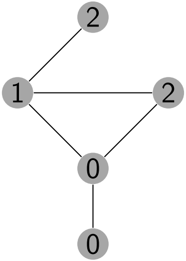
Computational Complexity Aspects of Graph Pebbling

Kevin G. Milans (milans@math.illinois.edu)
Joint with Bryan Clark

University of Illinois at Urbana-Champaign

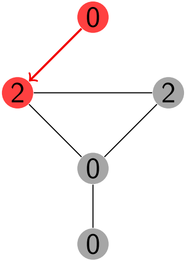
CanaDAM 2009
Montréal, Québec
26 May 2009

Graph Pebbling



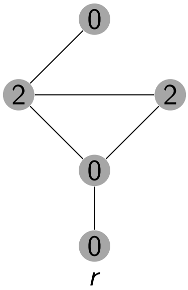
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor

Graph Pebbling



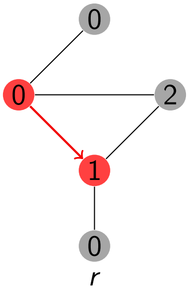
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor

Graph Pebbling



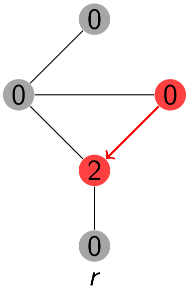
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?

Graph Pebbling



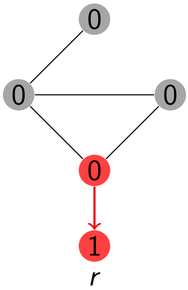
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?

Graph Pebbling



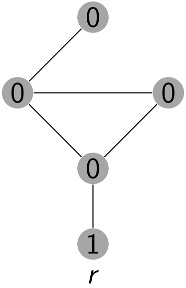
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?

Graph Pebbling



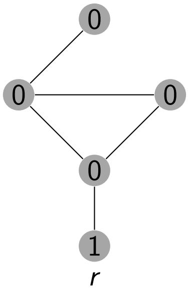
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?

Graph Pebbling



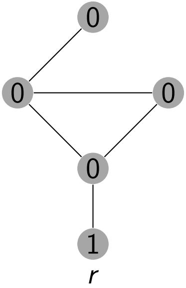
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?
- In this example: yes

Graph Pebbling



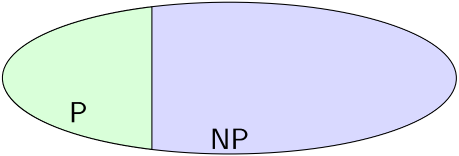
- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?
- In this example: yes
- Are there fast algorithms for this problem?

Graph Pebbling

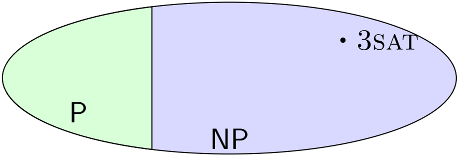


- A *pebbling move* removes two pebbles from a vertex and places one on a neighbor
- REACHABILITY: Given a graph G with pebbles and a target r, can we put a pebble on r?
- In this example: yes
- Are there fast algorithms for this problem?
- Probably not: many problems are special cases of REACHABILITY

Computational Complexity: NP-completeness

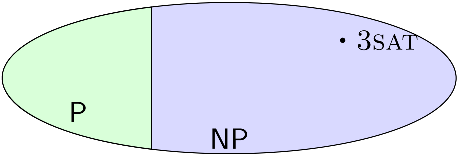


Computational Complexity: NP-completeness



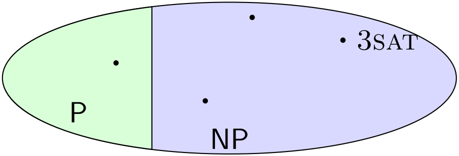
- Early 1970's: Stephen Cook and Leonid Levin independently made a remarkable discovery about a problem "3SAT":

Computational Complexity: NP-completeness



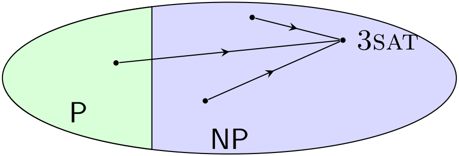
- Early 1970's: Stephen Cook and Leonid Levin independently made a remarkable discovery about a problem "3SAT":
- For each problem L in NP, it is possible to quickly convert instances of L to instances of 3SAT.

