

Komplexität von Suchproblemen und Beweissystemen

Ein Abschlusskolloquium von Anton Ehrmanntraut

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Graph G

Prädikatenlog.
Formel ϕ

Gesucht

Matching größter Kardinalität in G

Beweis im Sequenzkalkül für
Gültigkeit von ϕ
(oder „ ϕ ungültig ausgeben“)

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Graph $G, k \in \mathbb{N}$

Clique C in G mit $\geq k$ Knoten
(oder „gibt keine Clique d. Größe k “ ausgeben)

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Graph $G, k \in \mathbb{N}$

Clique C in G mit $\geq k$ Knoten
(oder „gibt keine Clique d. Größe k “ ausgeben)

Graph G

Clique C in G größter Kardinalität

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Graph $G, k \in \mathbb{N}$

Clique C in G mit $\geq k$ Knoten
(oder „gibt keine Clique d. Größe k “ ausgeben)

Graph G

Clique C in G größter Kardinalität

Graph G

Hamiltonzyklus P in G
(oder „nicht hamiltonisch“ ausgeben)

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Graph G , $k \in \mathbb{N}$

Clique C in G mit $\geq k$ Knoten
(oder „gibt keine Clique d. Größe k “ ausgeben)

Graph G

Clique C in G größter Kardinalität

Graph G

Hamiltonzyklus P in G
(oder „nicht hamiltonisch“ ausgeben)

Graph G ,
Hamiltonzyklus P

Hamiltonzyklus $P' \neq P$ in G
(oder „gibt keinen anderen“ ausgeben)

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Graph G , $k \in \mathbb{N}$

Clique C in G mit $\geq k$ Knoten
(oder „gibt keine Clique d. Größe k “ ausgeben)

Graph G

Clique C in G größter Kardinalität

Graph G

Hamiltonzyklus P in G
(oder „nicht hamiltonisch“ ausgeben)

Graph G ,
Hamiltonzyklus P

Hamiltonzyklus $P' \neq P$ in G
(oder „gibt keinen anderen“ ausgeben)

Graphen G , H

Graphisomorphismus σ von G nach H
(oder „nicht isomorph“ ausgeben)

Welches dieser Suchprobleme ist effizient (ließ: in P-Zeit) lösbar?

Gegeben

Gesucht

Graph G

Matching größter Kardinalität in G

Aussagenlog. Formel ψ

Belegung, welche ψ erfüllt
(oder „ ψ unerfüllbar ausgeben“)

Graph G , $k \in \mathbb{N}$

Clique C in G mit $\geq k$ Knoten
(oder „gibt keine Clique d. Größe k “ ausgeben)

Graph G

Clique C in G größter Kardinalität

Graph G

Hamiltonzyklus P in G
(oder „nicht hamiltonisch“ ausgeben)

Graph G ,
Hamiltonzyklus P

Hamiltonzyklus $P' \neq P$ in G
(oder „gibt keinen anderen“ ausgeben)

Graphen G, H

Graphisomorphismus σ von G nach H
(oder „nicht isomorph“ ausgeben)

Natürliche Zahl $k > 1$

Primfaktor von k

State of the art



Empirische Evidenz, oder *strategic ambiguity*

In der Praxis interessiert man sich vor allem für die Lösung der Optimierungsvariante.

Es ist leicht zu sehen, dass man die Entscheidungsvariante auf die Berechnungsvariante und die Berechnungsvariante auf die Optimierungsvariante zurückführen kann ($E \leq B \leq O$).

Umgekehrt kann man aber auch die Optimierungsvariante auf die Entscheidungsvariante zurückführen ($O \leq E$). Mit Hilfe binärer Suche und durch das Lösen der Entscheidungsvarianten mehrerer Teilprobleme lässt sich eine optimale Rundreise in polynomieller Zeit ermitteln.

Die drei Problemvarianten sind also gleich schwierig ($E \equiv B \equiv O$). Dies ist typisch für Optimierungsprobleme.

⇒ es genügt, die Komplexität der Entscheidungsvarianten zu untersuchen

Empirische Evidenz, oder *strategic ambiguity*

In der Praxis interessiert man sich vor allem für die Lösung der Optimierungsvariante.

Es ist leicht zu sehen, dass man die Entscheidungsvariante auf die Berechnungsvariante und die Berechnungsvariante auf die Optimierungsvariante zurückführen kann ($E \leq B \leq O$).

Umgekehrt kann man aber auch die Optimierungsvariante auf die Entscheidungsvariante zurückführen ($O \leq E$). Mit Hilfe binärer Suche und durch das Lösen der Entscheidungsvarianten mehrerer Teilprobleme lässt sich eine optimale Rundreise in polynomieller Zeit ermitteln. „search reduces to decision“

Die drei Problemvarianten sind also gleich schwierig ($E \equiv B \equiv O$). Dies ist typisch für Optimierungsprobleme.

⇒ es genügt, die Komplexität der Entscheidungsvarianten zu untersuchen

Empirische Evidenz, oder *strategic ambiguity*

In der Praxis interessiert man sich vor allem für die Lösung der Optimierungsvariante.

Es ist leicht zu sehen, dass man die Entscheidungsvariante auf die Berechnungsvariante und die Berechnungsvariante auf die Optimierungsvariante zurückführen kann ($E \leq B \leq O$).

Umgekehrt kann man aber auch die Optimierungsvariante auf die Entscheidungsvariante zurückführen ($O \leq E$). Mit Hilfe binärer Suche und durch das Lösen der Entscheidungsvarianten mehrerer Teilprobleme lässt sich eine optimale Rundreise in polynomieller Zeit ermitteln. „search reduces to decision“

Die drei Problemvarianten sind also gleich schwierig ($E \equiv B \equiv O$).

Dies ist typisch für Optimierungsprobleme.

⇒ es genügt, die Komplexität der Entscheidungsvarianten zu untersuchen

Empirische Evidenz, oder *strategic ambiguity*

In der Praxis interessiert man sich vor allem für die Lösung der Optimierungsvariante.

Es ist leicht zu sehen, dass man die Entscheidungsvariante auf die Berechnungsvariante und die Berechnungsvariante auf die Optimierungsvariante zurückführen kann ($E \leq B \leq O$).

