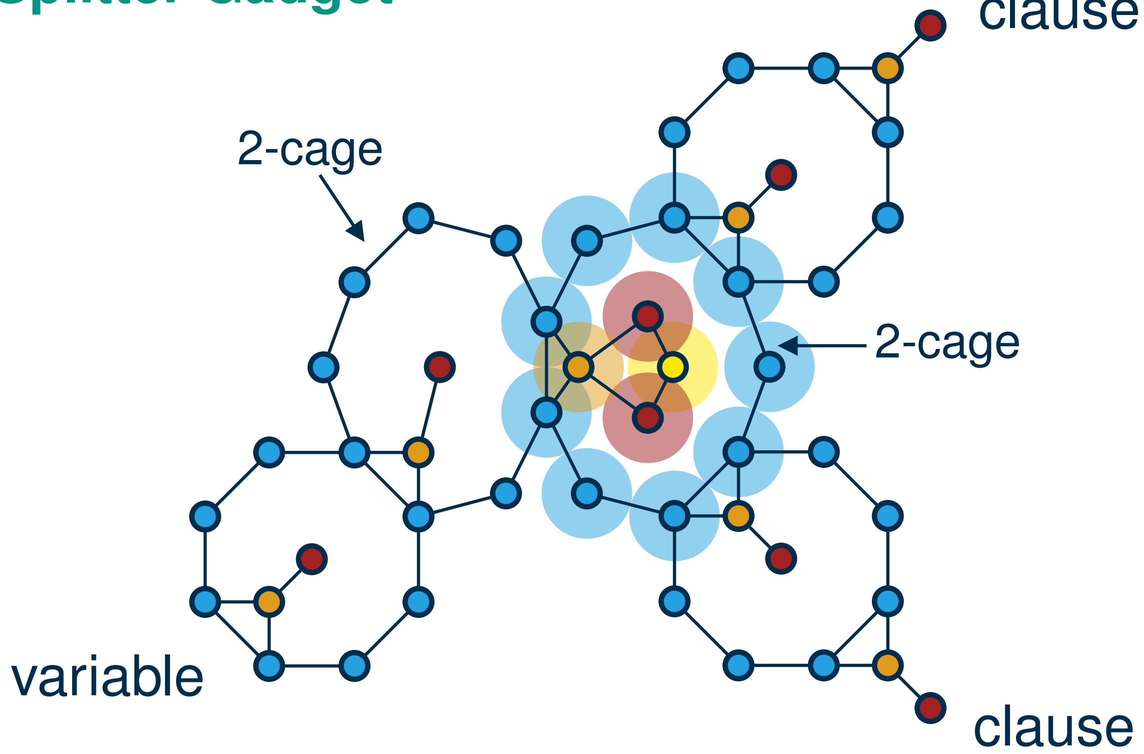


Note

- ○ forces both ● into the same 2-cage
- the 2-cage is realizable
- FALSE signal from the variable ⇒ FALSE signal to clauses (in every unit disk representation)

