# $P^m$-Kompilierung von Planungsaufgaben im Fast-Downward Planungssystem als alternative Berechnung der $h^m$-Heuristik

Raphael Imahorn

Universität Basel

8. Februar 2016

Universität Basel

# Einführung

Haslum: $h_P^m = h_{P^m}^{max}$

Welcher Ansatz ist schneller?
Welcher Ansatz benötigt weniger Speicher?
Welcher Ansatz löst mehr Instanzen?

Universität Basel

# *STRIPS*-Planungsformalismus

Eine *STRIPS* Planungsaufgabe ist ein 4-Tupel $\Pi = \langle V, A, I, G \rangle$ mit:

- $V$: einer endlichen Menge Atome;

# *STRIPS*-Planungsformalismus

Eine *STRIPS* Planungsaufgabe ist ein 4-Tupel $\Pi = \langle V, A, I, G \rangle$ mit:

- $V$: einer endlichen Menge Atome;
- $A$: einer endlichen Menge von Aktionen
  $a = \langle pre(a), add(a), del(a), cost(a) \rangle$;
  anwendbar in $s$, falls $pre(a) \subseteq s$
  Nachfolgezustand $s' = (s \setminus del(a)) \cup add(a)$

# *STRIPS*-Planungsformalismus

Eine *STRIPS* Planungsaufgabe ist ein 4-Tupel $\Pi = \langle V, A, I, G \rangle$ mit:

- $V$: einer endlichen Menge Atome;
- $A$: einer endlichen Menge von Aktionen $a = \langle pre(a), add(a), del(a), cost(a) \rangle$;
- $I \subseteq V$: dem Startzustand, und
- $G \subseteq V$: der Zielbeschreibung.

# *STRIPS*-Planungsformalismus

## Beispiel

### *STRIPS*-Planungsaufgabe

$\Pi = \langle V, A, I, G \rangle$ *mit:*

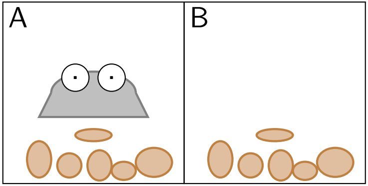
$V = \{$inRaum(A), inRaum(B), sauber(A), sauber(B)$\}$,

$A = \{$wechsleRaum(A,B), wechsleRaum(B,A), sauge(A), sauge(B)$\}$ *mit:*

wechlseRaum(X, Y)=$\langle\{$inRaum(X)$\}$, $\{$inRaum(Y)$\}$, $\{$inRaum(X)$\}$, 2$\rangle$

sauge(X)=$\langle\{$inRaum(X)$\}$, $\{$sauber(X)$\}$, $\emptyset$, 1$\rangle$,

$I = \{$inRaum(A)$\}$ *und* $G = \{$sauber(A), sauber(B)$\}$




Universität Basel

# $h^m$-Heuristiken

Eine $h^m$-Heuristik schätzt die maximalen Kosten ab, um eine Teilmenge von maximal $m$ Atomen zu erfüllen.

Die $h^{max}$-Heuristik entspricht der $h^1$-Heuristik.

# $P^m$-Kompilierung

$$P = \langle V, A, I, G \rangle \xrightarrow{P^m} P^m = \langle V^m, A^m, I^m, G^m \rangle$$

# $P^m$-Kompilierung

## Atommengen

$$X \subseteq V \quad \xrightarrow{P^m} \quad X^m = \{\pi_c \mid c \subseteq X,\ 1 \leq |c| \leq m\}$$

## $P^2$-Kompilierung: Variablen

$$V = \{\text{inRaum(A)}, \text{inRaum(B)}, \text{sauber(A)}, \text{sauber(B)}\}$$

$$\downarrow P^m$$

$$V^2 = \{\pi_{\{\text{inRaum(A)}\}}, \pi_{\{\text{inRaum(B)}\}}, \pi_{\{\text{sauber(A)}\}}, \pi_{\{\text{sauber(B)}\}}, \pi_{\{\text{inRaum(A), inRaum(B)}\}}, \pi_{\{\text{inRaum(A), sauber(A)}\}}, \pi_{\{\text{inRaum(A), sauber(B)}\}}, \pi_{\{\text{inRaum(B), sauber(A)}\}}, \pi_{\{\text{inRaum(B), sauber(B)}\}}, \pi_{\{\text{sauber(A), sauber(B)}\}}\}$$