Umgekehrt kann man aber auch die Optimierungsvariante auf die Entscheidungsvariante zurückführen ($O \leq E$). Mit Hilfe binärer Suche und durch das Lösen der Entscheidungsvarianten mehrerer

Teilprobleme
ermittelt man

Die drei

Dies ist

⇒ es
zu

Exercise 2 – Decision, optimisation, construction

Strictly speaking, most of complexity theory is about *decision* problems: e.g. does this graph have an independent set of size k . We have been talking mostly about *optimisation* problems: what is the maximum size of an independent set in this graph. In practice, one often cares about the *construction* problem: compute an independent set of maximum size in this graph.

It is well known that for problems in the class NP these three notions are basically (polynomial-time) equivalent. That is, if you can solve one, you can solve the others. But that doesn't mean that there aren't better and worse ways to go about it.

Empirische Evidenz, oder *strategic ambiguity*

In der Praxis interessiert man sich vor allem für die Lösung der Optimierungsvariante.

Es ist leicht zu sehen, dass man die Entscheidungsvariante auf die Berechnungsvariante und die Berechnungsvariante auf die Optimierungsvariante zurückführen kann ($E \leq B \leq O$).

Umgekehrt kann man aber auch die Optimierungsvariante auf die Entscheidungsvariante zurückführen ($O \leq E$). Mit Hilfe binärer Suche und durch das Lösen der Entscheidungsvarianten mehrerer

Teilprobleme
ermittelt man

Die drei

Dies ist

⇒ es
zu

Exercise 2 – Decision, optimisation, construction

Strictly speaking, most of complexity theory is about *decision* problems: e.g. does this graph have an independent set of size k . We have been talking mostly about *optimisation* problems: what is the maximum size of an independent set in this graph. In practice, one often cares about the *construction* problem: compute an independent set of maximum size in this graph. offensichtlich „NP-vollständig“ gemeint!

It is well known that for problems in the class NP these three notions are basically (polynomial-time) equivalent. That is, if you can solve one, you can solve the others. But that doesn't mean that there aren't better and worse ways to go about it.

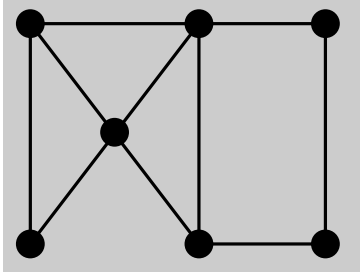
Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