Computational Complexity: NP-completeness



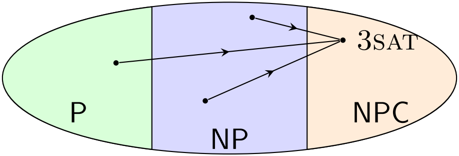
- Early 1970's: Stephen Cook and Leonid Levin independently made a remarkable discovery about a problem "3SAT":
- For each problem L in NP, it is possible to quickly convert instances of L to instances of 3SAT.

Computational Complexity: NP-completeness



- Early 1970's: Stephen Cook and Leonid Levin independently made a remarkable discovery about a problem "3SAT":
- For each problem L in NP, it is possible to quickly convert instances of L to instances of 3SAT.
- The procedure that translates instances of L to instances of 3SAT is called a *reduction*.

Computational Complexity: NP-completeness

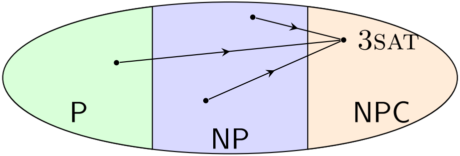


- Early 1970's: Stephen Cook and Leonid Levin independently made a remarkable discovery about a problem "3SAT":
- For each problem L in NP, it is possible to quickly convert instances of L to instances of 3SAT.
- The procedure that translates instances of L to instances of 3SAT is called a *reduction*.

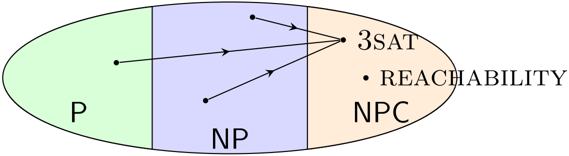
Theorem (Cook; Levin)

3SAT *is NP-complete.*

Complexity of REACHABILITY



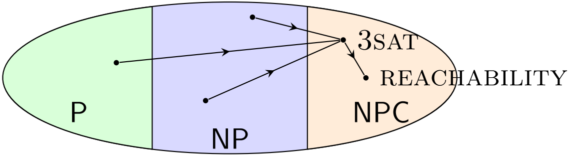
Complexity of REACHABILITY



Fact

REACHABILITY is in NP.

Complexity of REACHABILITY



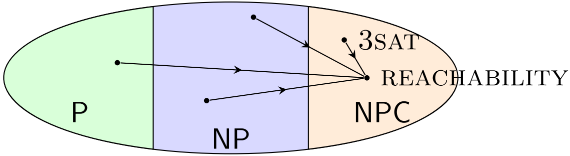
Fact

REACHABILITY is in NP.

Theorem

There is a polynomial time reduction from 3SAT *to* REACHABILITY*.*

Complexity of REACHABILITY



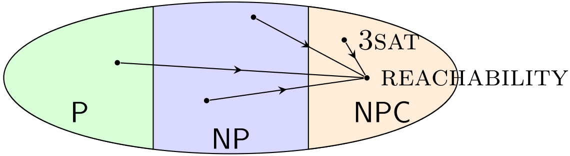
Fact

REACHABILITY is in NP.

Theorem

There is a polynomial time reduction from 3SAT *to* REACHABILITY.

Complexity of REACHABILITY



Fact

REACHABILITY is in NP.

Theorem

There is a polynomial time reduction from 3SAT *to* REACHABILITY.

Corollary (Hurlbert-Kierstead; Watson; Clark-Milans)

REACHABILITY *is NP-complete. If there is a polynomial time algorithm for* REACHABILITY, *then P=NP.*

3SAT

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"

3SAT

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"
- A boolean formula in 3CNF:

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

3SAT

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"
- A boolean formula in 3CNF:

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- 3SAT: Given a 3CNF formula ϕ, is ϕ satisfiable?

3SAT

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"
- A boolean formula in 3CNF:

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- 3SAT: Given a 3CNF formula ϕ, is ϕ satisfiable?

Definition

A 3CNF formula ϕ is *simple* if

3SAT

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"
- A boolean formula in 3CNF:

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- 3SAT: Given a 3CNF formula ϕ, is ϕ satisfiable?

