Hybrid Planning: From Theory to Practice

Hybrides Planen – Von der Theorie zur Praxis (GI-Beitragstitel)

Pascal Bercher

Institut für Künstliche Intelligenz

Betreuerin: Prof. Dr. Susanne Biundo-Stephan

8. Mai 2018

Verteidigung: 4. Dezember 2017

ulm university universität ulm

Companion-Technologie

Heutige technische Geräte sind:

- komplex, haben breiten Funktionsumfang und
- erfordern Expertenwissen.

Companion-Technologie

Heutige technische Geräte sind:

- komplex, haben breiten Funktionsumfang und
- erfordern Expertenwissen.

Companion-Technologie

Vision:

Technische Geräte der Zukunft sind *Companion*-Systeme, d.h. sie sind kompetente Assistenten, die:

- vollautomatisch ihre Nutzer unterstützen,
- auf unvorhergesehene Eventualitäten reagieren und
- transparent agieren durch Erklärungsfunktionalität.

Companion-Technologie

Vision:

Technische Geräte der Zukunft sind *Companion*-Systeme, d.h. sie sind kompetente Assistenten, die:

- vollautomatisch ihre Nutzer unterstützen,
- auf unvorhergesehene Eventualitäten reagieren und
- transparent agieren durch Erklärungsfunktionalität.

Companion-Technologie

Vision:

Technische Geräte der Zukunft sind *Companion*-Systeme, d.h. sie sind kompetente Assistenten, die:

- vollautomatisch ihre Nutzer unterstützen,
- auf unvorhergesehene Eventualitäten reagieren und
- transparent agieren durch Erklärungsfunktionalität.

Fragestellungen

Die Doktorarbeit behandelt folgende Fragen:

- Welcher Formalismus bietet sich an?
- Welche theoretischen Eigenschaften hat der Formalismus und wie lassen sie sich ausnutzen?
- Wie können *gute* Hinweise und Instruktionen *schnell* generiert werden?
- Welche speziellen Herausforderungen stellen sich durch den „Anwendungskontext Mensch“?

Wir realisieren dies durch Techniken der Künstlichen Intelligenz: *Hierarchische* Handlungsplanung.

Fragestellungen

Die Doktorarbeit behandelt folgende Fragen:

- Welcher Formalismus bietet sich an?
- Welche theoretischen Eigenschaften hat der Formalismus und wie lassen sie sich ausnutzen?
- Wie können *gute* Hinweise und Instruktionen *schnell* generiert werden?
- Welche speziellen Herausforderungen stellen sich durch den „Anwendungskontext Mensch“?

Wir realisieren dies durch Techniken der Künstlichen Intelligenz: *Hierarchische* Handlungsplanung.

Fragestellungen

Die Doktorarbeit behandelt folgende Fragen:

- Welcher Formalismus bietet sich an?
- Welche theoretischen Eigenschaften hat der Formalismus und wie lassen sie sich ausnutzen?
- Wie können *gute* Hinweise und Instruktionen *schnell* generiert werden?
- Welche speziellen Herausforderungen stellen sich durch den „Anwendungskontext Mensch“?

Wir realisieren dies durch Techniken der Künstlichen Intelligenz: *Hierarchische* Handlungsplanung.

Fragestellungen

Die Doktorarbeit behandelt folgende Fragen:

- Welcher Formalismus bietet sich an?
- Welche theoretischen Eigenschaften hat der Formalismus und wie lassen sie sich ausnutzen?
- Wie können *gute* Hinweise und Instruktionen *schnell* generiert werden?
- Welche speziellen Herausforderungen stellen sich durch den „Anwendungskontext Mensch“?

Wir realisieren dies durch Techniken der Künstlichen Intelligenz: *Hierarchische* Handlungsplanung.

Fragestellungen

Die Doktorarbeit behandelt folgende Fragen:

- Welcher Formalismus bietet sich an?
- Welche theoretischen Eigenschaften hat der Formalismus und wie lassen sie sich ausnutzen?
- Wie können *gute* Hinweise und Instruktionen *schnell* generiert werden?
- Welche speziellen Herausforderungen stellen sich durch den „Anwendungskontext Mensch“?