$(G, 3)$

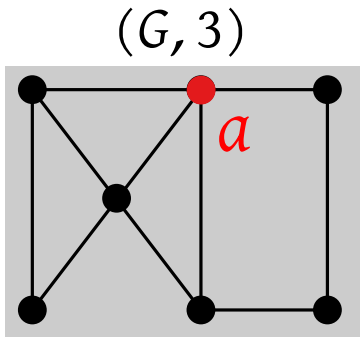


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

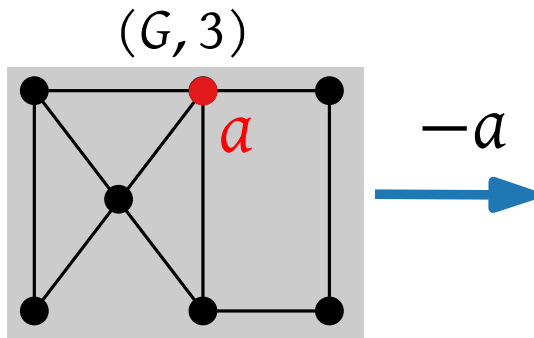


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

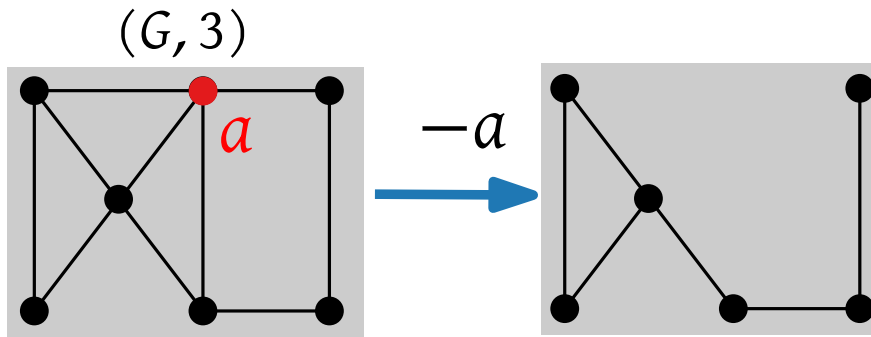


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

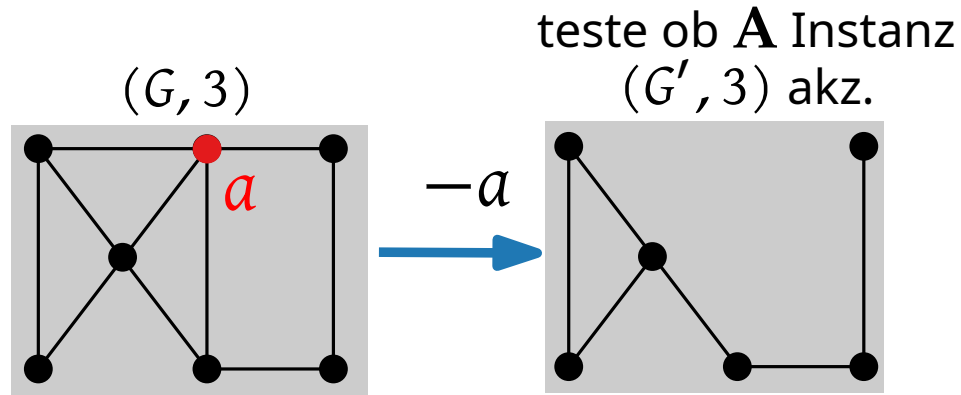


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