# $P^m$-Kompilierung

## Aktionen: $f$

Für jede Menge $f \subseteq V$ mit $0 \leq |f| \leq m-1$, so dass $f$ disjunkt von $add(a)$ und von $del(a)$ gilt: $a \subseteq A \xrightarrow{P^m} \alpha_{a,f} \subseteq A^m$

## $P^2$-Kompilierung: $f$

für wechlseRaum(X,Y): $f \in \{\emptyset$, sauber(X), sauber(Y)$\}$
für sauge(X): $f \in \{\emptyset$, inRaum(X), inRaum(Y), sauber(Y)$\}$

# $P^m$-Kompilierung

## Aktionen: Vorbedingungen

Für jede Menge $f \subseteq V$ mit $0 \leq |f| \leq m-1$, so dass $f$ disjunkt von $add(a)$ und von $del(a)$ gilt: $a \subseteq A \xrightarrow{P^m} \alpha_{a,f} \subseteq A^m$ mit:
$pre(\alpha_{a,f}) = \{\pi_c \mid c \subseteq (pre(a) \cup f), 1 \leq |c| \leq m\}$

## $P^2$-Kompilierung: Vorbedingungen

$pre(\alpha_{\text{wechsleRaum(X,Y)},\emptyset}) = \{\pi_{\{\text{inRaum(X)}\}}\}$
$pre(\alpha_{\text{wechsleRaum(X,Y),sauber(X)}}) = \{\pi_{\{\text{inRaum(X)}\}}, \pi_{\{\text{sauber(X)}\}}, \pi_{\{\text{inRaum(X), sauber(X)}\}}\}$
$pre(\alpha_{\text{wechsleRaum(X,Y),sauber(Y)}}) = \{\pi_{\{\text{inRaum(X)}\}}, \pi_{\{\text{sauber(Y)}\}}, \pi_{\{\text{inRaum(X), sauber(Y)}\}}\}$
$pre(\alpha_{\text{sauge(X)},\emptyset}) = \{\pi_{\{\text{inRaum(X)}\}}\}$
$pre(\alpha_{\text{sauge(X),inRaum(X)}}) = \{\pi_{\{\text{inRaum(X)}\}}\}$
$pre(\alpha_{\text{sauge(X),inRaum(Y)}}) = \{\pi_{\{\text{inRaum(X)}\}}, \pi_{\{\text{inRaum(Y)}\}}, \pi_{\{\text{inRaum(X), inRaum(Y)}\}}\}$
$pre(\alpha_{\text{sauge(X),sauber(Y)}}) = \{\pi_{\{\text{inRaum(X)}\}}, \pi_{\{\text{sauber(Y)}\}}, \pi_{\{\text{inRaum(X), sauber(Y)}\}}\}$

# $P^m$-Kompilierung

## Aktionen: Add-Effekte

Für jede Menge $f \subseteq V$ mit $0 \leq |f| \leq m - 1$, so dass $f$ disjunkt von $add(a)$ und von $del(a)$ gilt: $a \subseteq A \xrightarrow{P^m} \alpha_{a,f} \subseteq A^m$ mit:

$add(\alpha_{a,f}) = \{\pi_c \mid c \subseteq (add(a) \cup f), (c \cap add(a)) \neq \emptyset, 1 \leq |c| \leq m\}$

## $P^2$-Kompilierung: Add-Effekte

$$add(\alpha_{\text{wechsleRaum(X,Y)},\emptyset}) = \{\pi_{\{\text{inRaum(Y)}\}}\}$$
$$add(\alpha_{\text{wechsleRaum(X,Y)},\text{sauber(X)}}) = \{\pi_{\{\text{inRaum(Y)}\}}, \pi_{\{\text{inRaum(Y), sauber(X)}\}}\}$$
$$add(\alpha_{\text{wechsleRaum(X,Y)},\text{sauber(Y)}}) = \{\pi_{\{\text{inRaum(Y)}\}}, \pi_{\{\text{inRaum(Y), sauber(Y)}\}}\}$$
$$add(\alpha_{\text{sauge(X)},\emptyset}) = \{\pi_{\{\text{sauber(X)}\}}\}$$
$$add(\alpha_{\text{sauge(X)},\text{inRaum(X)}}) = \{\pi_{\{\text{sauber(X)}\}}, \pi_{\{\text{sauber(X), inRaum(X)}\}}\}$$
$$add(\alpha_{\text{sauge(X)},\text{inRaum(Y)}}) = \{\pi_{\{\text{sauber(X)}\}}, \pi_{\{\text{sauber(X), inRaum(Y)}\}}\}$$
$$add(\alpha_{\text{sauge(X)},\text{sauber(Y)}}) = \{\pi_{\{\text{sauber(X)}\}}, \pi_{\{\text{sauber(X), sauber(Y)}\}}\}$$