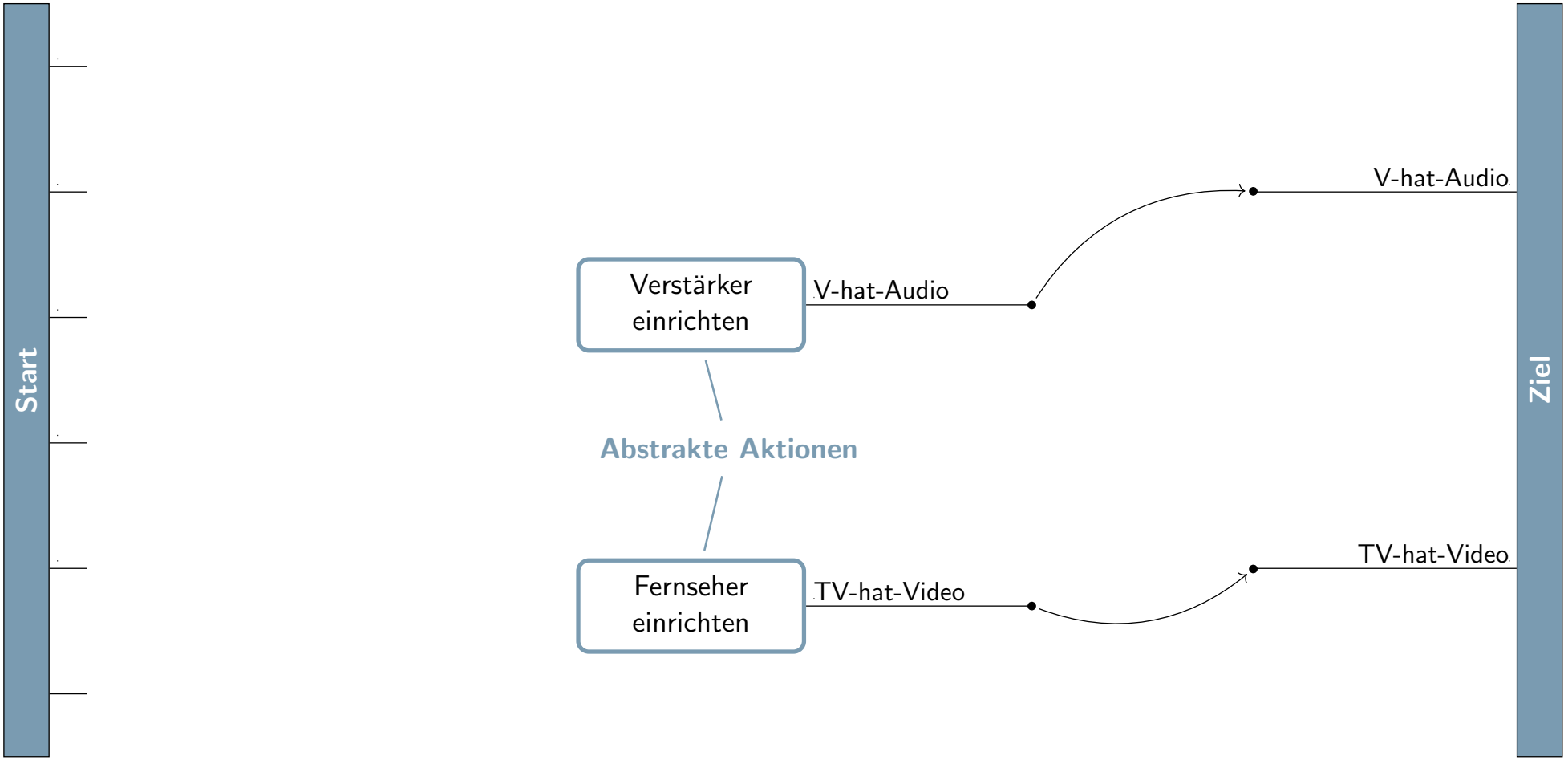
Wir realisieren dies durch Techniken der Künstlichen Intelligenz: *Hierarchische* Handlungsplanung.