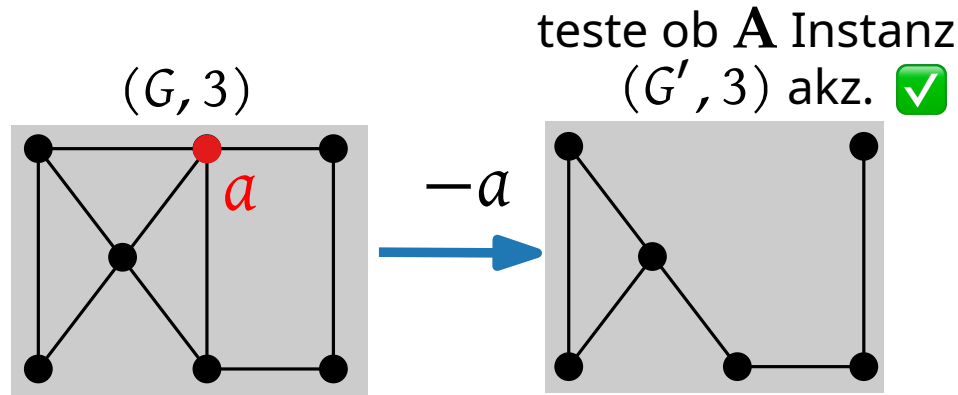


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

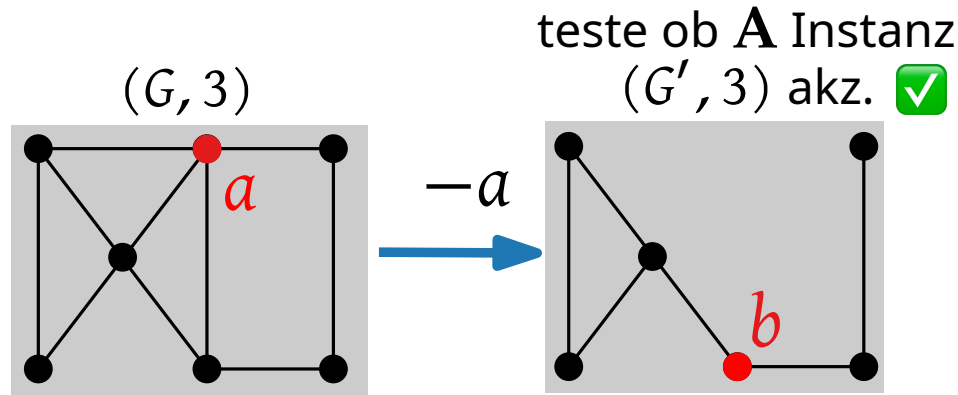


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

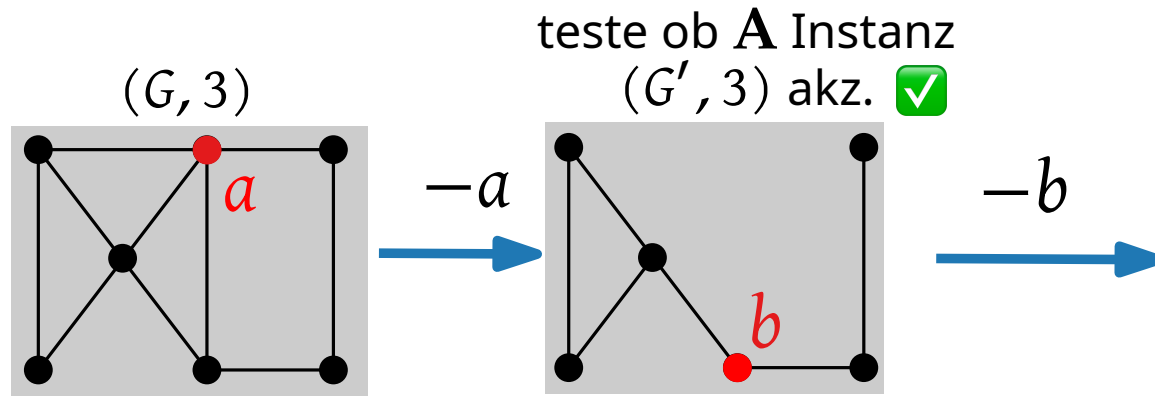


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

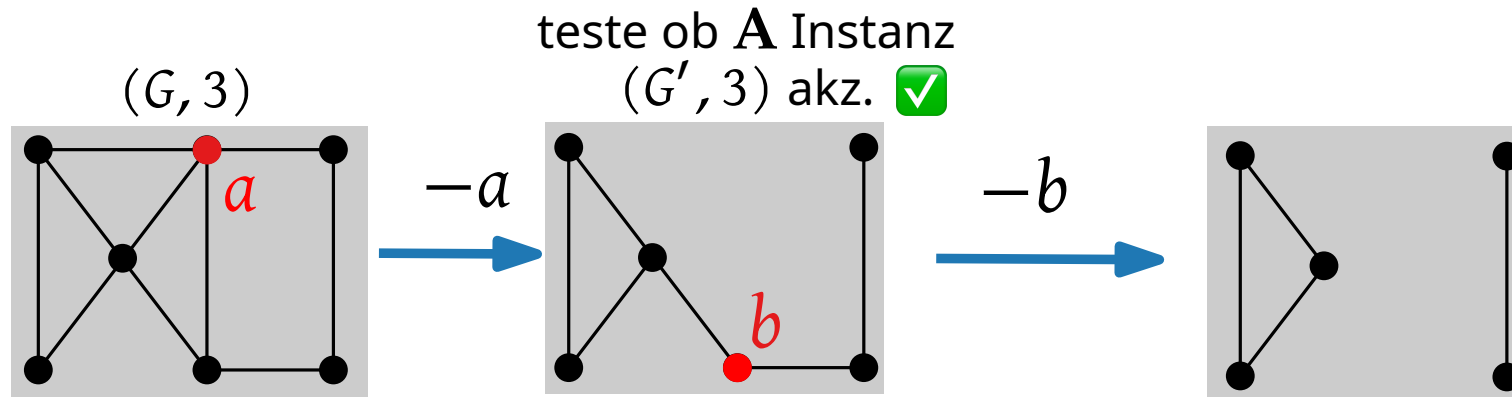


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