# $P^m$-Kompilierung

## Aktionen: Delete-Effekte

Für jede Menge $f \subseteq V$ mit $0 \leq |f| \leq m - 1$, so dass $f$ disjunkt von $add(a)$ und von $del(a)$ gilt: $a \subseteq A \xrightarrow{P^m} \alpha_{a,f} \subseteq A^m$ mit:
$del(\alpha_{a,f}) = \emptyset$

## $P^2$-Kompilierung: Delete-Effekte

$del(\alpha_{\text{wechsleRaum(X,Y)},f}) = \emptyset$
$del(\alpha_{\text{sauge(X)},f}) = \emptyset$

# $P^m$-Kompilierung

## Aktionen: Kosten

Für jede Menge $f \subseteq V$ mit $0 \leq |f| \leq m - 1$, so dass $f$ disjunkt von $add(a)$ und von $del(a)$ gilt: $a \subseteq A \xrightarrow{P^m} \alpha_{a,f} \subseteq A^m$ mit:
$cost(\alpha_{a,f}) = cost(a)$

## $P^2$-Kompilierung: Aktionskostens

$cost(\alpha_{\text{wechsleRaum(X,Y)},f}) = 2$
$cost(\alpha_{\text{sauge(X)},f}) = 1$

# $P^m$-Kompilierung

## Atommengen

$$X \subseteq V \quad \xrightarrow{P^m} \quad X^m = \{\pi_c \mid c \subseteq X,\ 1 \leq |c| \leq m\}$$

## $P^2$-Kompilierung: Startzustand

$$I = \{\text{inRaum(A)}\} \quad \xrightarrow{P^m} \quad I^2 = \{\pi_{\{\text{inRaum(A)}\}}\}$$

## $P^2$-Kompilierung: Zielbeschreibung

$$G = \{\text{sauber(A)}, \text{sauber(B)}\}$$

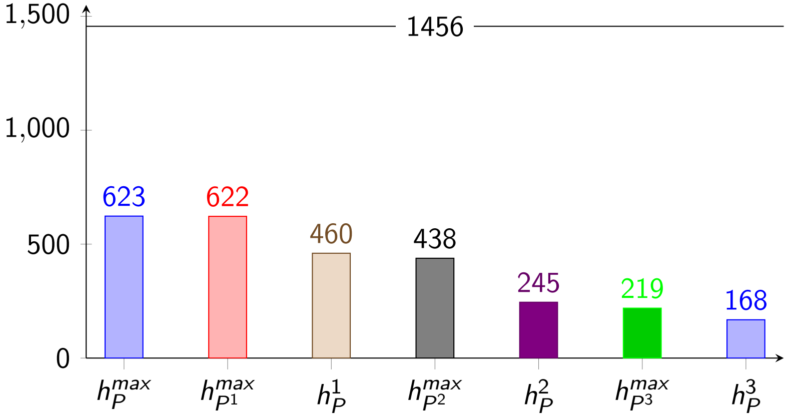
$$\downarrow P^m$$

$$G^2 = \{\pi_{\{\text{sauber(A)}\}}, \pi_{\{\text{sauber(B)}\}}, \pi_{\{\text{sauber(A)}, \text{sauber(B)}\}}\}$$

