Fö. 6: Komplexitetsteori och NP-fullständighet I.

Fö. 7: Komplexitetsteori och NP-fullständighet II.

Fö. 8: Algoritmer för NP-fullständiga problem.

Algoritmer för NP-fullständiga problem

Fö. 8

- Algoritm 1 för 3SAT.
- Algoritm för Oberoende Mängd.
- Algoritm för 3-Färgning.
- Algoritm 2 för 3SAT.

Algoritmer för NP-fullständiga problem

Syfte: Demonstrera en klass (intressanta) problem som (troligen) inte kan lösas i polynomisk tid.

Algoritmer för NP-fullständiga problem

Syfte: Demonstrera en klass (intressanta) problem som (troligen) inte kan lösas i polynomisk tid.

Alla problem i **NP** kan lösas i $2^{\mathrm{poly}(n)}$ tid.

Algoritmer för NP-fullständiga problem

Syfte: Demonstrera en klass (intressanta) problem som (troligen) inte kan lösas i polynomisk tid.

Alla problem i **NP** kan lösas i $2^{\mathrm{poly}(n)}$ tid.

Många kan lösas i $2^{c \cdot n}$ tid.

Algoritmer för NP-fullständiga problem

Syfte: Demonstrera en klass (intressanta) problem som (troligen) inte kan lösas i polynomisk tid.

Alla problem i **NP** kan lösas i $2^{\mathrm{poly}(n)}$ tid.

Många kan lösas i $2^{c \cdot n}$ tid.

Att minska c är en bra idé.

Algoritmer för NP-fullständiga problem

Generell algoritm för problem i **NP**

Algoritmer för NP-fullständiga problem

Generell algoritm för problem i **NP**

Låt X vara ett godtyckligt problem i **NP**.

Algoritmer för NP-fullständiga problem

Generell algoritm för problem i **NP**

Låt X vara ett godtyckligt problem i **NP**.

Låt A_X vara (den polynomiska) verifikationsalgoritmen och n^k en övre gräns på lösningslängd för problem av längd n.

algoritm $B_X(I)$
räkna upp alla strängar av längd $||I||^k$
om A_X accepterar någon av dessa svara "ja" annars "nej"

Algoritmer för NP-fullständiga problem

Problem	Trivial alg.	Förbättrad alg.
3SAT 1	$2^n \cdot \mathrm{poly}(\|\|I\|\|)$	$1.84^n \cdot \mathrm{poly}(\|\|I\|\|)$
Oberoende Mängd	$2^n \cdot \mathrm{poly}(\|\|I\|\|)$	$1.38^n \cdot \mathrm{poly}(\|\|I\|\|)$
3-Färgning	$3^n \cdot \mathrm{poly}(\|\|I\|\|)$	$1.73^n \cdot \mathrm{poly}(\|\|I\|\|)$
3SAT 2	$2^n \cdot \mathrm{poly}(\|\|I\|\|)$	$1.73^n \cdot \mathrm{poly}(\|\|I\|\|)$

Algoritm 1 för 3SAT

3SAT 1

En klausul $(p \vee q \vee r)$ kan satisfieras på 7 olika sätt.

3SAT 1

En klausul $(p \vee q \vee r)$ kan satisfieras på 7 olika sätt.

p	q	r
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

3SAT 1

En klausul $(p \vee q)$ kan satisfieras på 3 olika sätt.

3SAT 1

En klausul $(p \vee q)$ kan satisfieras på 3 olika sätt.

p	q
0	1
1	0
1	1

3SAT 1

En klausul (p) kan endast satisfieras på 1 sätt.

3SAT 1

En klausul (p) kan endast satisfieras på 1 sätt.

Sådana klausuler kan "propageras" bort.

$$(p) \wedge (q) \wedge (p \vee \neg q \vee r) \wedge (q \vee r \vee \neg s) \wedge (p \vee s)$$

$$\Rightarrow$$

$$(q) \wedge \underbrace{(p \vee \neg q \vee r)}_{\text{satisfierad av } p} \wedge (q \vee r \vee \neg s) \wedge \underbrace{(p \vee s)}_{\text{satisfierad av } p}$$

$$\Rightarrow$$

$$\underbrace{(q \vee r \vee \neg s)}_{\text{satisfierad av } q}$$

$$\Rightarrow$$

Hela formeln satisfierad.

3SAT 1

$(p) \wedge (q) \wedge (\neg p \vee \neg q)$

$\Rightarrow$

$(q) \wedge \underbrace{(\neg p \vee \neg q)}_{\neg p \text{ kan tas bort}}$

$\Rightarrow$

$\underbrace{(\neg q)}_{\neg q \text{ kan tas bort}}$

$\Rightarrow$

$()$

Tomma klausuler kan inte satisfieras.