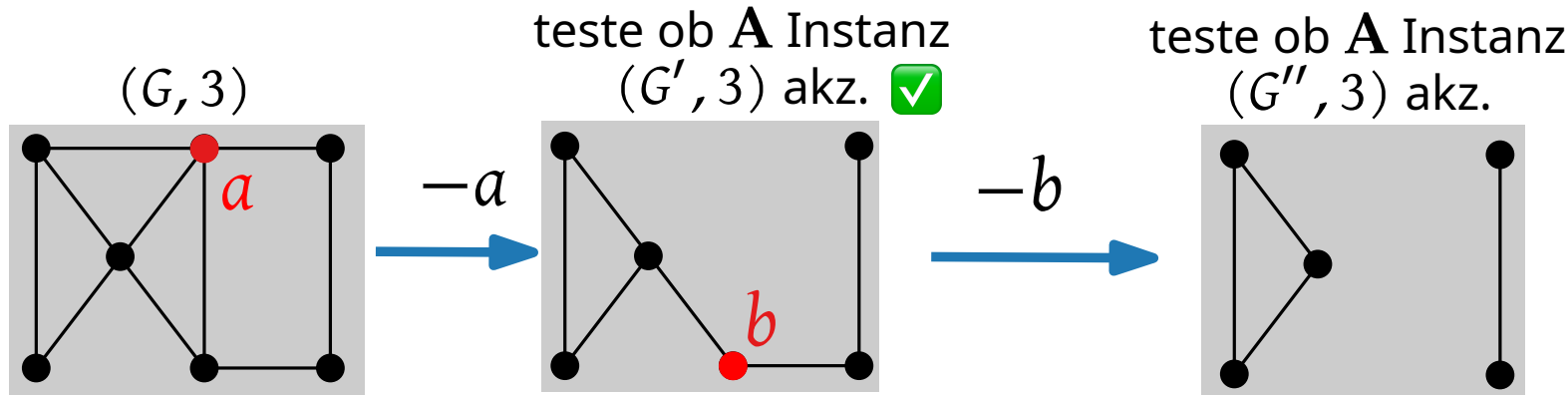


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

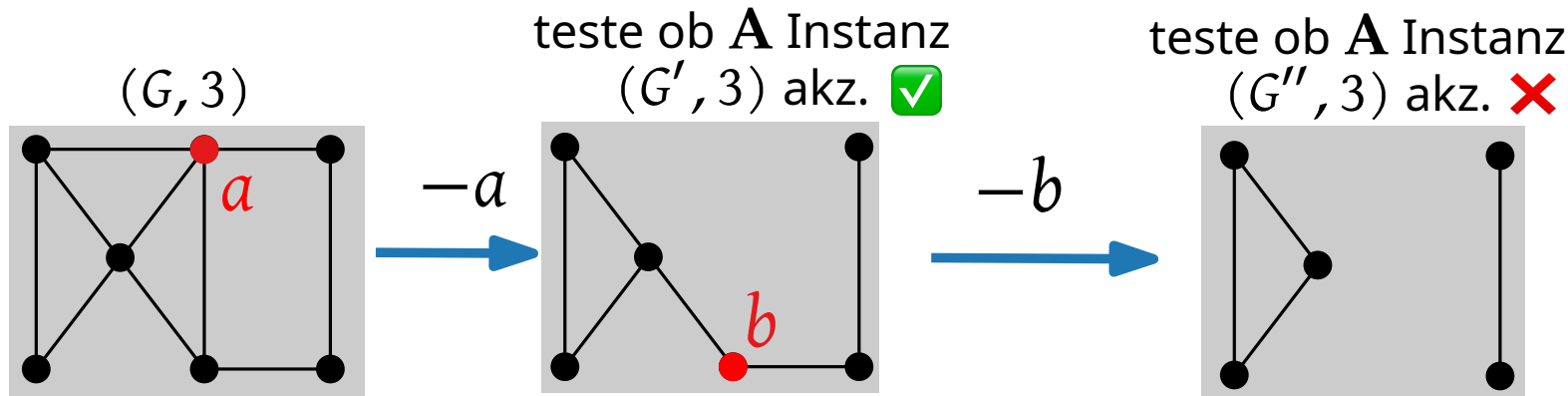


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

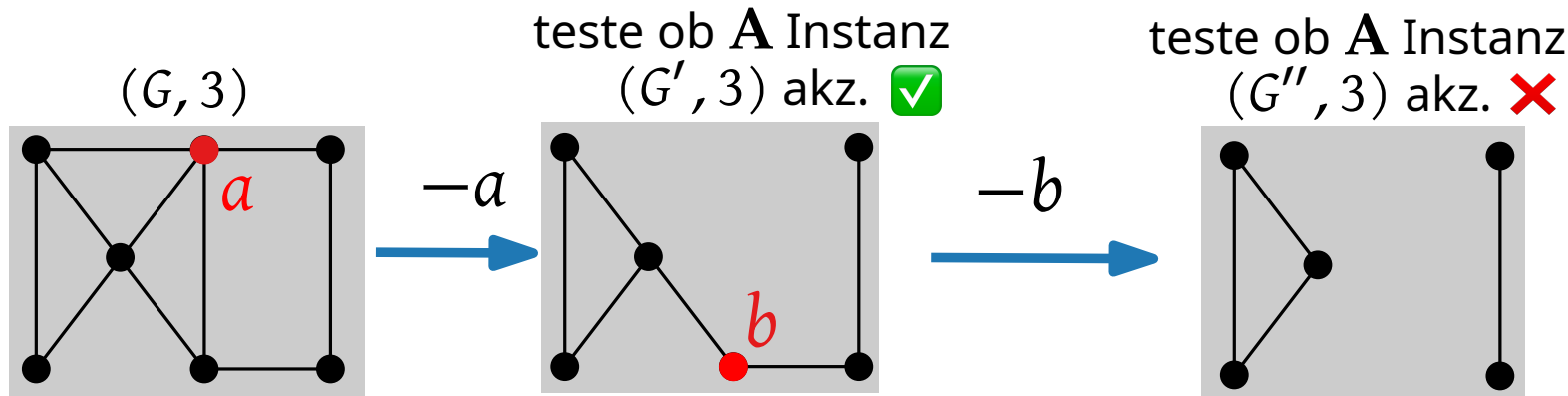


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET



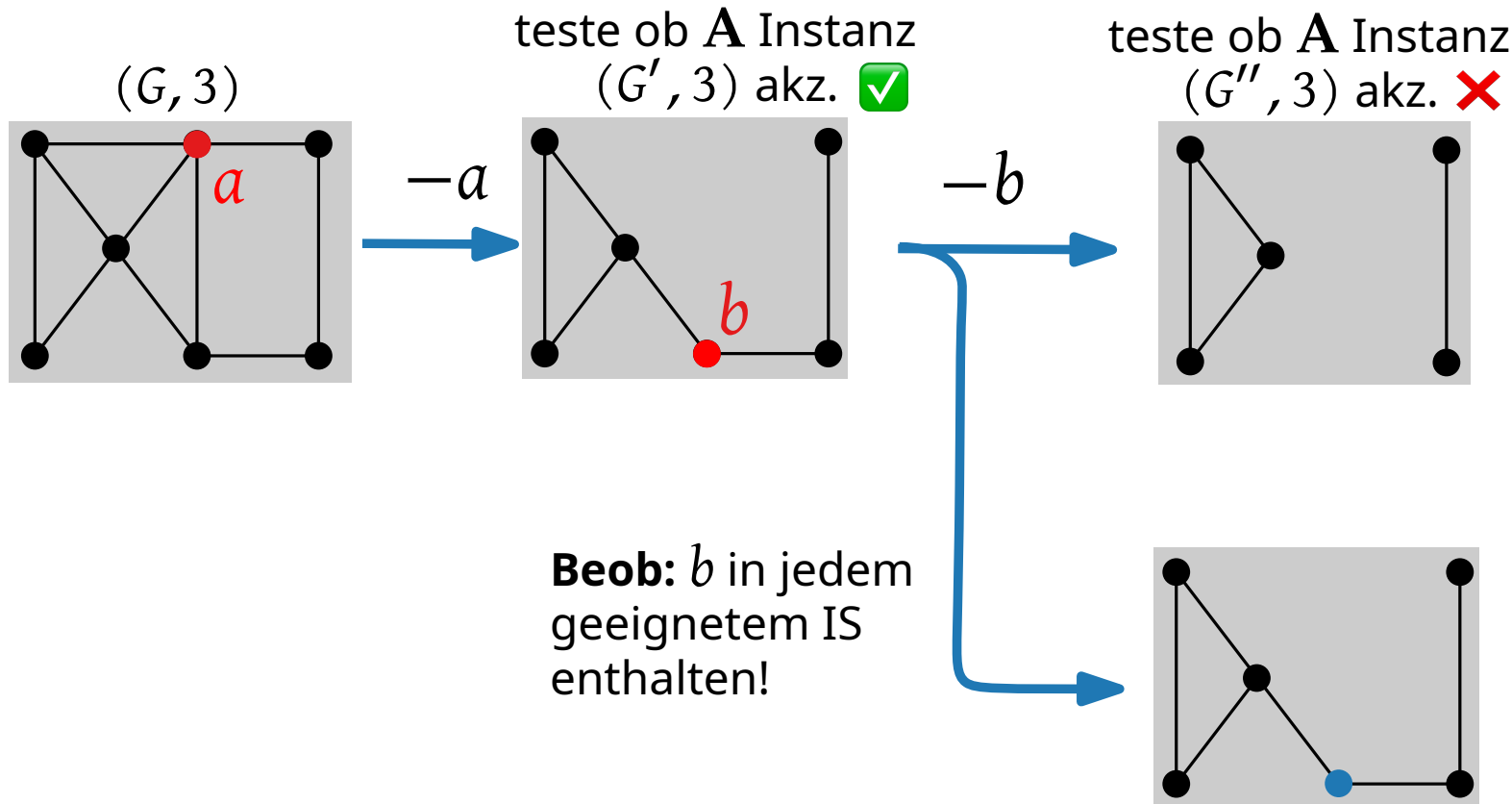
Beob: b in jedem
geeignetem IS
enthalten!

Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

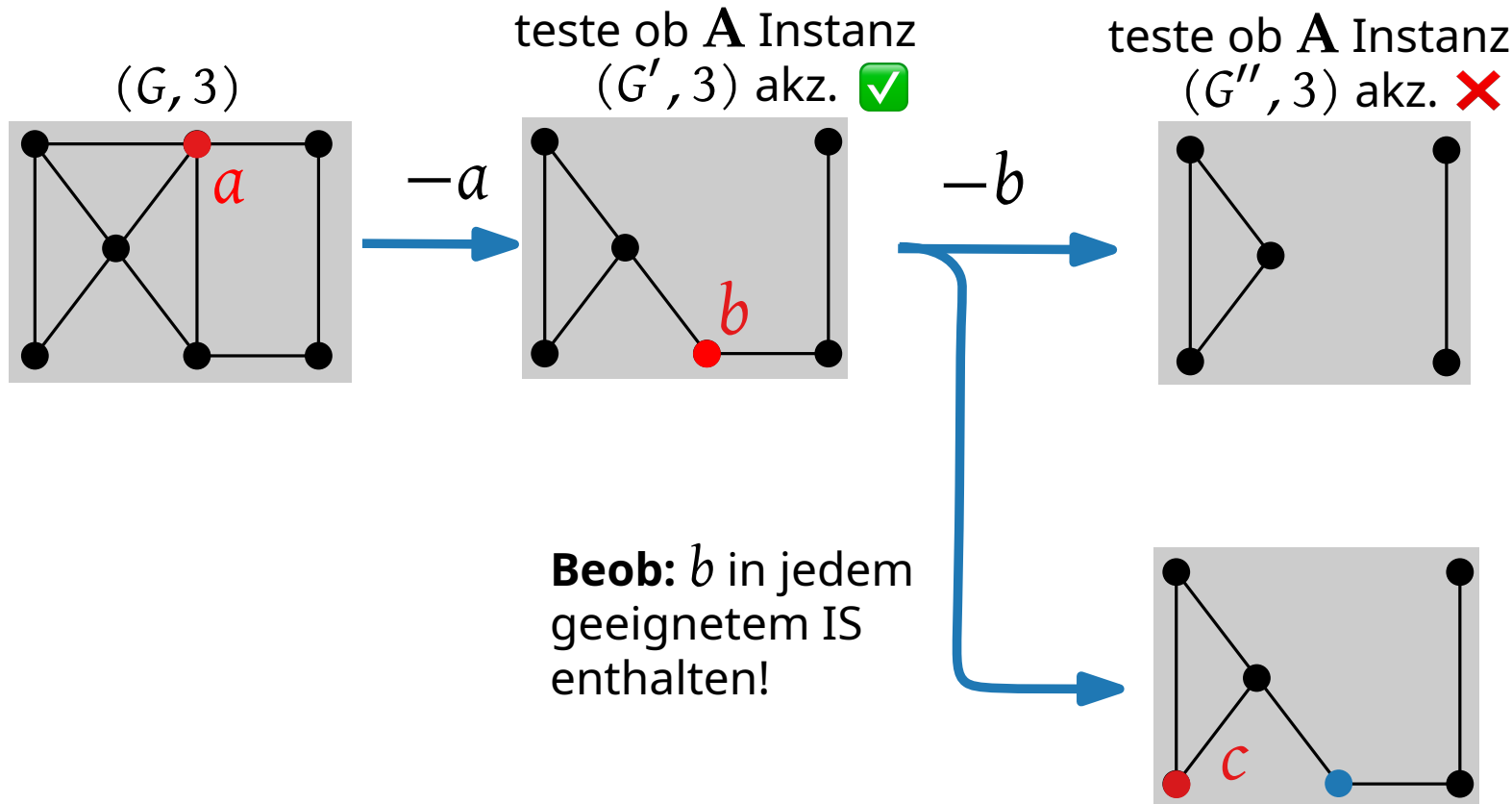


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

Gegeben: eff. Entscheidungsalg. **A** für INDSET

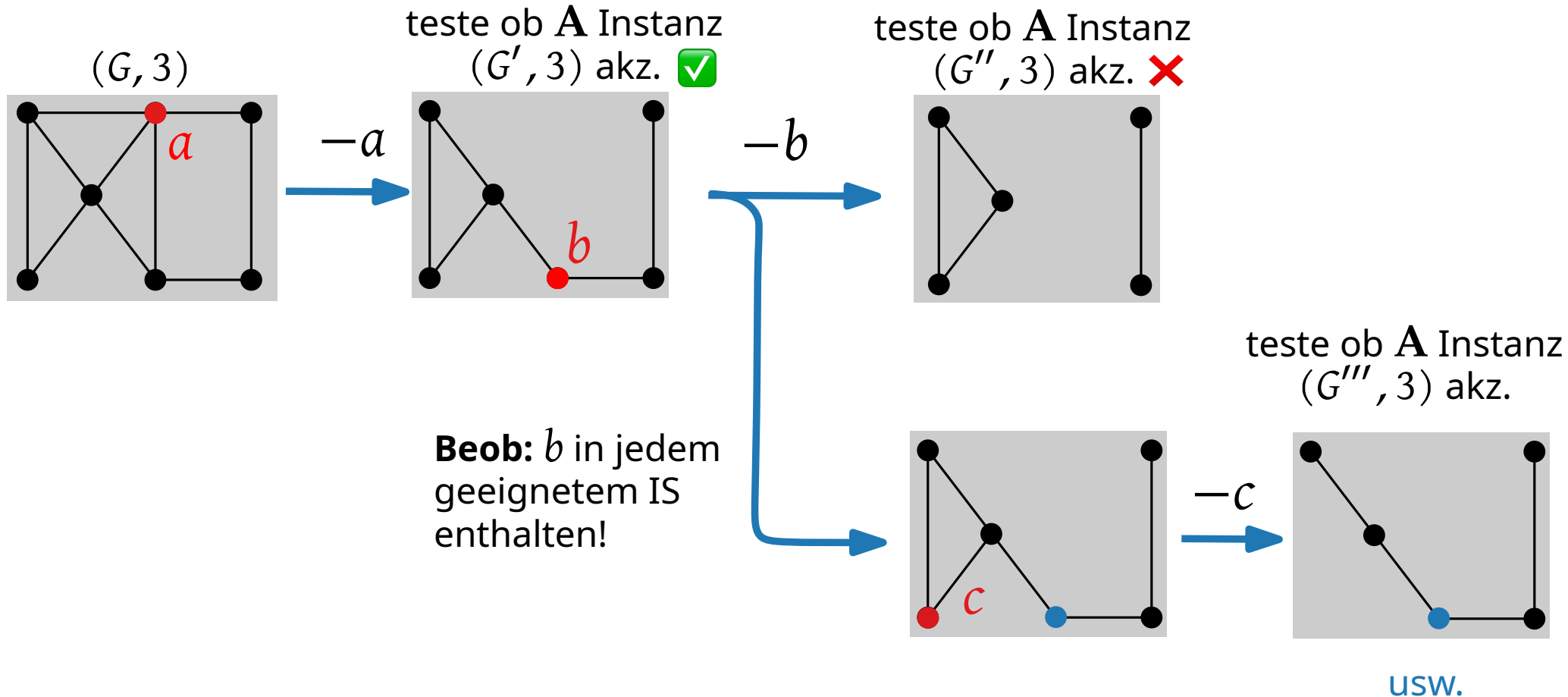


Search reduces to decision für „Unabh. Menge“

$$\text{rINDSET} = \{((G, k), I) \mid I \text{ unabh. Menge in } G, |I| \geq k\}$$

$$\text{INDSET} = \{(G, k) \mid \text{ex. unabh. Menge } I \text{ in } G, |I| \geq k\} = \text{Proj}(\text{rINDSET})$$

