# $AP$-Reduktion

Definition: $P$ ist $AP$-reduzierbar auf $Q$, $P \leq_{AP} Q$, falls es Funktionen $f$ und $g$ und $\alpha \geq 1$ gibt mit:

- für $x \in I_P$ und $1 < r \in \mathbb{Q}$ ist $f(x, r) \in I_Q$,
- ist $S_P(x) \neq \emptyset$, dann ist $S_Q(f(x, r)) \neq \emptyset$,
- für $y \in S_Q(f(x, r))$ ist $g(x, y, r) \in S_P(x)$,
- für alle $r > 1$ sind $\lambda x.f(x, r)$ und $\lambda x, y \,.\, g(x, y, r)$ in **FP**,
- für alle $y \in S_Q(f(x, r))$ gilt

$$R_Q(f(x, r), y) \leq r \quad \Longrightarrow \quad R_P(x, g(x, y, r) \leq 1 + \alpha(r - 1)\ .$$

# Abschluss unter $AP$-Reduktion

Lemma:

$$Q \in \textsf{\textbf{APX}} \quad \text{und} \quad P \leq_{AP} Q \quad \Longrightarrow \quad P \in \textsf{\textbf{APX}}$$

$$Q \in \textsf{\textbf{PTAS}} \quad \text{und} \quad P \leq_{AP} Q \quad \Longrightarrow \quad P \in \textsf{\textbf{PTAS}}$$

Beispiele:

$\textsc{Maximum Satisfiability} \leq_{AP} \textsc{Maximum Clique}$

$\textsc{Maximum Clique} \leq_{AP} \textsc{Maximum Independent Set}$

Korollar: $\textsf{\textbf{P}} \neq \textsf{\textbf{NP}} \quad \Longrightarrow \textsc{Maximum Independent Set} \notin \textsf{\textbf{APX}}$

# L-Reduktion

Definition: $P$ ist L-reduzierbar auf $Q$, $P \leq_L Q$, falls es Funktionen $f : I_P \to I_Q$ und $g$ in **FP** und $\beta, \gamma \geq 0$ gibt mit:

- ist $S_P(x) \neq \emptyset$, dann ist $S_Q(f(x)) \neq \emptyset$,
- für $y \in S_Q(f(x))$ ist $g(x, y) \in S_P(x)$,
- für alle $x \in I_P$ ist $m_Q^*(f(x)) \leq \beta m_P^*(x)$
- für alle $y \in S_Q(f(x))$ gilt

$$|m_P^*(x) - m_P(x, g(x, y))| \leq \gamma |m_Q^*(f(x)) - m_Q(f(x), y)| \,.$$

Satz: Ist $P \leq_L Q$ und $P \in$ **APX**, dann auch $P \leq_{AP} Q$.

# Ein NPO-vollständiges Problem

MAXIMUM (MINIMUM) WEIGHTED SATISFIABILITY:

| | |
|---|---|
| Instanz: | Aussagenlogische Formel $\varphi$ in Variablen $x_1, \ldots x_n$, für jede Variable $x_i$ Gewicht $w_i \geq 0$ |
| Lösung: | Bewertung $\alpha$ mit $\alpha \models \varphi$ |
| Maß: | $\max(1, \sum_{i=1}^{n} w_i \alpha(x_i))$ |

Satz: MAXIMUM (MINIMUM) WEIGHTED SATISFIABILITY ist vollständig für die Klasse der Maximierungs- (Minimierungs-) probleme in **NPO**.

Satz: MAXIMUM WEIGHTED SATISFIABILITY $\leq_{AP}$ MINIMUM WEIGHTED SATISFIABILITY, und umgekehrt.

# **APX**-Vollständigkeit

Satz: Maximum 3-Satisfiability ist vollständig für die Klasse der Maximierungsprobleme in **APX**.

Satz: Für jedes Minimierungsproblem $\mathsf{P}$ in **APX** gibt es ein Maximierungsproblem $\mathsf{P}' \in$ **APX** mit $\mathsf{P} \leq_{AP} \mathsf{P}'$.

Korollar: Maximum 3-Satisfiability ist **APX**-vollständig.

# Mehr **APX**-vollständige Probleme

Lemma:

1. Maximum 3-Satisfiability $\leq_{\mathrm{L}}$ Maximum 2-Satisfiability
2. Maximum 2-Satisfiability $\leq_{\mathrm{L}}$ Maximum Not-All-Equal 3-Satisfiability
3. Maximum Not-All-Equal 3-Satisfiability $\leq_{\mathrm{L}}$ Maximum Cut

Ausserdem: Maximum Not-All-Equal 3-Satisfiability $\in$ **APX**

Korollar: Maximum 2-Satisfiability, Maximum Not-All-Equal 3-Satisfiability und Maximum Cut sind **APX**-vollständig.

## Eingeschränkte Erfüllbarkeitsprobleme:

MAXIMUM 3-SATISFIABILITY($b$): Einschränkung von MAXIMUM 3-SATISFIABILITY auf Instanzen, in denen jede Variable höchstens $b$ mal vorkommt.