Multiplying Information

Splitter Gadget

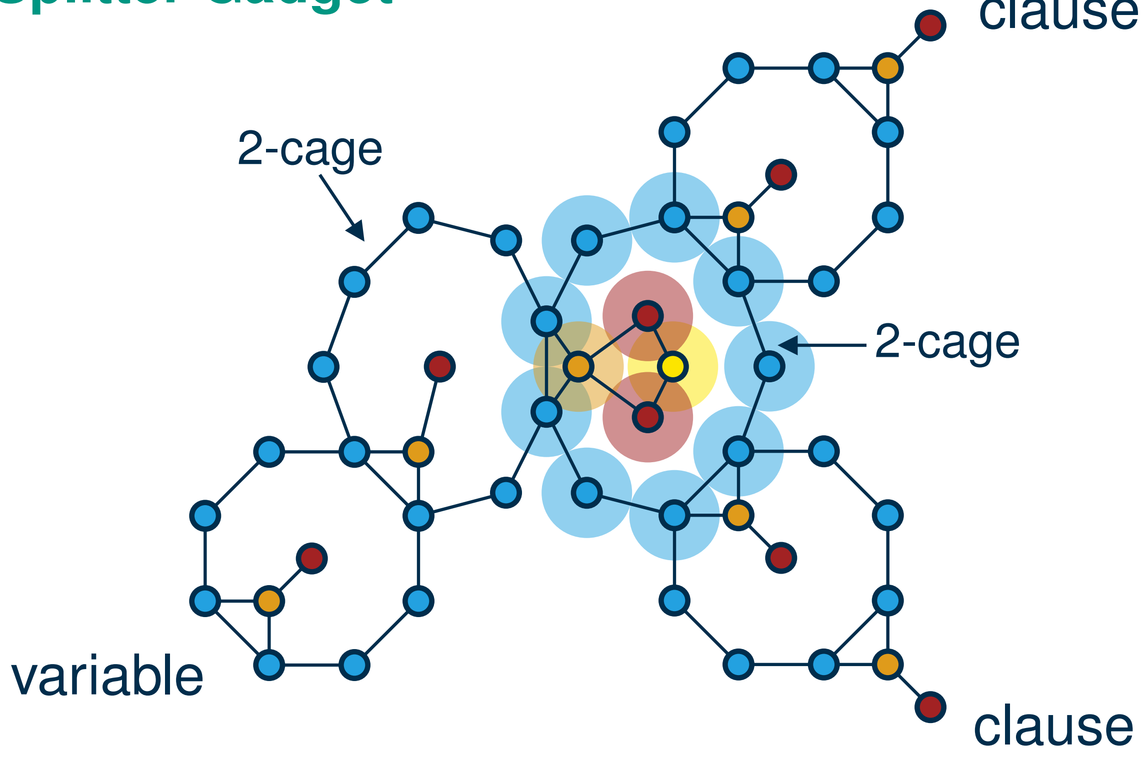


Note

- forces both into the same 2-cage
- the 2-cage is realizable
- FALSE signal from the variable ⇒ FALSE signal to clauses (in every unit disk representation)
- TRUE signal from variable ⇒ TRUE signals to clauses possible

Multiplying Information

Splitter Gadget



Note

- ● forces both ● into the same 2-cage
- the 2-cage is realizable
- FALSE signal from the variable ⇒ FALSE signal to clauses (in every unit disk representation)
- TRUE signal from variable ⇒ TRUE signals to clauses possible
- gadget does what it should (flip from TRUE to FALSE is ok)

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula
- resulting instance has polynomial size and can be computed in polynomial time

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula
- resulting instance has polynomial size and can be computed in polynomial time
- graph has unit disk representation $\Rightarrow$ 3-SAT formula satisfiable
- 3-SAT formula satisfiable $\Rightarrow$ graph has a unit disk representation

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula
- resulting instance has polynomial size and can be computed in polynomial time
- graph has unit disk representation $\Rightarrow$ 3-SAT formula satisfiable
- 3-SAT formula satisfiable $\Rightarrow$ graph has a unit disk representation

Is The Problem NP-Complete?

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula
- resulting instance has polynomial size and can be computed in polynomial time
- graph has unit disk representation $\Rightarrow$ 3-SAT formula satisfiable
- 3-SAT formula satisfiable $\Rightarrow$ graph has a unit disk representation

Is The Problem NP-Complete?

- probably not $\rightarrow$ what goes wrong?

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula
- resulting instance has polynomial size and can be computed in polynomial time
- graph has unit disk representation $\Rightarrow$ 3-SAT formula satisfiable
- 3-SAT formula satisfiable $\Rightarrow$ graph has a unit disk representation

Is The Problem NP-Complete?

- probably not $\rightarrow$ what goes wrong?
  - guess positions as certificate and check whether $uv \in E \Leftrightarrow \text{dist}(u, v) \leq 2$

Graphs That Are Hard To Recognize

Theorem
It is NP-hard to decide whether a given graph is a unit disk graph.

What Do We Need To Think About For The Proof?

- length and exact positioning of the transport gadgets: follows given drawing of 3-SAT formula
- resulting instance has polynomial size and can be computed in polynomial time
- graph has unit disk representation $\Rightarrow$ 3-SAT formula satisfiable
- 3-SAT formula satisfiable $\Rightarrow$ graph has a unit disk representation

Is The Problem NP-Complete?

- probably not $\rightarrow$ what goes wrong?
  - guess positions as certificate and check whether $uv \in E \Leftrightarrow \mathrm{dist}(u, v) \leq 2$
  - problem: this certificate sometimes needs to be exponentially large

(because we need double exponentially precise coordinates)

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \ldots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n \; F(X_1, \ldots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n \; F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n \; F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n\ F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \ldots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n \; F(X_1, \ldots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Recognizing Unit Disk Graphs

- problem lies in $\exists\mathbb{R}$

Why?

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n \; F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Recognizing Unit Disk Graphs

- problem lies in $\exists\mathbb{R}$
- it is actually $\exists\mathbb{R}$-complete

Why?

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \ldots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n\ F(X_1, \ldots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Recognizing Unit Disk Graphs

- problem lies in $\exists\mathbb{R}$
- it is actually $\exists\mathbb{R}$-complete

Why?

Relation To Other Classes

- $NP \subseteq \exists\mathbb{R}$

Why?

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n \; F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Recognizing Unit Disk Graphs

- problem lies in $\exists\mathbb{R}$
- it is actually $\exists\mathbb{R}$-complete

Why?

Relation To Other Classes

- $\text{NP} \subseteq \exists\mathbb{R}$
- $\exists\mathbb{R} \subseteq \text{PSPACE}$

Why?