Themen

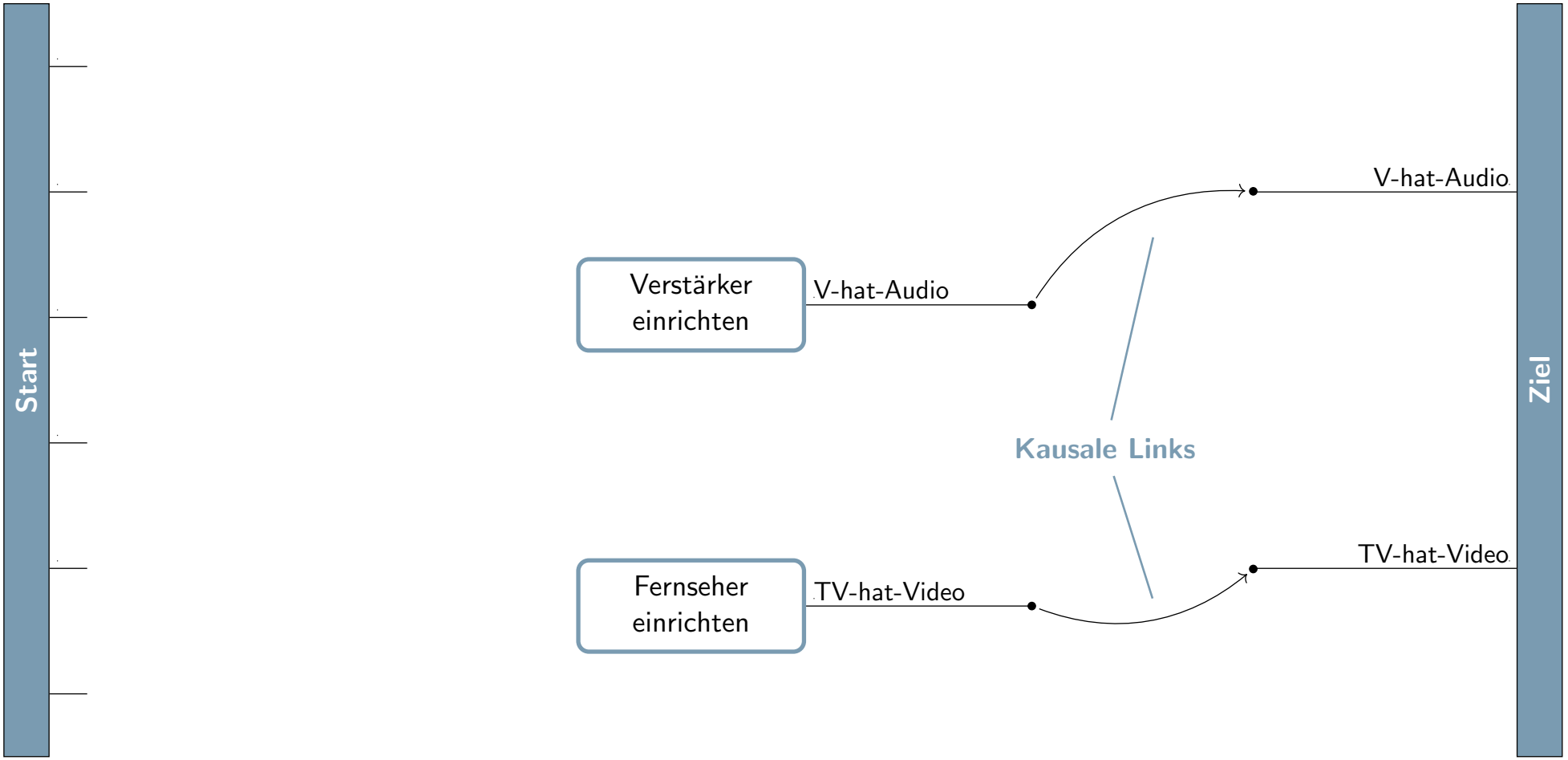
Hybrid Planning — From Theory to Practice