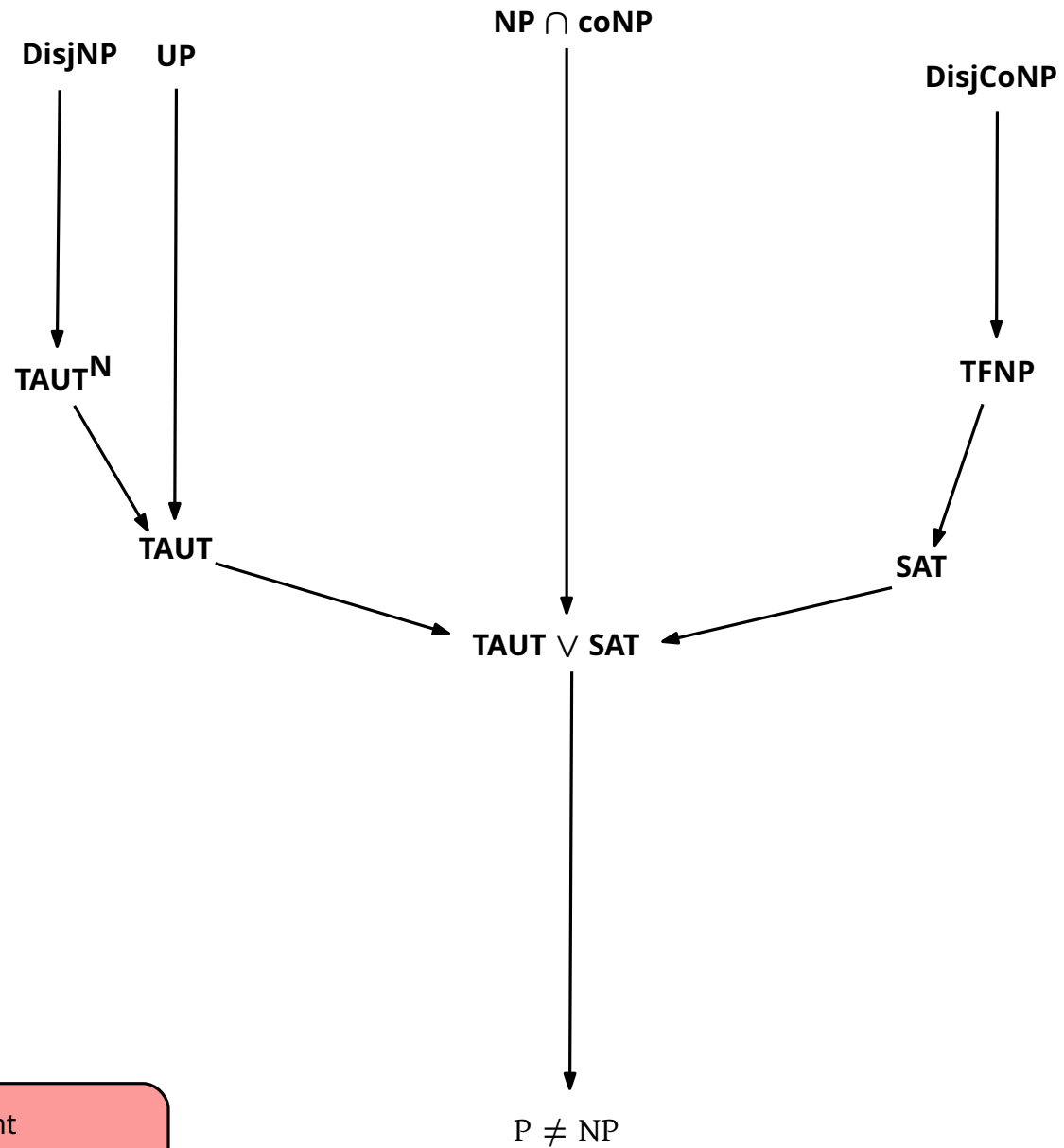
Gegeben: eff. Entscheidungsalg. **A** für INDSET



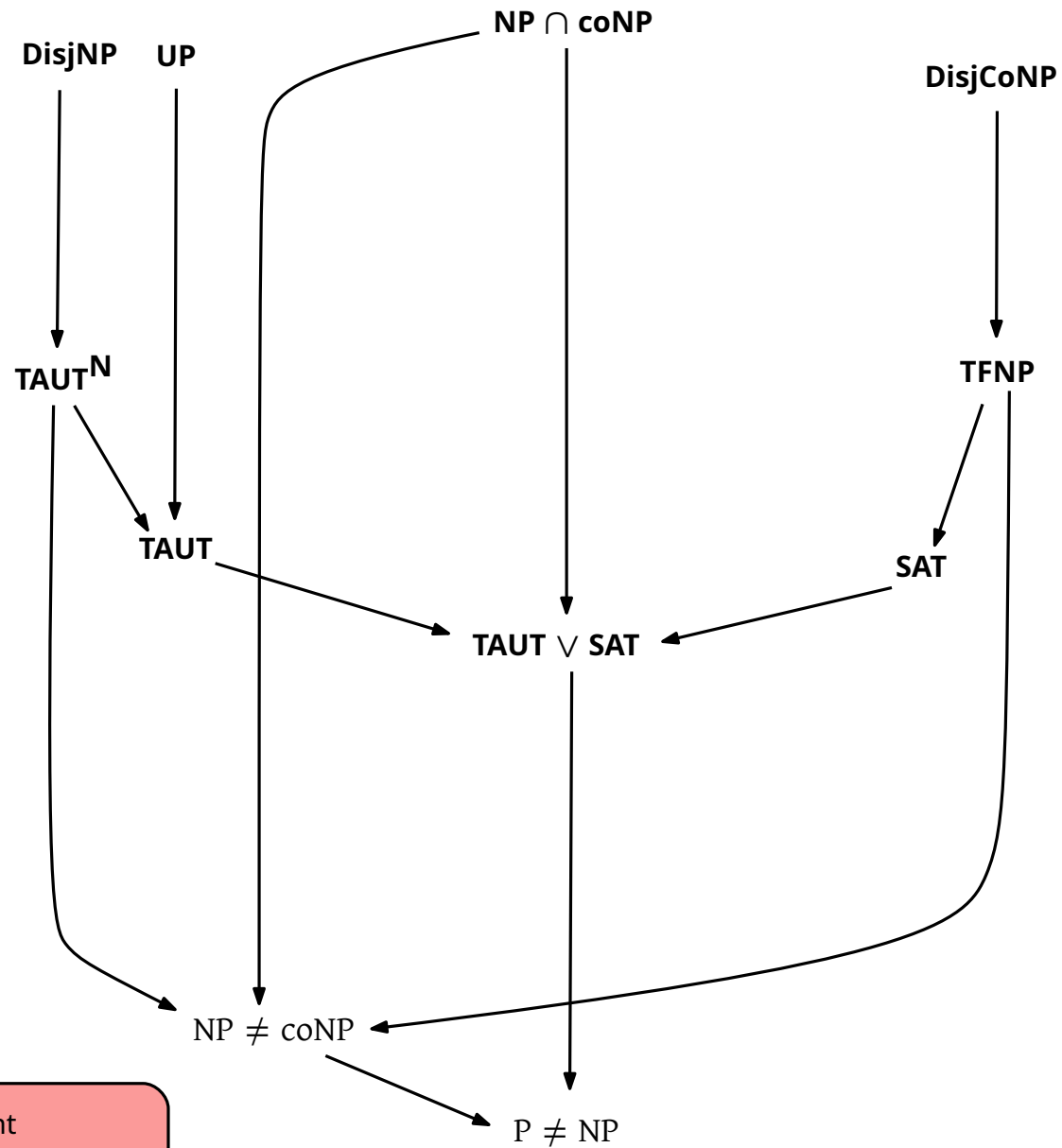
- TAUT** : ex. kein p-optimales Beweissystem für TAUT
- TAUT^N** : ex. keine optimales Beweissystem für TAUT
- SAT** : ex. kein p-optimales Beweissystem für SAT
- TFNP** : ex. keine Levin-vollständige NP-Relation für TFNP
- NP \cap coNP** : ex. keine many-one-vollständige Menge für NP \cap coNP
- UP** : ex. keine many-one-vollständige Menge für UP
- DisjNP** : ex. kein many-one-vollständiges Paar für DisjNP
- DisjCoNP** : ex. kein many-one-vollständiges Paar für DisjCoNP