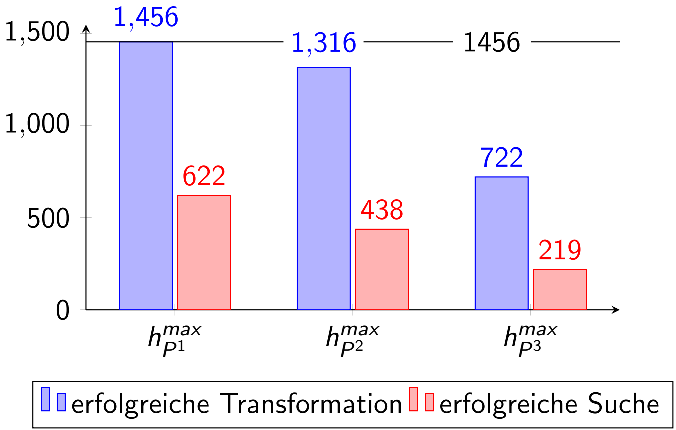
# Experimente

- 1456 Probleminstanzen
- Zeitlimit: 30 Minuten
- Speicherlimit: 2048 MB
- $m \in \{1, 2, 3\}$
- $P^m$-Kompilierung vor der Suche berechnet und im Speicher gehalten.

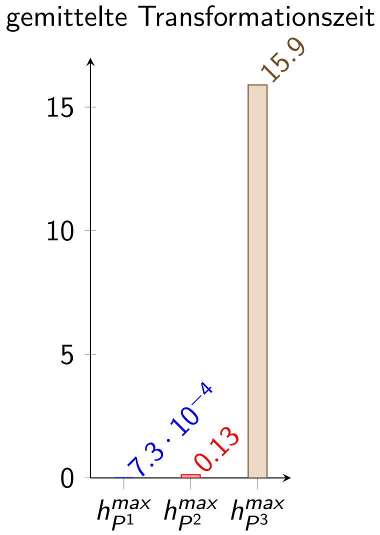
# Erfolgsrate

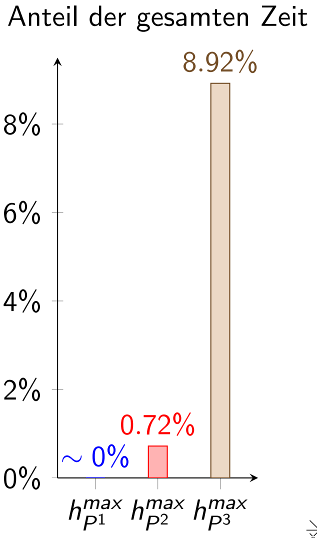


# Erfolgsrate und erfolgreiche Transformationen

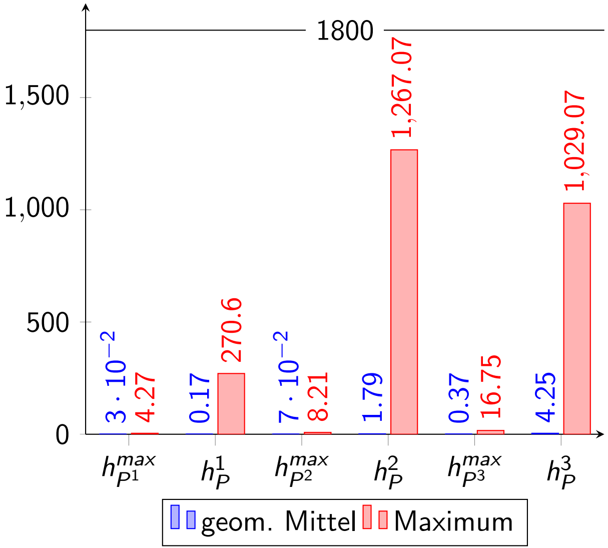


# Transformationszeit

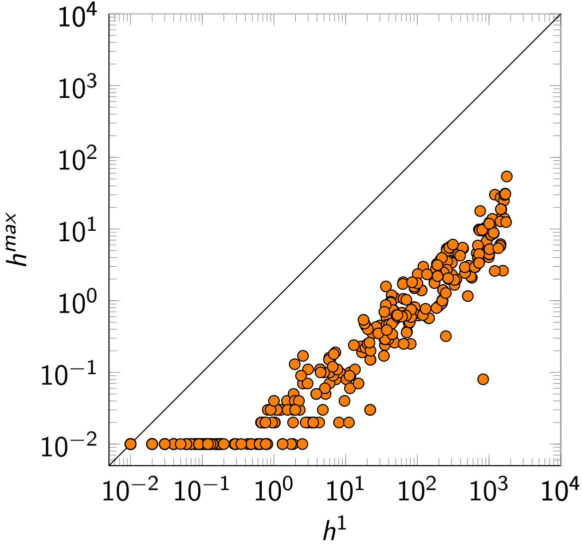


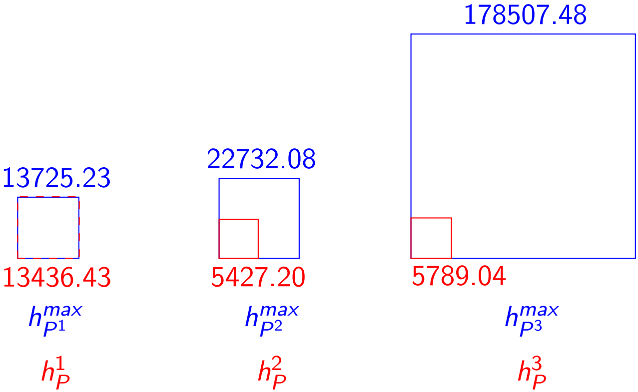
# Zeitbedarf


1800
1,500
1,000
500
0
3 . 10^−2
4.27
0.17
270.6
7 . 10^−2
8.21
1.79
1,267.07
0.37
16.75
4.25
1,029.07
$h^{max}_{P^1}$
$h^1_P$
$h^{max}_{P^2}$
$h^2_P$
$h^{max}_{P^3}$
$h^3_P$
geom. Mittel
Maximum