Hybrides Planen

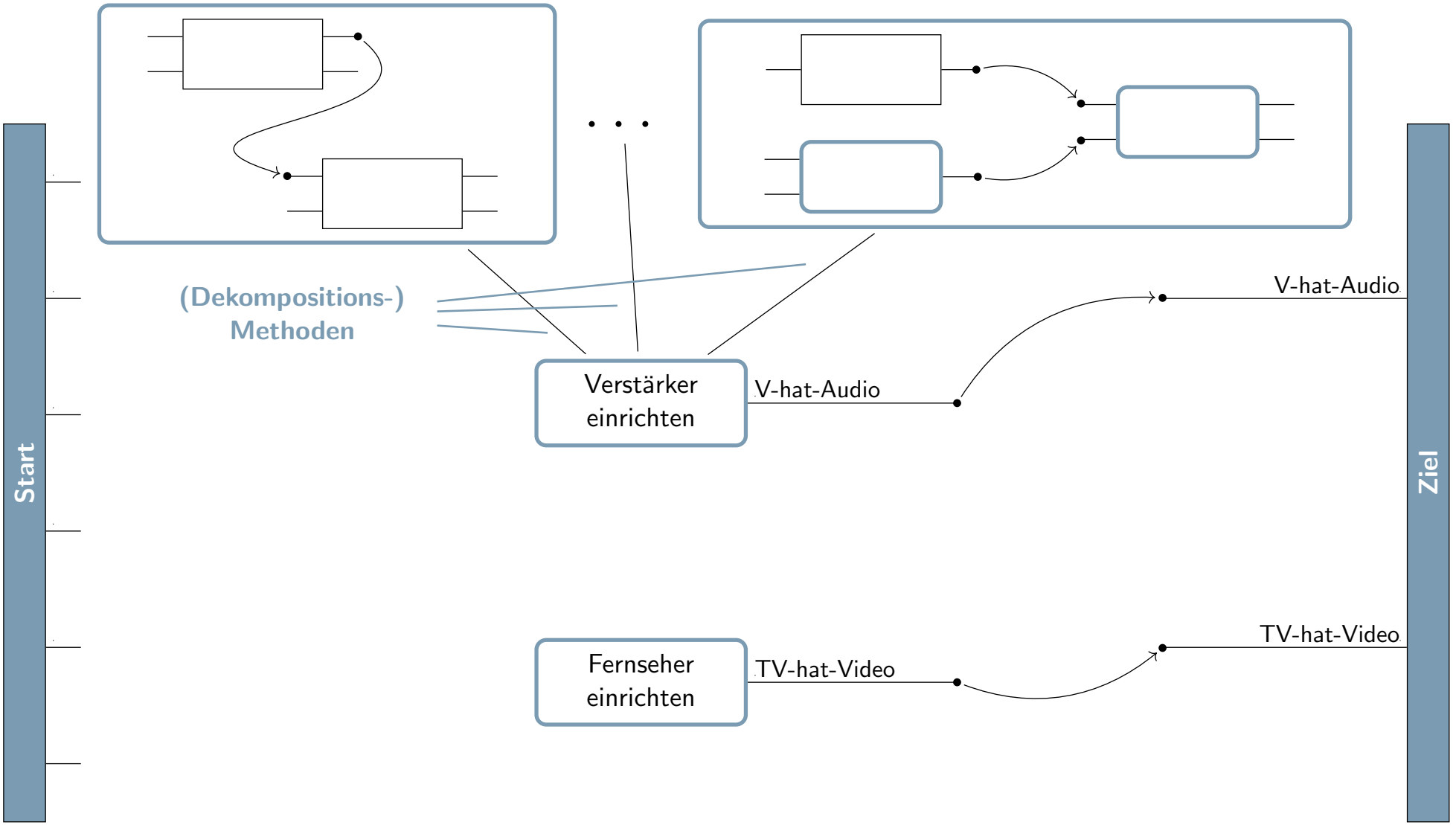
Hybrides Planen