Definition

A 3CNF formula ϕ is *simple* if

1. each variable appears at most twice in its positive form, and

$3\textsc{Sat}$

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"
- A boolean formula in 3CNF:

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- $3\textsc{Sat}$: Given a 3CNF formula ϕ, is ϕ satisfiable?

Definition

A 3CNF formula ϕ is *simple* if

1. each variable appears at most twice in its positive form, and
2. each variable appears at most once in its negative form.

3SAT

- $\wedge$ means "and", $\vee$ means "or", $\overline{x}$ means "not x"
- A boolean formula in 3CNF:

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- 3SAT: Given a 3CNF formula ϕ, is ϕ satisfiable?

Definition

A 3CNF formula ϕ is *simple* if

1. each variable appears at most twice in its positive form, and
2. each variable appears at most once in its negative form.

Proposition

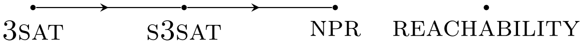
There is a polynomial time algorithm to convert a 3CNF formula to an equivalent simple 3CNF formula.

3SAT to REACHABILITY

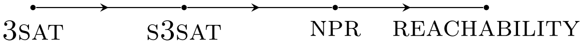
• 3SAT	• REACHABILITY

3SAT to REACHABILITY

3SAT to REACHABILITY



3SAT to REACHABILITY



3SAT to REACHABILITY

- Step 1. Straightforward.

3SAT to REACHABILITY

- Step 1. Straightforward.
- Step 2. NPR: Given a graph G with pebbles and a target r, can we put a pebble on r using each edge at most once?

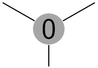
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

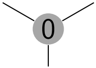
And Gadget



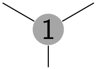
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

And Gadget



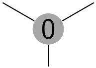
Or Gadget



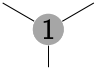
3SAT to REACHABILITY

$$\phi = (w \lor x) \land (w \lor \overline{x}) \land (\overline{w} \lor y \lor z) \land (x \lor \overline{y} \lor \overline{z})$$

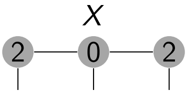
And Gadget



Or Gadget

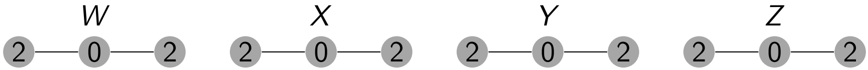


Variable Gadget



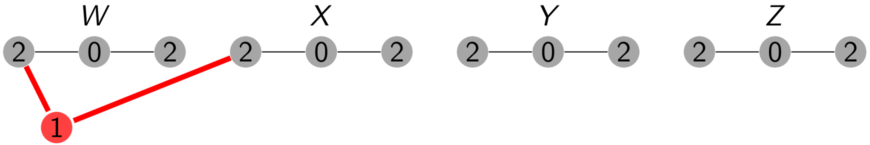
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



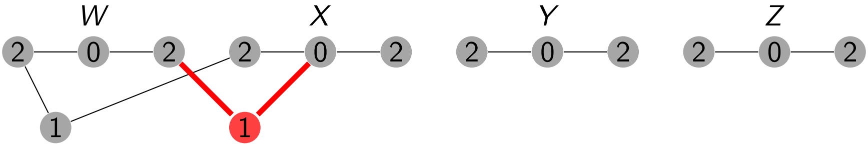
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



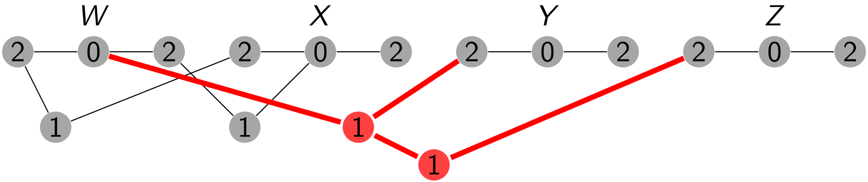
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



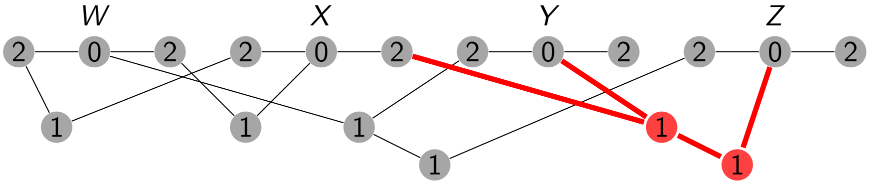
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