# $h^1$ vs $h^{max}$

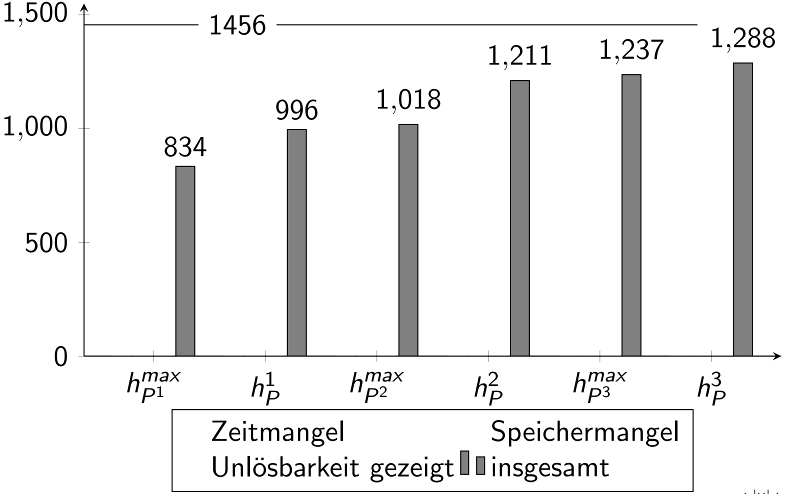


# Speicherbedarf

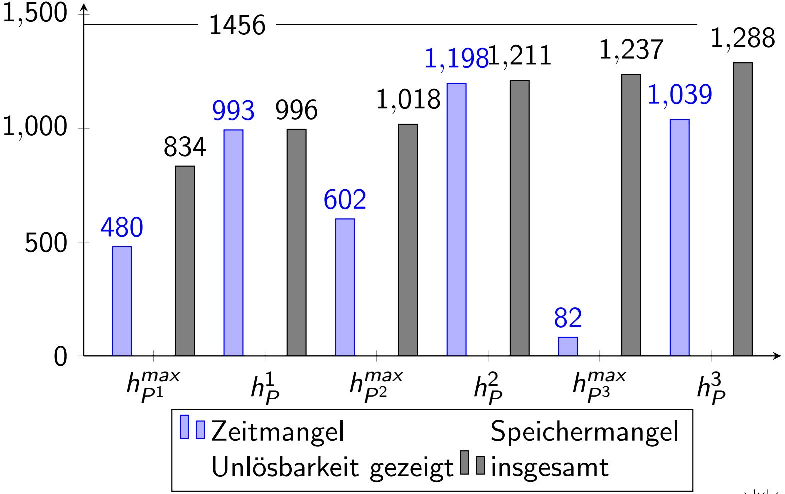



Universität Basel

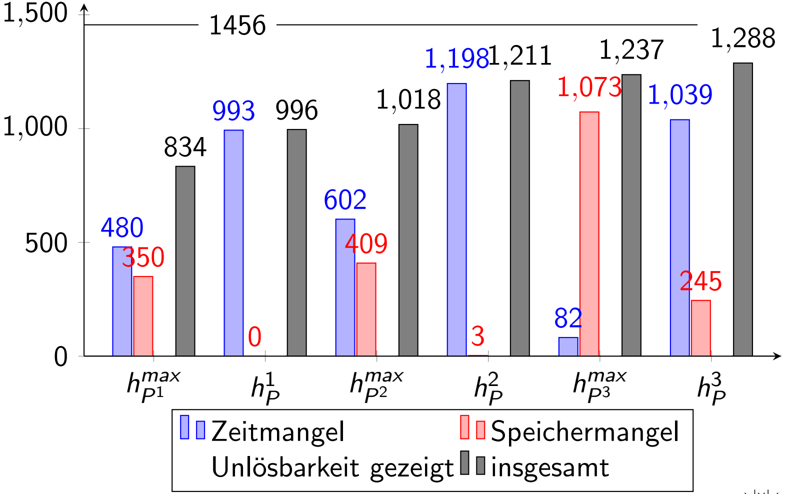
# Ursachen für ungelöste Instanzen



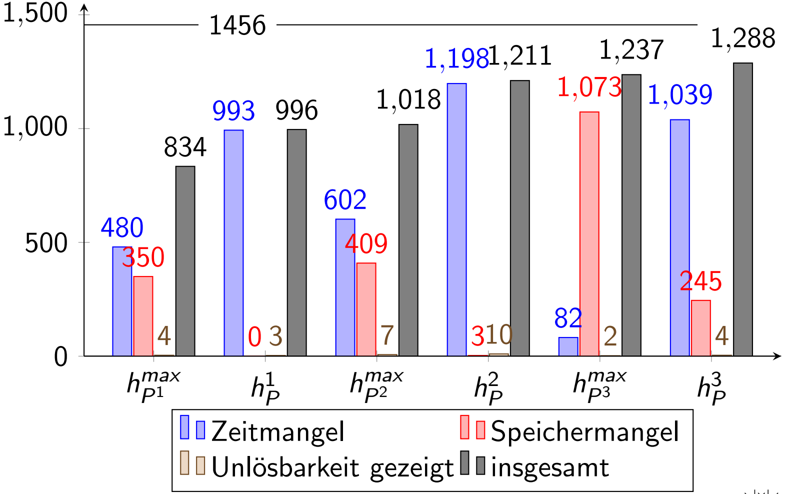
# Ursachen für ungelöste Instanzen



# Ursachen für ungelöste Instanzen



# Ursachen für ungelöste Instanzen



# Zusammenfassung

Grundsätzlich kleineres $m$ besser

| $h_P^m$ | Faktor | $h_{P^m}^{max}$ |
|---|---|---|
| $-$ | Erfolgsrate | $+$ |

Universität Basel

# Zusammenfassung

Grundsätzlich kleineres $m$ besser

| $h_P^m$ | Faktor | $h_{P^m}^{max}$ |
|---|---|---|
| − | Erfolgsrate | + |
| − | Zeit | + |

# Zusammenfassung

Grundsätzlich kleineres $m$ besser

| $h_P^m$ | Faktor | $h_{P^m}^{max}$ |
|---|---|---|
| − | Erfolgsrate | + |
| − | Zeit | + |
| + | Speicher | − |

# Fazit

Unter den getesteten Bedingungen kann mittels der $P^m$-Kompilierung eine bessere Performance erreicht werden.

Die Resultate sind abhängig von der Implementierung.

# Quellen

Patrik Haslum
$h^m(P) = h^1(P^m)$: Alternative Characterisations of the Generalisation From $h^{\max}$ To $h^m$.
*Proceedings of the 19th International Conference on Automated Planning and Scheduling*, 354–357, 2009.

# Mutex-Pruning

Der Agent kann nicht zugleich in Raum A und in Raum B sein.
Somit fallen im Beispiel folgende Inhalte weg:

$$\pi_{\{\text{inRaum(A), inRaum(B)}\}}$$

$$pre(\alpha_{a,\text{inRaum(Y)}}) = \{\pi_{\{\text{inRaum(X)}\}}, \pi_{\{\text{inRaum(Y)}\}}, \pi_{\{\text{inRaum(X), inRaum(Y)}\}}\}$$