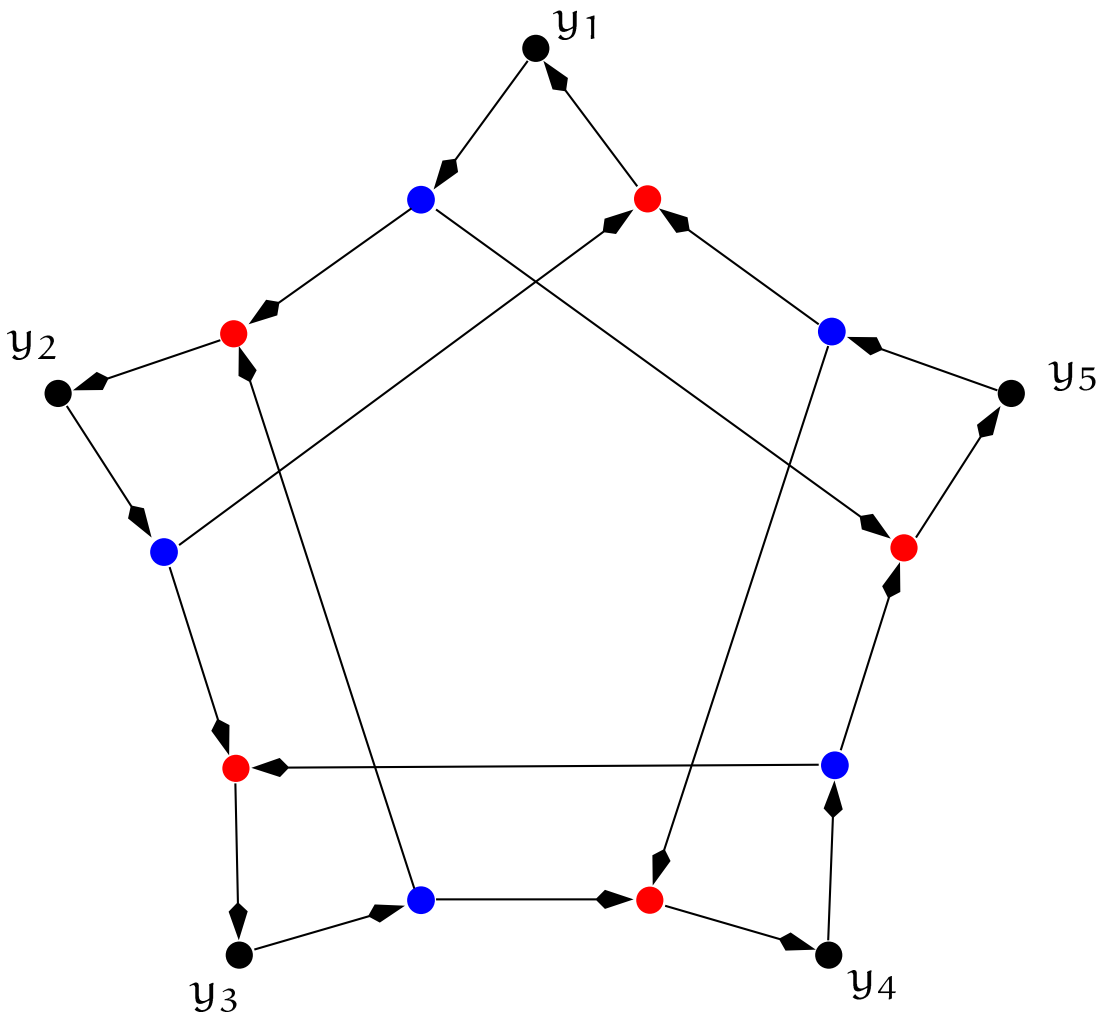
Lemma:

MAXIMUM 3-SATISFIABILITY $\leq_L$ MAXIMUM 3-SATISFIABILITY(29)

MAXIMUM 3-SATISFIABILITY(29) $\leq_L$ MAXIMUM 3-SATISFIABILITY(5) (Übung)

MAXIMUM 3-SATISFIABILITY(5) $\leq_L$ MAXIMUM 3-SATISFIABILITY(3)

Satz: MAXIMUM 3-SATISFIABILITY(3) ist **APX**-volständig.



$y_1$
$y_2$
$y_5$
$y_3$
$y_4$

# Ein Problem in log-APX

MINIMUM SET COVER

| | |
|---|---|
| Instanz: | Menge $C$ von Teilmengen einer Menge $S$ |
| Lösung: | Teilmenge $C' \subseteq C$ sodass $\forall s \in S\ \exists c \in C'$ mit $s \in c$ |
| Maß: | $\lvert C' \rvert$ |

MINIMUM SET COVER ist äquivalent zu MINIMUM HITTING SET.

# Greedy Algorithmus für Minimum Set Cover

Sei $C := \{c_1, \ldots, c_m\}$, $|S| = n$.

$U := S$ ; $C' := \emptyset$

$\mathtt{for}\ i := 1\ \mathtt{to}\ m\ \mathtt{do}\ c_i' := c_i$

$\mathtt{while}\ U \neq \emptyset\ \mathtt{do}$

  $\mathtt{choose}\ i\ \mathtt{with}\ |c_i'|\ \mathtt{maximal}$

  $C' := C' \cup \{c_i\}$

  $U := U \setminus c_i'$

  $\mathtt{for}\ j := 1\ \mathtt{to}\ m\ \mathtt{do}\ c_j' := c_j' \setminus c_i$

$\mathtt{return}\ C'$

Satz: Der obige Algorithmus ist ein $(\lfloor \ln n \rfloor + 1)$-Approximationsverfahren, also Minimum Set Cover $\in$ log-**APX**.

## Algorithmus für Minimum Graph Coloring

$U := V$ ; $i := 0$;
`while` $U \neq \emptyset$ `do`
  $i := i + 1$ ; $W := U$
  `while` $W \neq \emptyset$ `do`
    `choose` $v$ `of minimal degree in` $W$
    $f(v) := i$
    $W := W \setminus (\{v\} \cup \Gamma(v))$
  $U := U \setminus f^{-1}[i]$
`return` $f$

Satz: Der obige Algorithmus färbt einen k-färbbaren Graphen mit höchstens $3|V|/\log_k |V|$ Farben.

Korollar: Der Algorithmus ist ein $O(n/\log n)$-Approximationsverfahren, also ist Minimum Graph Coloring $\in$ poly-**APX**.