3SAT 1

Indata 3SAT-formel F

Propagera bort alla enhetsklausuler

Om en tom klausul uppstår så svara "nej"

Om formeln blir tom så svara "ja"

Välj en klausul $(p \vee q \vee r)$ i formeln F.

Gör 7 rekursiva anrop på formlerna $F \wedge (p) \wedge (\neg q) \wedge (\neg r)$, $F \wedge (\neg p) \wedge (q) \wedge (r)$ etc.

3SAT 1

Tidskomplexitet i antal variabler n och $||F||$.

$T(1) = 1$

klausul längd 1: $T(n) = T(n-1) + \mathrm{poly}(||F||)$

klausul längd 2: $T(n) = 3T(n-2) + \mathrm{poly}(||F||)$

klausul längd 3: $T(n) = 7T(n-3) + \mathrm{poly}(||F||)$

3SAT 1

Från Fö 1.

$$T(n) = \sum_{i=1}^{k} T(n - r_i) + poly(n)$$

Låt $f(x) = 1 - \sum_{i=1}^{k} x^{-r_i}$.

Hitta största reella nollstället c till f.

$T(n) \in O(c^n)$.

3SAT 1

Observation. Rekursiva ekvationer av typen

$$T(1) = 1$$
$$T(n) = aT(n-b) + \mathrm{poly}(||I||)$$

har lösningen

$$T(n) \in a^{n/b} \cdot \mathrm{poly}(||I||).$$

3SAT 1

Tidskomplexitet i antal variabler n och $||F||$.

$T(1) = 1$

klausul längd 1: $T(n) = \text{poly}(||F||)$

klausul längd 2: $T(n) = 3^{n/2} + \text{poly}(||F||)$

klausul längd 3: $T(n) = 7^{n/3} + \text{poly}(||F||)$

Värsta fall: $T(n) = 7^{n/3} \cdot \text{poly}(||F||) \leq 1.92^n \cdot \text{poly}(||F||)$.

3SAT 1

Algoritmen kan enkelt förbättras.

3SAT 1

Algoritmen kan enkelt förbättras.

$(p \vee q \vee r)$ "täcks" av tre fall.

1. (p)
2. $(\neg p) \wedge (q)$
3. $(\neg p) \wedge (\neg q) \wedge (r)$.

3SAT 1

Tidskomplexitet i antal variabler n och $||F||$.

$T(1) = 1$

klausul längd 1: $T(n) = T(n-1) + \mathrm{poly}(||F||)$

klausul längd 2: $T(n) = 3T(n-2) + \mathrm{poly}(||F||)$

klausul längd 3:
$T(n) = T(n-1) + T(n-2) + T(n-3) + \mathrm{poly}(||F||)$

3SAT 1

Tidskomplexitet i antal variabler n och $||F||$.

$T(1) = 1$

klausul längd 1: $T(n) = \text{poly}(||F||)$

klausul längd 2: $T(n) = 1.73^n + \text{poly}(||F||)$

klausul längd 3: $T(n) = 1.84^n + \text{poly}(||F||)$

Värsta fall: $T(n) \leq 1.84^n \cdot \text{poly}(||F||)$.

Algoritm för Oberoende Mängd

Oberoende Mängd

Försök identifiera polynomiskt lösbara fall.

Oberoende Mängd

Försök identifiera polynomiskt lösbara fall.

Vad händer om graden är begränsad?