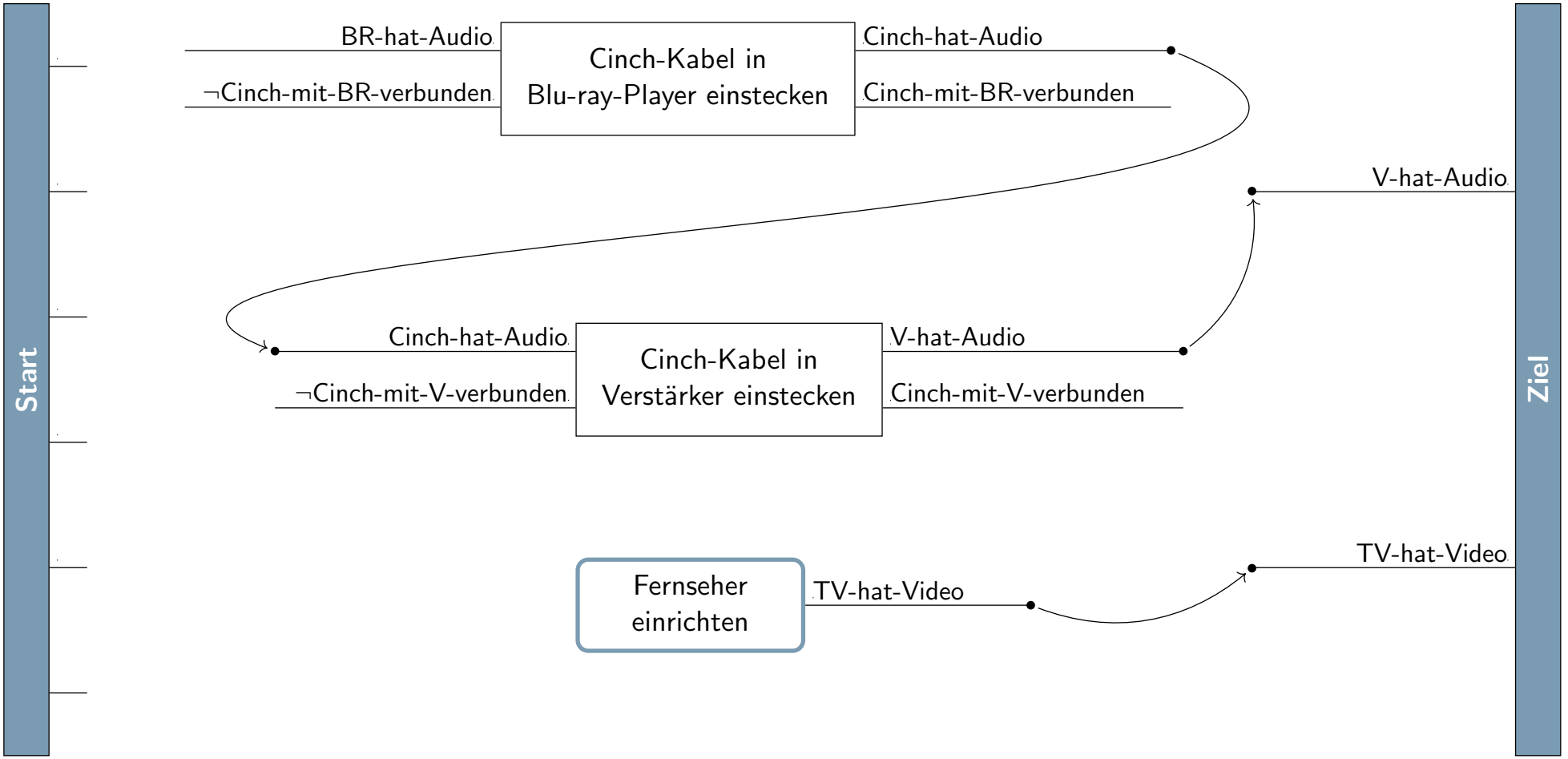
Hybrides Planen



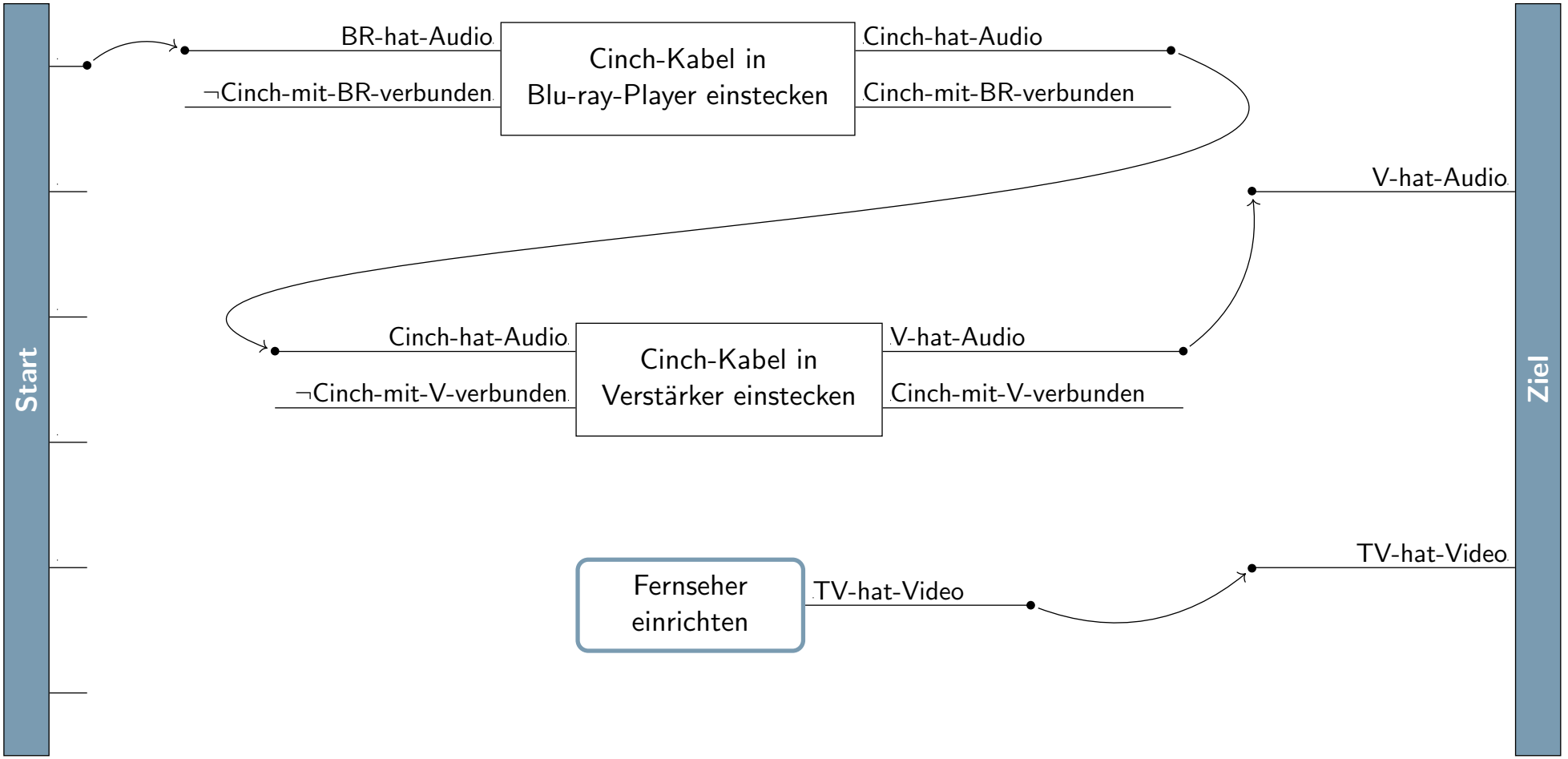