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n\ F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Recognizing Unit Disk Graphs

- problem lies in $\exists\mathbb{R}$
- it is actually $\exists\mathbb{R}$-complete

Why?

Relation To Other Classes

- NP $\subseteq \exists\mathbb{R}$
- $\exists\mathbb{R} \subseteq$ PSPACE
- conjecture: NP $\subset \exists\mathbb{R} \subset$ PSPACE

Why?

Existential Theory Of The Reals

Problem: Existential Theory Of The Reals
Let $F(X_1, \dots, X_n)$ be a quantifier-free Boolean formula over (in-)equalities of real polynomials.
Is $\exists X_1 \cdots \exists X_n\ F(X_1, \dots, X_n)$ true?

(In logic, a *theory* is a set of statements. The *existential theory of the reals* is the set of all true statements of this form.)

The Complexity Class $\exists\mathbb{R}$

- $\Pi \in \exists\mathbb{R} \Leftrightarrow \Pi$ has a polynomial reduction to the existential theory of the reals (at most as hard)
- Π is $\exists\mathbb{R}$-hard $\Leftrightarrow$ all problems in $\exists\mathbb{R}$ have a polynomial reduction to Π (at least as hard)
- Π is $\exists\mathbb{R}$-complete $\Leftrightarrow \Pi \in \exists\mathbb{R}$ and Π $\exists\mathbb{R}$-hard (equally hard)

Recognizing Unit Disk Graphs

Why?

- problem lies in $\exists\mathbb{R}$
- it is actually $\exists\mathbb{R}$-complete
- we believe: recognizing unit disk graphs is strictly harder than every problem in NP

Relation To Other Classes

Why?

- NP $\subseteq \exists\mathbb{R}$
- $\exists\mathbb{R} \subseteq$ PSPACE
- conjecture: NP $\subset \exists\mathbb{R} \subset$ PSPACE

Wrap-Up

Seen Today

- problems: proportional symbol maps (cartography), recognition of unit disk graphs
- complexity class $\exists\mathbb{R}$
- for geometric problems, it is often unclear whether they lie in NP

Wrap-Up

Seen Today

- problems: proportional symbol maps (cartography), recognition of unit disk graphs
- complexity class $\exists\mathbb{R}$
- for geometric problems, it is often unclear whether they lie in NP
- reductions from SAT variants are often easy; you just need:
  - variable gadget
  - clause gadget
  - transportation gadget

Wrap-Up

Seen Today

- problems: proportional symbol maps (cartography), recognition of unit disk graphs
- complexity class $\exists\mathbb{R}$
- for geometric problems, it is often unclear whether they lie in NP
- reductions from SAT variants are often easy; you just need:
  - variable gadget
  - clause gadget
  - transportation gadget
  - maybe splitter gadget (if the variable gadget produces too few literals)

Wrap-Up

Seen Today

- problems: proportional symbol maps (cartography), recognition of unit disk graphs
- complexity class $\exists\mathbb{R}$
- for geometric problems, it is often unclear whether they lie in NP
- reductions from SAT variants are often easy; you just need:
  - variable gadget
  - clause gadget
  - transportation gadget
  - maybe splitter gadget (if the variable gadget produces too few literals)

What Else Is There?

- negation gadget (if you have a place where you need negative literals but can only get positive literals)

Wrap-Up

Seen Today

- problems: proportional symbol maps (cartography), recognition of unit disk graphs
- complexity class $\exists\mathbb{R}$
- for geometric problems, it is often unclear whether they lie in NP
- reductions from SAT variants are often easy; you just need:
  - variable gadget
  - clause gadget
  - transportation gadget
  - maybe splitter gadget (if the variable gadget produces too few literals)

What Else Is There?

- negation gadget (if you have a place where you need negative literals but can only get positive literals)
- crossing gadget (if you reduce from a non-planar SAT variant)

Literature

- **Algorithmic Aspects of Proportional Symbol Maps** (2009)
  Sergio Cabello, Herman Haverkort, Marc van Kreveld, Bettina Speckmann
  https://doi.org/10.1007/s00453-009-9281-8

- **Unit disk graph recognition is NP-hard** (1998)
  Heinz Breu, David G. Kirkpatrick
  https://doi.org/10.1016/S0925-7721(97)00014-X

- **Optimal Binary Space Partitions in the Plane** (2010)
  Mark de Berg, Amirali Khosravi
  https://doi.org/10.1007/978-3-642-14031-0_25
  (NP-hardness for planar monotone 3-SAT)

- **Sphere and Dot Product Representations of Graphs** (2012)
  Ross J. Kang, Tobias Müller
  https://doi.org/10.1007/s00454-012-9394-8
  ($\exists\mathbb{R}$-hardness for recognizing unit disk graphs)