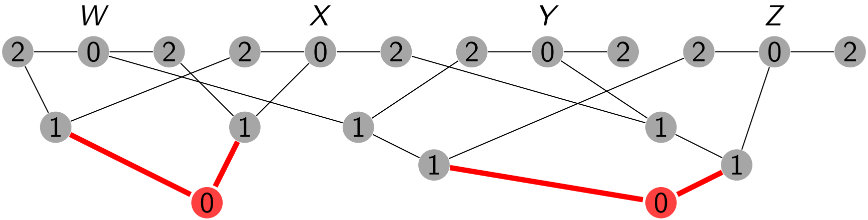
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



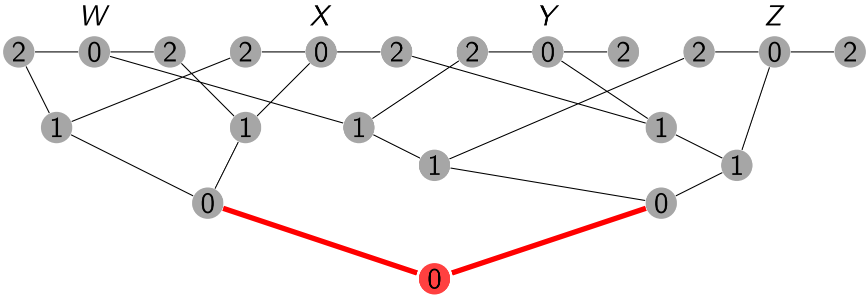
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



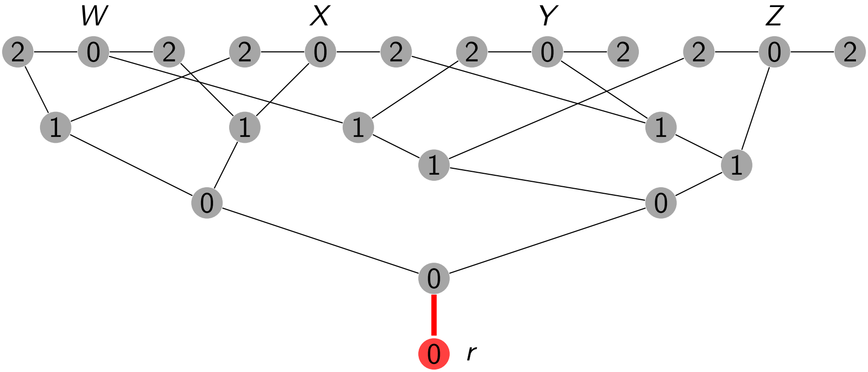
3SAT to REACHABILITY

$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



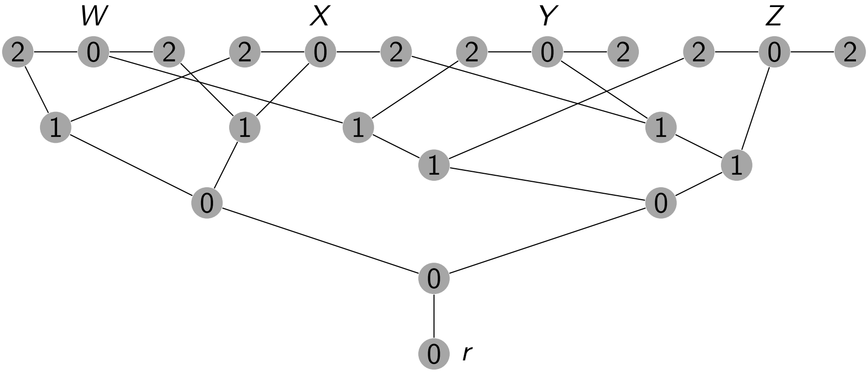
3SAT to REACHABILITY

$$\phi = (w \lor x) \land (w \lor \overline{x}) \land (\overline{w} \lor y \lor z) \land (x \lor \overline{y} \lor \overline{z})$$



3SAT to REACHABILITY

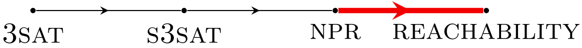
$$\phi = (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$



3SAT to REACHABILITY

- Step 1. Straightforward.
- Step 2. NPR: Given a graph G with pebbles and a target r, can we put a pebble on r using each edge at most once?