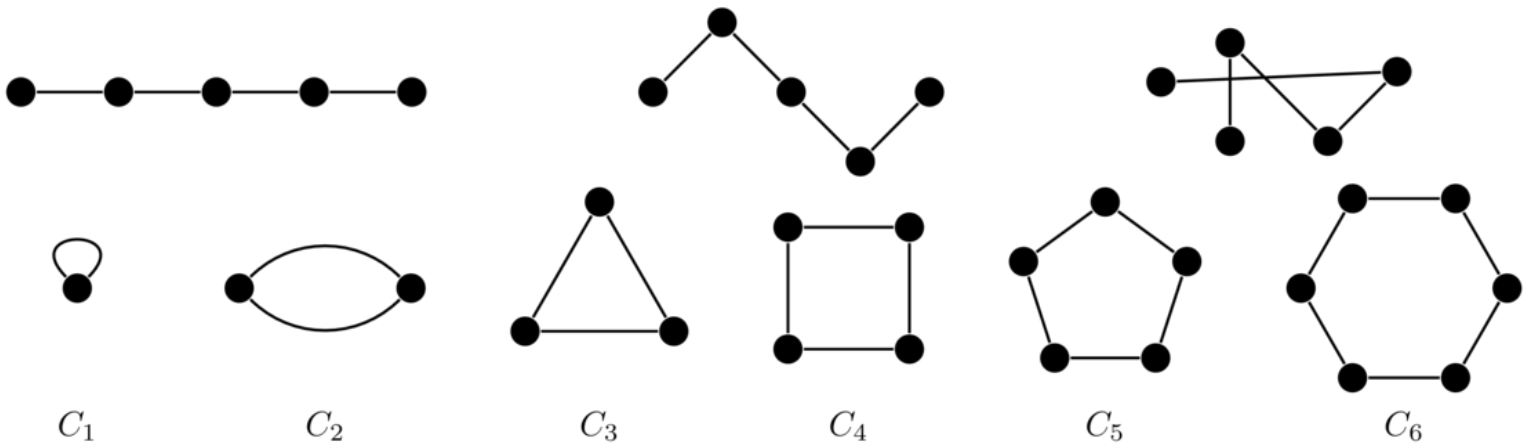
Oberoende Mängd

Försök identifiera polynomiskt lösbara fall.

Vad händer om graden är begränsad?

Hur ser en graf med grad 2 ut?

Oberoende Mängd



Oberoende Graf är lösbar i polynomisk tid under denna begränsning.

Oberoende Mängd

algoritm $A((V, E))$

om G har grad ≤ 2 så beräkna korrekt svar med den polynomiska algoritmen

välj godtyckligt en nod $v \in V$ med grad minst 3

$a = A(G - \{v\})$

$b = 1 + A(G - (\{v\} \cup grannar(v)))$

returnera $\max(a, b)$

Oberoende Mängd

Tidskomplexitet mätt i antalet noder n och $||G||$.

$$T(1) = 1$$

$$T(n) = T(n-1) + T(n-4) + poly(||G||)$$

Oberoende Mängd

Tidskomplexitet mätt i antalet noder n och $||G||$.

$$T(1) = 1$$

$$T(n) = T(n-1) + T(n-4) + poly(||G||)$$

$$T(n) \leq 1.38^n \cdot \mathrm{poly}(||G||).$$

Algoritm för 3-Färgning

3-Färgning

Enkel algoritm för 3-Färgning.

Räkna upp alla tilldelningar av "färgerna" R och G/B till noderna i grafen.

Om noderna som färgats med R inte bildar en oberoende mängd så svara "nej".

Om delgrafen med de noder som färgats G/B inte är 2-färgbar så svar "nej".

Svara "ja".

3-Färgning

Enkel algoritm för 3-Färgning.

Räkna upp alla tilldelningar av "färgerna" R och G/B till noderna i grafen.

Om noderna som färgats med R inte bildar en oberoende mängd så svara "nej".

Om delgrafen med de noder som färgats G/B inte är 2-färgbar så svar "nej".

Svara "ja".

Tidskomplexitet: $O(2^n \cdot \mathrm{poly}(||G||))$ där n är antalet noder.

3-Färgning

Matchning i en graf $G = (V, E)$.

Delmängd bågar $E' \subseteq E$ så att varje nod ingår i högst en av dessa bågar.

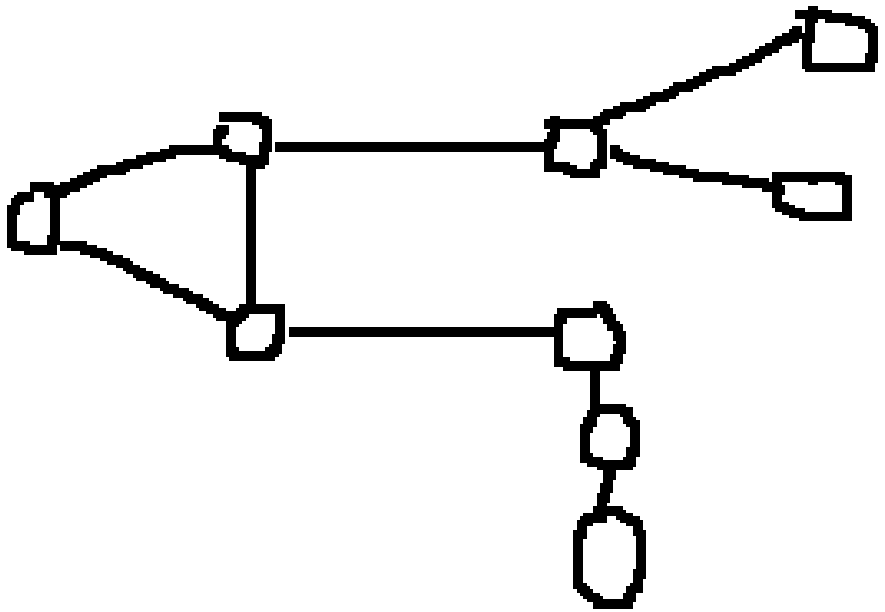
3-Färgning

Matchning i en graf $G = (V, E)$.