- TAUT** : ex. kein p-optimales Beweissystem für TAUT
- TAUT^N** : ex. keine optimales Beweissystem für TAUT
- SAT** : ex. kein p-optimales Beweissystem für SAT
- TFNP** : ex. keine Levin-vollständige NP-Relation für TFNP
- NP** \cap **coNP** : ex. keine many-one-vollständige Menge für NP \cap coNP
- UP** : ex. keine many-one-vollständige Menge für UP
- DisjNP** : ex. kein many-one-vollständiges Paar für DisjNP
- DisjCoNP** : ex. kein many-one-vollständiges Paar für DisjCoNP

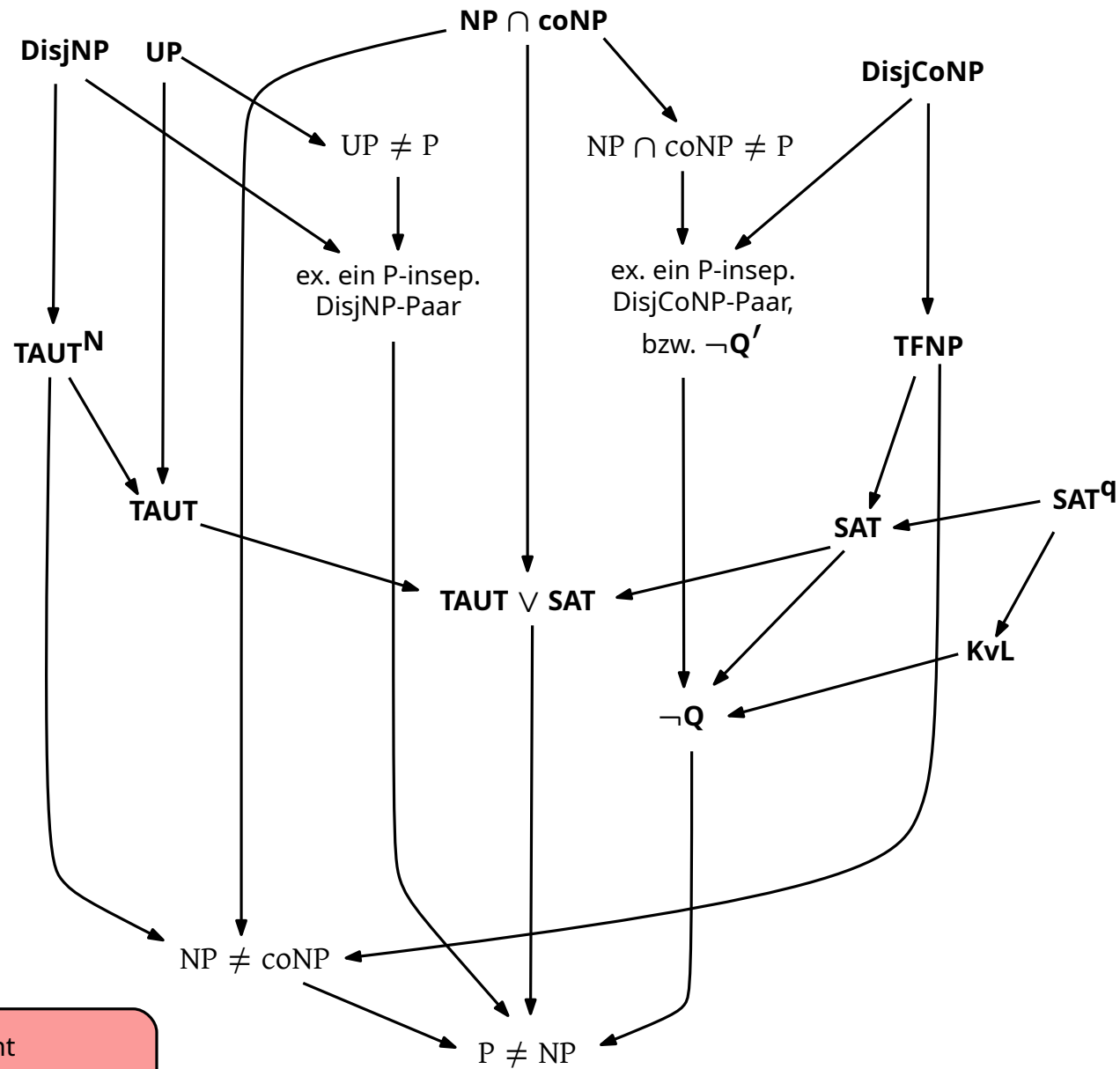
Konvention: $\mathbf{X} \triangleq X$ hat kein $\begin{cases} \text{vollst. Element} \\ \text{p-opt. Beweissystem} \end{cases}$



X : X hat kein $\begin{cases} \text{vollst. Element} \\ \text{p-opt. Beweissystem} \end{cases}$



X : X hat kein $\begin{cases} \text{vollst. Element} \\ \text{p-opt. Beweissystem} \end{cases}$



X : X hat kein $\begin{cases} \text{vollst. Element} \\ \text{p-opt. Beweissystem} \end{cases}$

Offene Fragen

- **Karp-vs-Levin-Reduktion** Wenn $\text{Proj}(R) \leq_m^p$ -vollst., ist dann auch $R \leq_L^P$ -vollst.?
- **Charakterisierungen von Q:** Gilt $Q \Leftrightarrow \neg \text{KvL}$?
Allgemein Beziehung zw. **Q**, **KvL**, **SAT**.
- **Orakelkonstruktionen:** Können wir relativierende Implikationen $\text{KvL} \Rightarrow \dots$ ausschließen?
- **Selbstreduzierbarkeit:** Erarbeitung einer robuster Definition von Selbstred. nach unten für NP-Relationen. Eigenschaften?