Hybrides Planen



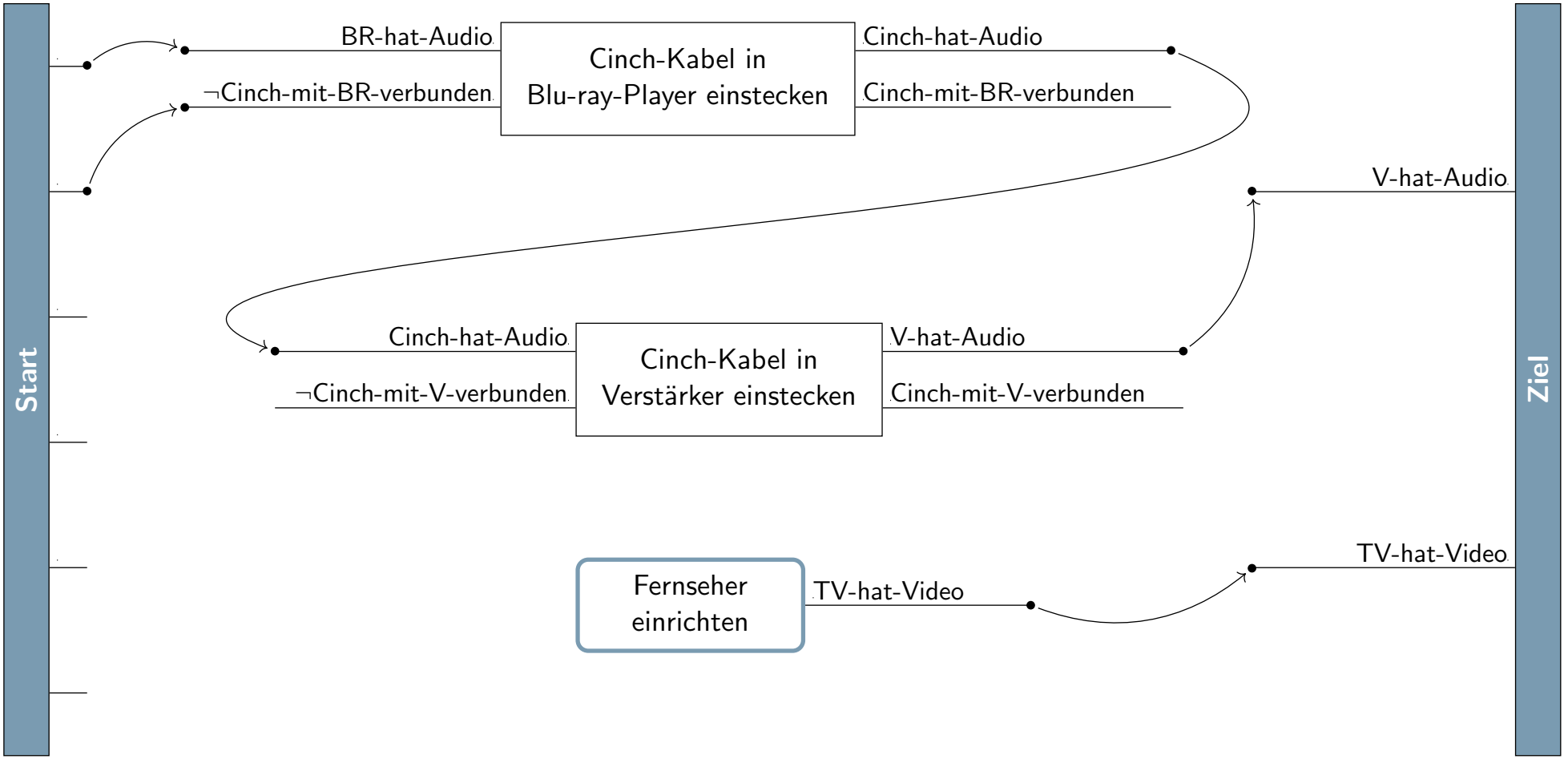
Hybrides Planen