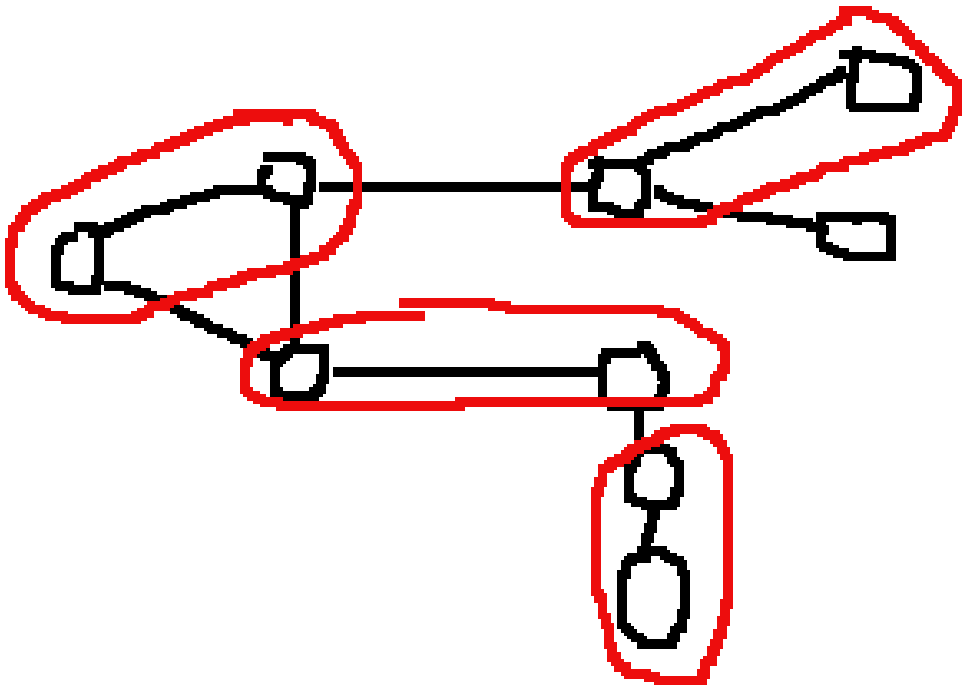
Delmängd bågar $E' \subseteq E$ så att varje nod ingår i högst en av dessa bågar.

Låt E' vara en matchning. Om en nod ingår i någon av dessa bågar kallas den *mättad* och, i annat fall, *omättad*.

3-Färgning



3-Färgning



3-Färgning

Maximal matchning i en graf $G = (V, E)$.

Matchning $E' \subseteq E$ sådan att man inte kan lägga till någon ytterligare båge och fortfarande ha en matchning.

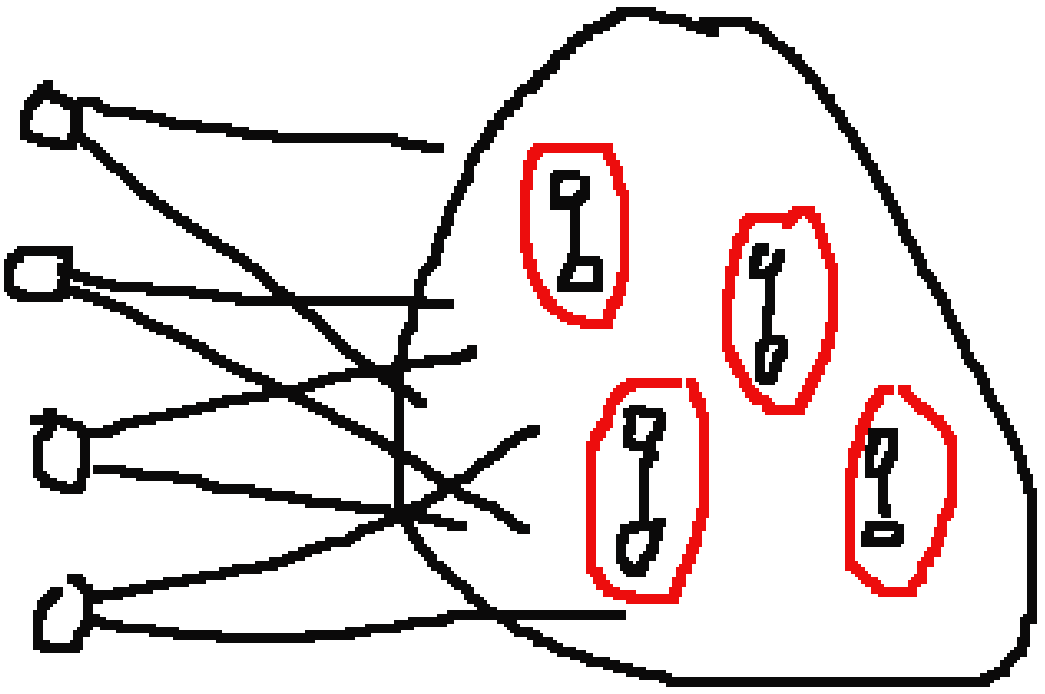
En maximal matchning kan hittas i polynomisk tid med en enkel girig algoritm.