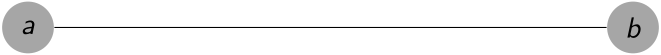
3SAT to REACHABILITY



- Step 1. Straightforward.
- Step 2. NPR: Given a graph G with pebbles and a target r, can we put a pebble on r using each edge at most once?
- Step 3. Replace each edge with a "one-use" path.

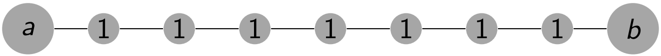
3SAT to REACHABILITY

- Step 1. Straightforward.
- Step 2. NPR: Given a graph G with pebbles and a target r, can we put a pebble on r using each edge at most once?
- Step 3. Replace each edge with a “one-use” path.

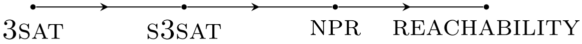


3SAT to REACHABILITY

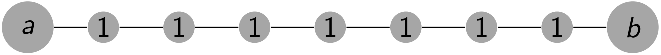
- Step 1. Straightforward.
- Step 2. NPR: Given a graph G with pebbles and a target r, can we put a pebble on r using each edge at most once?
- Step 3. Replace each edge with a “one-use” path.



3SAT to REACHABILITY



- Step 1. Straightforward.
- Step 2. NPR: Given a graph G with pebbles and a target r, can we put a pebble on r using each edge at most once?
- Step 3. Replace each edge with a "one-use" path.



Theorem

REACHABILITY *is NP-complete even for bipartite graphs with* $\Delta(G) \leq 3$ *and at most 2 pebbles on each vertex.*

Pebbling Number

- A distribution of pebbles is solvable if every vertex is reachable

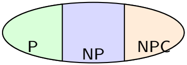
Pebbling Number

- A distribution of pebbles is solvable if every vertex is reachable
- $\pi(G)$: min k such that each dist. of k pebbles is solvable.

Pebbling Number

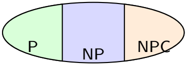
- A distribution of pebbles is solvable if every vertex is reachable
- $\pi(G)$: min k such that each dist. of k pebbles is solvable.
- PEBBLING-NUMBER: given G and k, is $\pi(G) \leq k$?

Beyond NP



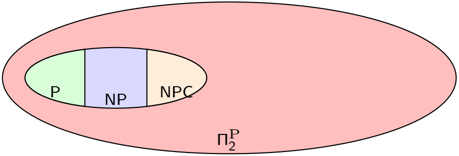
- We usually think of NP as containing “hard problems”.

Beyond NP



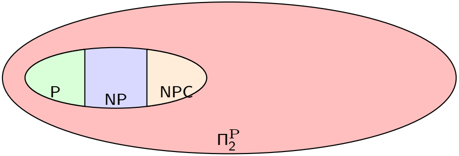
- We usually think of NP as containing “hard problems”.
- It’s all relative: some problems make NP look easy.

Beyond NP



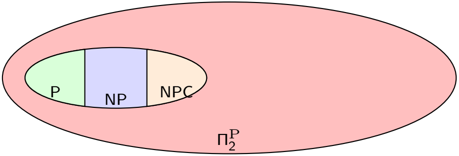
- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^P: class of problems containing NP.

Beyond NP



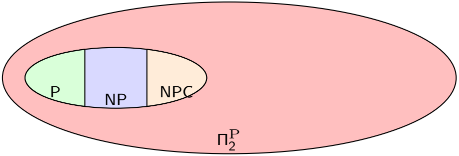
- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^P: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^P.

Beyond NP



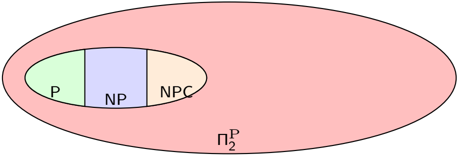
- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^{P}: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^{P}.
- How difficult can Π_2^{P} problems be?