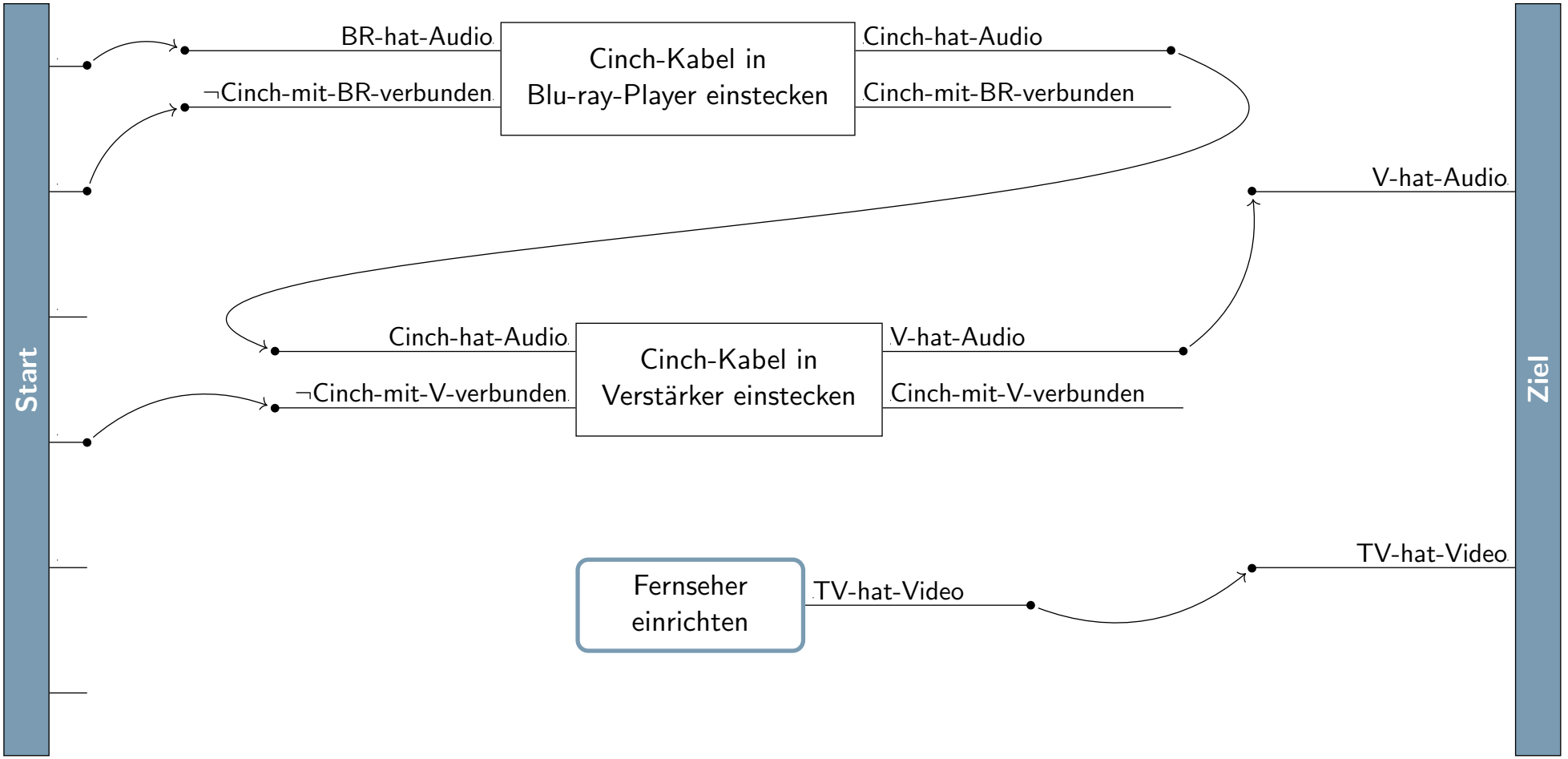
Hybrides Planen



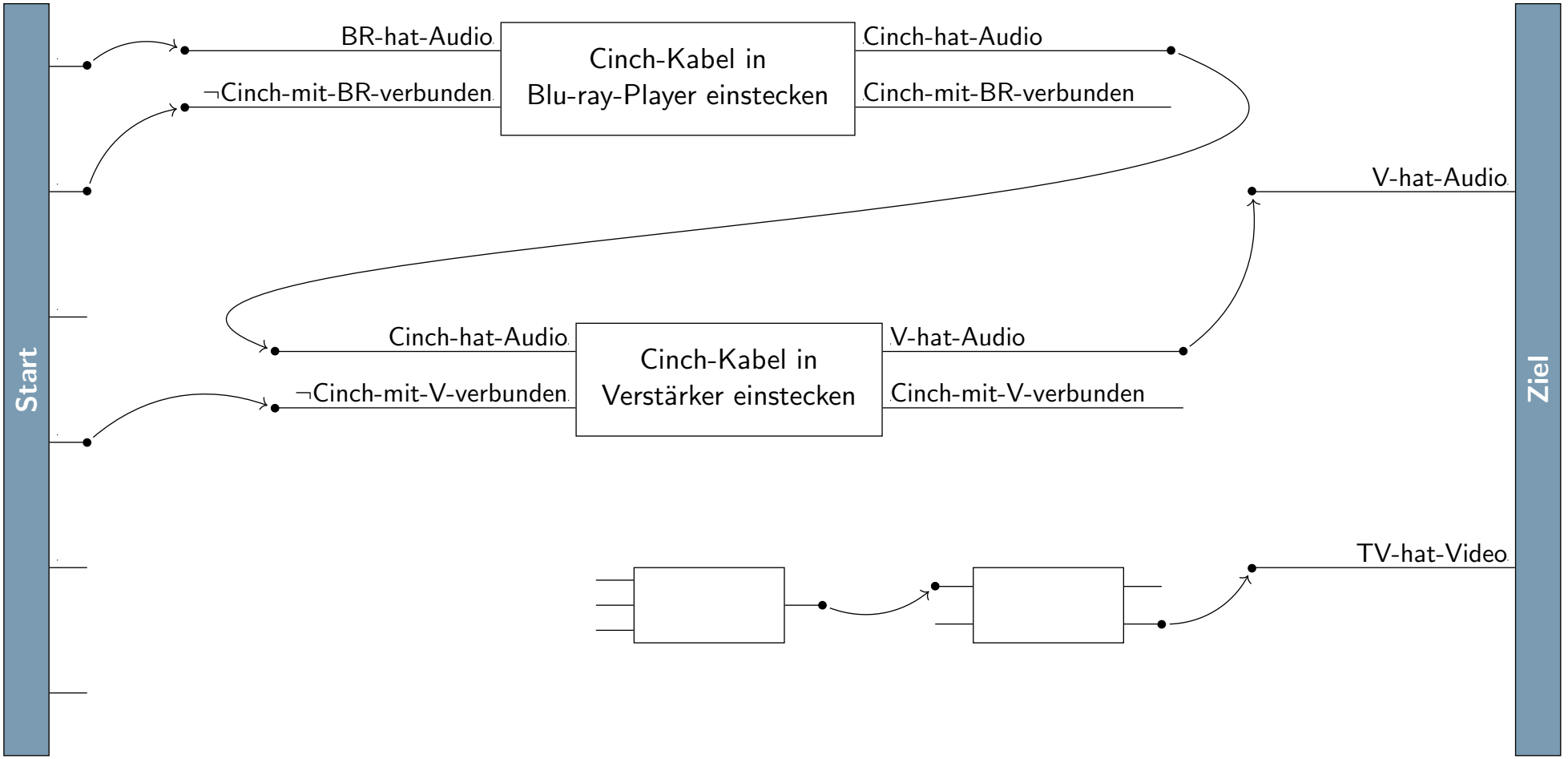