3-Färgning

Observation:

Låt $G = (V, E)$ vara en graf med en matchning $E' \subseteq E$. De omättade noderna bildar en oberoende mängd.

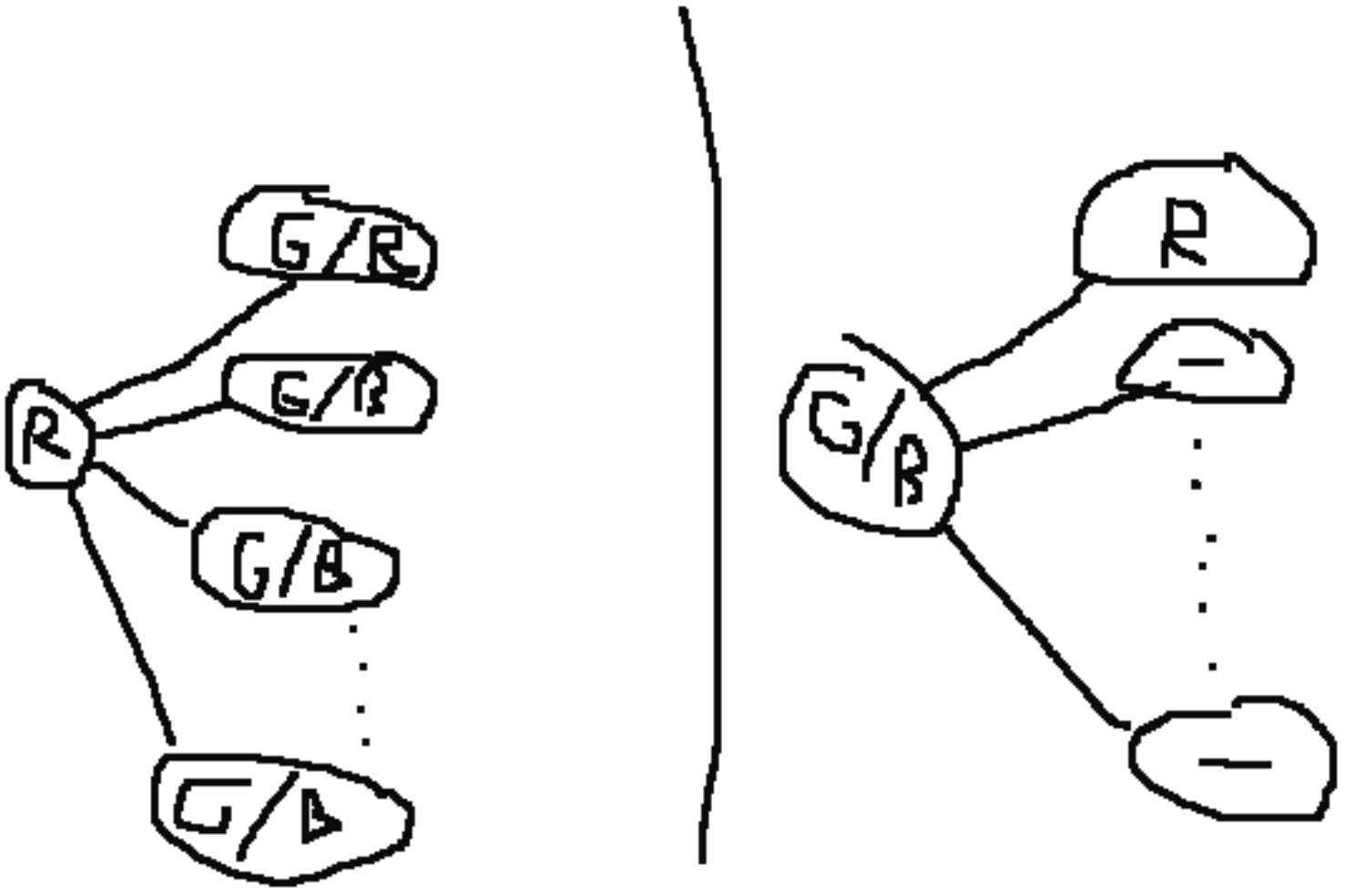
3-Färgning



3-Färgning

Antag att de mättade noderna har tilldelats "färgerna" R och G/B.
Hur färga de omättade noderna?