Beyond NP



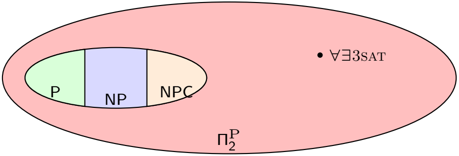
- We usually think of NP as containing “hard problems”.
- It’s all relative: some problems make NP look easy.
- Π_2^P: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^P.
- How difficult can Π_2^P problems be?
- (Probably) can’t solve them quickly, *even* if you have a magical device that solves problems in NP instantly.

Beyond NP



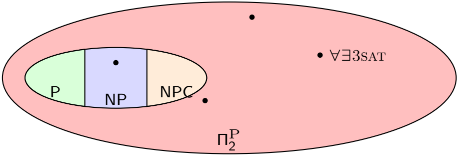
- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^P: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^P.
- How difficult can Π_2^P problems be?
- (Probably) can't solve them quickly, *even* if you have a magical device that solves problems in NP instantly.
- Just like P vs. NP, most believe that NP $\subsetneq \Pi_2^P$.

Beyond NP



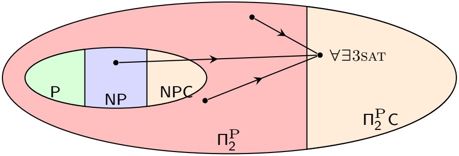
- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^{P}: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^{P}.
- How difficult can Π_2^{P} problems be?
- (Probably) can't solve them quickly, *even* if you have a magical device that solves problems in NP instantly.
- Just like P vs. NP, most believe that NP $\subsetneq \Pi_2^{\mathrm{P}}$.
- Analogous to 3SAT in NP: $\forall\exists 3\text{SAT}$ in Π_2^{P}.

Beyond NP



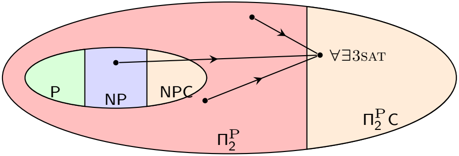
- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^{P}: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^{P}.
- How difficult can Π_2^{P} problems be?
- (Probably) can't solve them quickly, *even* if you have a magical device that solves problems in NP instantly.
- Just like P vs. NP, most believe that NP $\subsetneq \Pi_2^{\mathrm{P}}$.
- Analogous to $3\textsc{sat}$ in NP: $\forall\exists 3\textsc{sat}$ in Π_2^{P}.

Beyond NP



- We usually think of NP as containing "hard problems".
- It's all relative: some problems make NP look easy.
- Π_2^P: class of problems containing NP.
- Roughly: P is to NP as NP is to Π_2^P.
- How difficult can Π_2^P problems be?
- (Probably) can't solve them quickly, *even* if you have a magical device that solves problems in NP instantly.
- Just like P vs. NP, most believe that NP $\subsetneq \Pi_2^P$.
- Analogous to 3SAT in NP: ∀∃3SAT in Π_2^P.

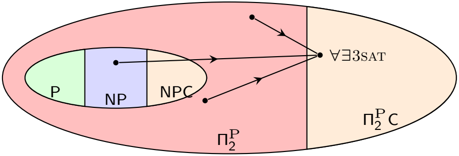
Beyond NP



- 3SAT example:

$$(w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

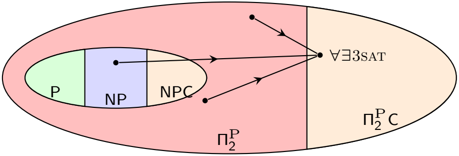
Beyond NP



- 3SAT example:

$$\exists w \exists y \exists x \exists z \quad (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

Beyond NP



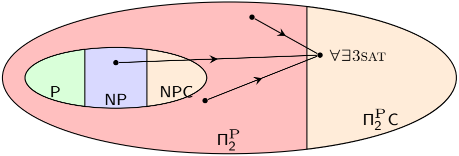
- 3SAT example:

$$\exists w \exists y \exists x \exists z \quad (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- $\forall\exists 3\text{SAT}$ example:

$$\forall w \forall y \exists x \exists z \quad (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

Beyond NP



- 3SAT example:

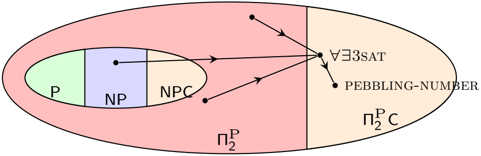
$$\exists w \exists y \exists x \exists z \quad (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- $\forall\exists 3\text{SAT}$ example:

$$\color{red}{\forall w \forall y} \color{black}{\exists x \exists z} \quad (w \vee x) \wedge (w \vee \overline{x}) \wedge (\overline{w} \vee y \vee z) \wedge (x \vee \overline{y} \vee \overline{z})$$

- $\forall\exists 3\text{SAT}$ example is a “no” instance: if w is false, first two clauses are unsatisfiable.