3-Färgning



3-Färgning

Tag fram en maximal matchning av G

Räkna upp alla tänkbara tilldelningar av de matchade paren

1. R — G/B
2. G/B — R
3. G/B — G/B

För varje sådan tilldelning, färga de omättade noderna enligt tidigare.

Kontrollera om färgningen är ok.

3-Färgning

Tidskomplexitet.

I värsta fallet är alla noder mättade. Vi har tre alternativ för varje matchad båge. Detta ger $3^{n/2} \cdot \mathrm{poly}(||G||) \leq 1.733^n \cdot \mathrm{poly}(||G||)$.

Algoritm 2 för 3SAT

3SAT 2

F — 3SAT-formel

a — en variabeltilldelning

d — sökdjup

3SAT 2

algoritm $X(F, a, d)$

om a satisfierar F så svara "ja"

om $d = 0$ så svara "nej"

välj en klausul C i F som inte satisfieras av a, exempelvis $C = (p \vee q \vee r)$

gör tre rekursiva anrop

1. $X(F, a[p = 1], d - 1)$
2. $X(F, a[q = 1], d - 1)$
3. $X(F, a[r = 1], d - 1)$

om något anrop ger svaret "ja" så svara "ja" och annars "nej"

3SAT 2

Antag att F innehåller n variabler.

Då kollar $X(F, (0, \ldots, 0), n)$ om F är satisfierbar eller inte.

3SAT 2

Antag att F innehåller n variabler.

Då kollar $X(F, (0, \ldots, 0), n)$ om F är satisfierbar eller inte.

Tidskomplexitet: $3^n \cdot \mathrm{poly}(||F||)$.

3SAT 2

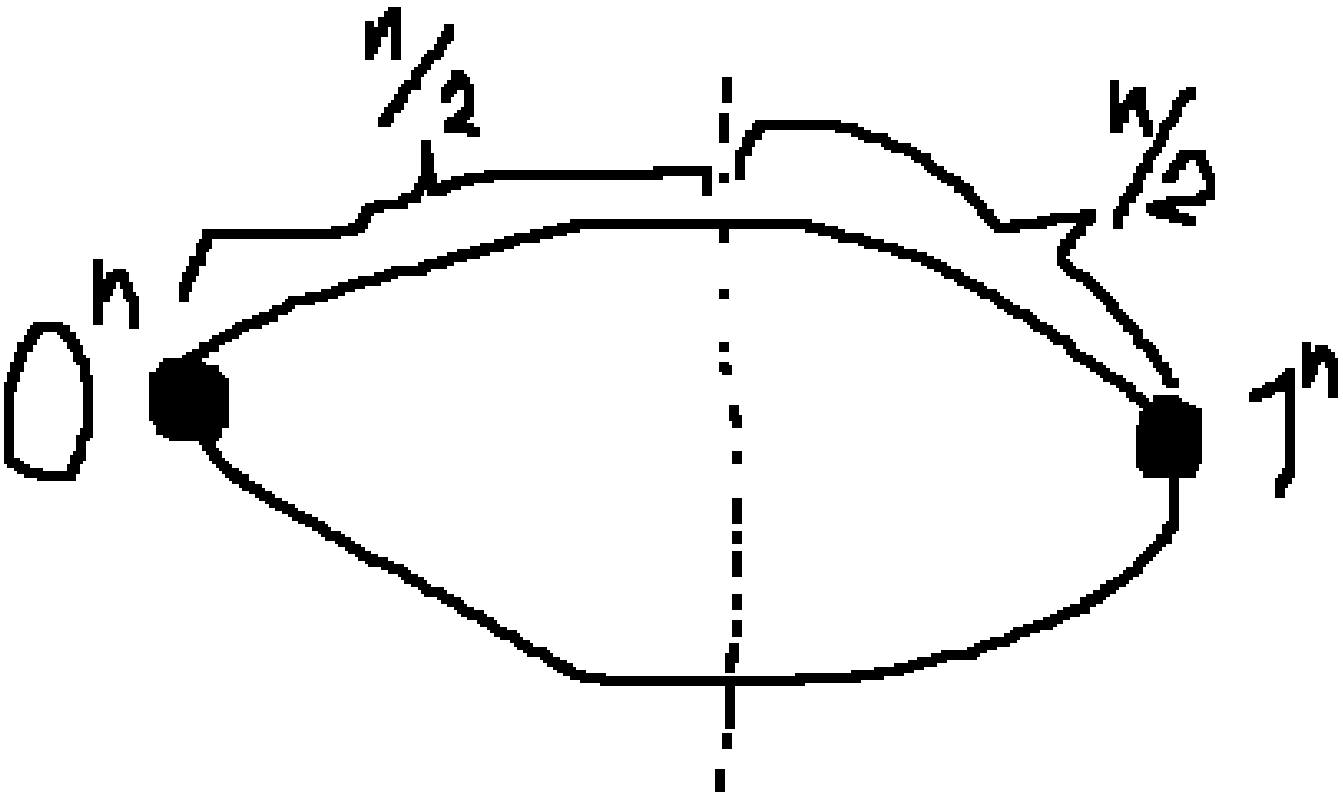
Antag att F innehåller n variabler.

Då kollar $X(F, (0, \ldots, 0), n)$ om F är satisfierbar eller inte.

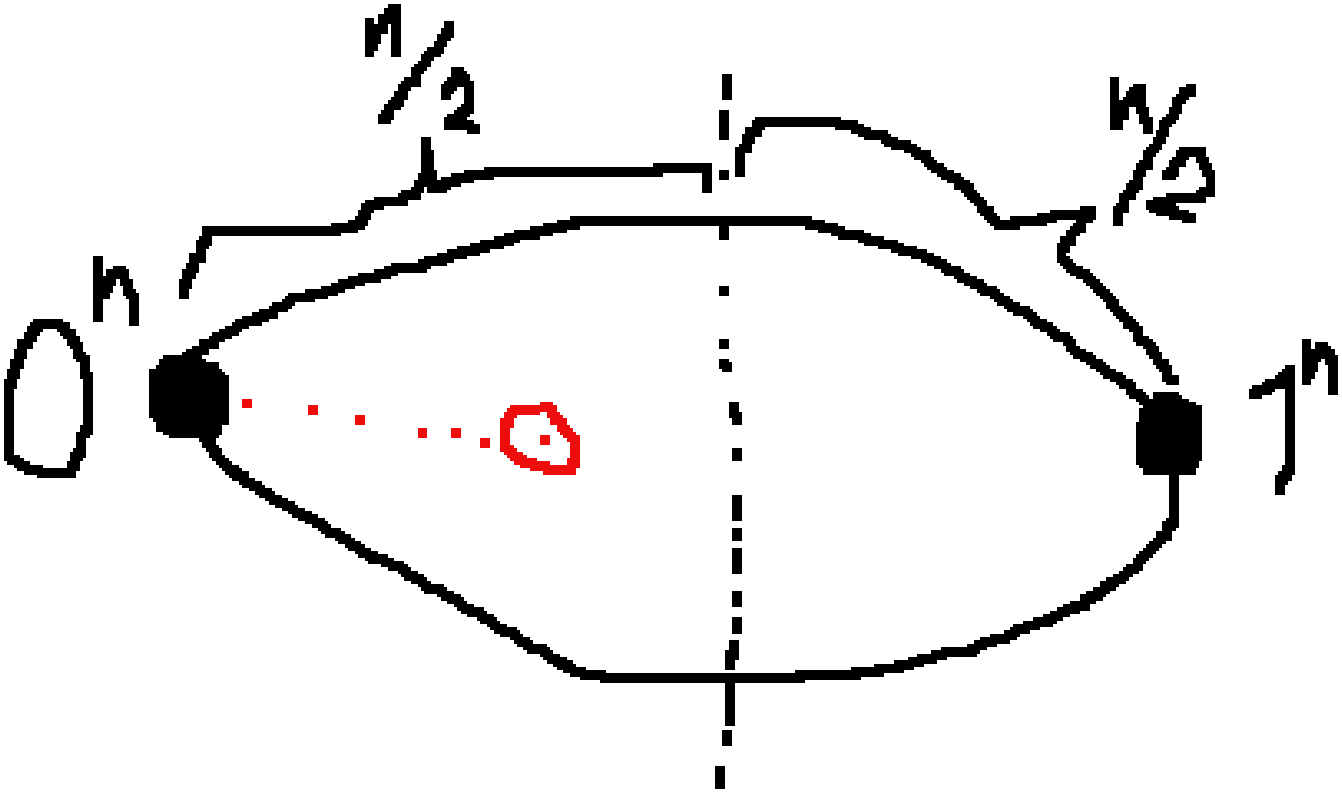
Tidskomplexitet: $3^n \cdot \mathrm{poly}(||F||)$.