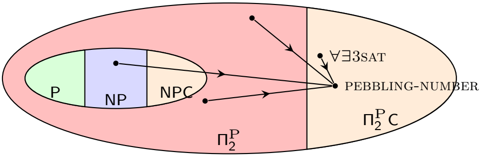
Beyond NP



Theorem

There is a polynomial time reduction from ∀∃3SAT *to* PEBBLING-NUMBER.

Beyond NP



Theorem

There is a polynomial time reduction from ∀∃3SAT *to* PEBBLING-NUMBER.

Corollary

PEBBLING-NUMBER *is* Π_2^P*-complete.*

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.
- $\gamma_f(G)$: min. k such that every dist. of k pebbles covers f.

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.
- $\gamma_f(G)$: min. k such that every dist. of k pebbles covers f.
- $\gamma(G)$: special case of $\gamma_f(G)$ where f is the unit distribution.

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.
- $\gamma_f(G)$: min. k such that every dist. of k pebbles covers f.
- $\gamma(G)$: special case of $\gamma_f(G)$ where f is the unit distribution.

Parameter	Question	Complexity
Cover Pebbling Number	Is $\gamma(G) \leq k$?	polynomial time[1]
Optimal Pebbling Number	Is $\widehat{\pi}(G) \leq k$?	NP-complete
Pebbling Number	Is $\pi(G) \leq k$?	Π_2^{P}-complete

[1] Vuong–Wyckoff, Sjöstrand

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.
- $\gamma_f(G)$: min. k such that every dist. of k pebbles covers f.
- $\gamma(G)$: special case of $\gamma_f(G)$ where f is the unit distribution.

Parameter	Question	Complexity
Cover Pebbling Number	Is $\gamma(G) \leq k$?	polynomial time[1]
Optimal Pebbling Number	Is $\widehat{\pi}(G) \leq k$?	NP-complete
Pebbling Number	Is $\pi(G) \leq k$?	Π_2^{P}-complete

Open Problems

- Recall: always $\pi(G) \geq |V(G)|$.

[1] Vuong–Wyckoff, Sjöstrand

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.
- $\gamma_f(G)$: min. k such that every dist. of k pebbles covers f.
- $\gamma(G)$: special case of $\gamma_f(G)$ where f is the unit distribution.

Parameter	Question	Complexity
Cover Pebbling Number	Is $\gamma(G) \leq k$?	polynomial time[1]
Optimal Pebbling Number	Is $\widehat{\pi}(G) \leq k$?	NP-complete
Pebbling Number	Is $\pi(G) \leq k$?	Π_2^{P}-complete

Open Problems

- Recall: always $\pi(G) \geq |V(G)|$.
- What is the complexity of deciding whether $\pi(G) = |V(G)|$?

[1] Vuong–Wyckoff, Sjöstrand

Complexity of Pebbling Number Variants

- $\widehat{\pi}(G)$: min. k such that there is a solvable dist. of size k.
- A dist. of pebbles covers a function $f : V(G) \to \mathbb{N}$ if there is a sequence of pebbling moves after which: $\forall v$ at least $f(v)$ pebbles on v.
- $\gamma_f(G)$: min. k such that every dist. of k pebbles covers f.
- $\gamma(G)$: special case of $\gamma_f(G)$ where f is the unit distribution.

Parameter	Question	Complexity
Cover Pebbling Number	Is $\gamma(G) \leq k$?	polynomial time[1]
Optimal Pebbling Number	Is $\widehat{\pi}(G) \leq k$?	NP-complete
Pebbling Number	Is $\pi(G) \leq k$?	Π_2^{P}-complete

Open Problems

- Recall: always $\pi(G) \geq |V(G)|$.
- What is the complexity of deciding whether $\pi(G) = |V(G)|$?
- Approximation algorithms for $\pi(G)$.

[1] Vuong–Wyckoff, Sjöstrand