Långsammare än att räkna upp alla tänkbara tilldelningar (2^n).

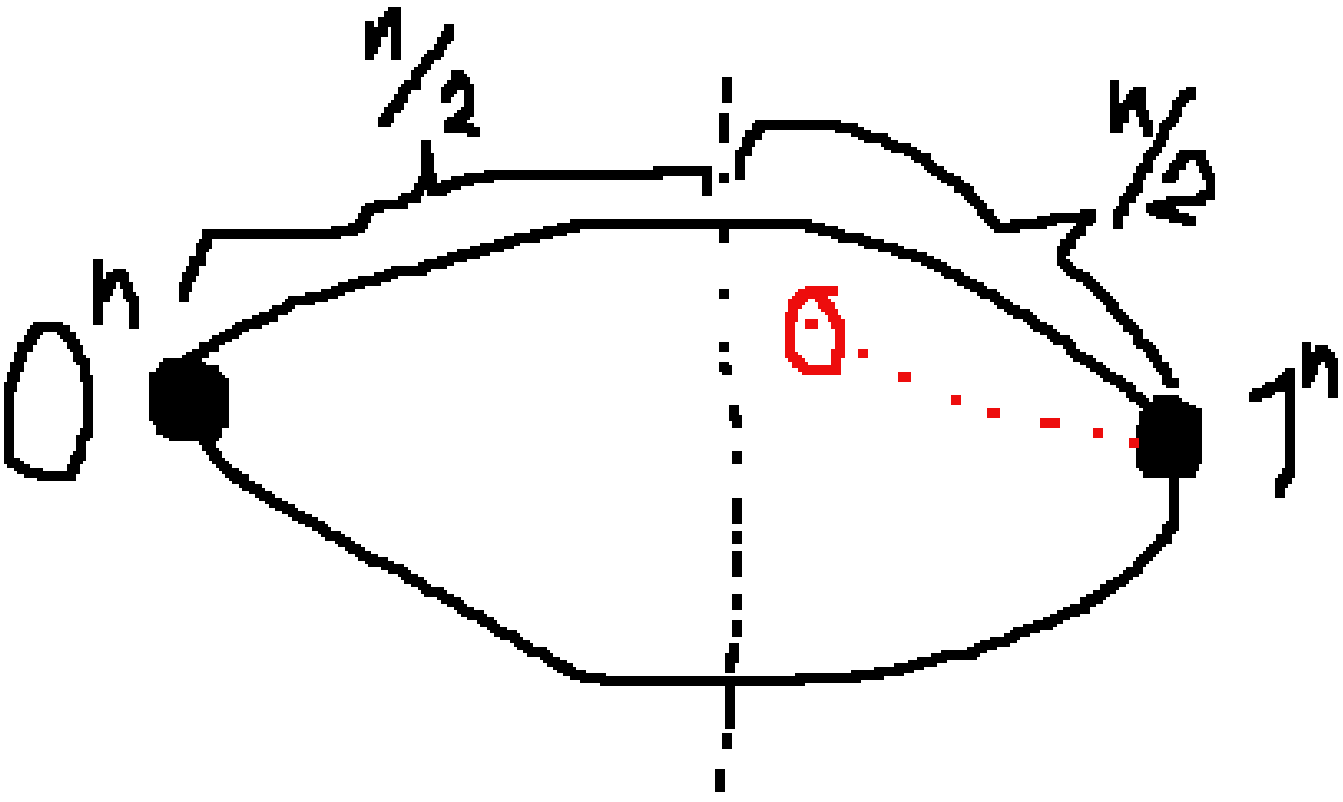
3SAT 2



3SAT 2



3SAT 2



3SAT 2

Man kan alltså kontrollera F:s satisfierbarhet via två anrop till X.

$X(F, (0, \ldots, 0), \lceil n/2 \rceil)$

$X(F, (1, \ldots, 1), \lceil n/2 \rceil)$

3SAT 2

Man kan alltså kontrollera F:s satisfierbarhet via två anrop till X.

$X(F, (0, \ldots, 0), \lceil n/2 \rceil)$

$X(F, (1, \ldots, 1), \lceil n/2 \rceil)$

Tidskomplexitet: $2 \cdot 3^{n/2} \cdot \mathrm{poly}(||F||)$ eller ungefär $O(1.73^n \cdot \mathrm{poly}(||F||))$.

Algoritm 1 för 3SAT går i $O(1.84^n \cdot \mathrm{poly}(||F||))$ tid.