Hybrides Planen



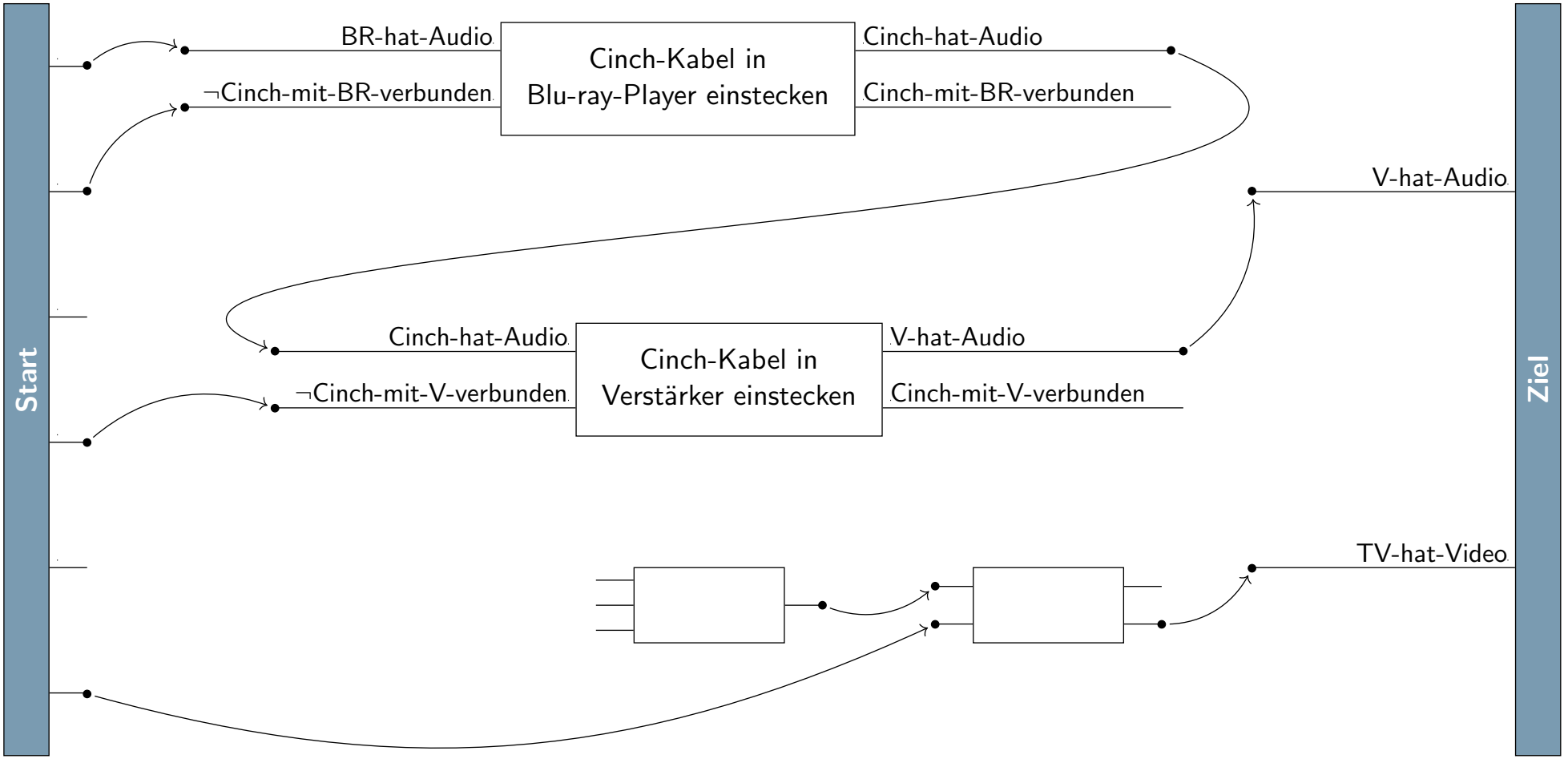
Hybrides Planen



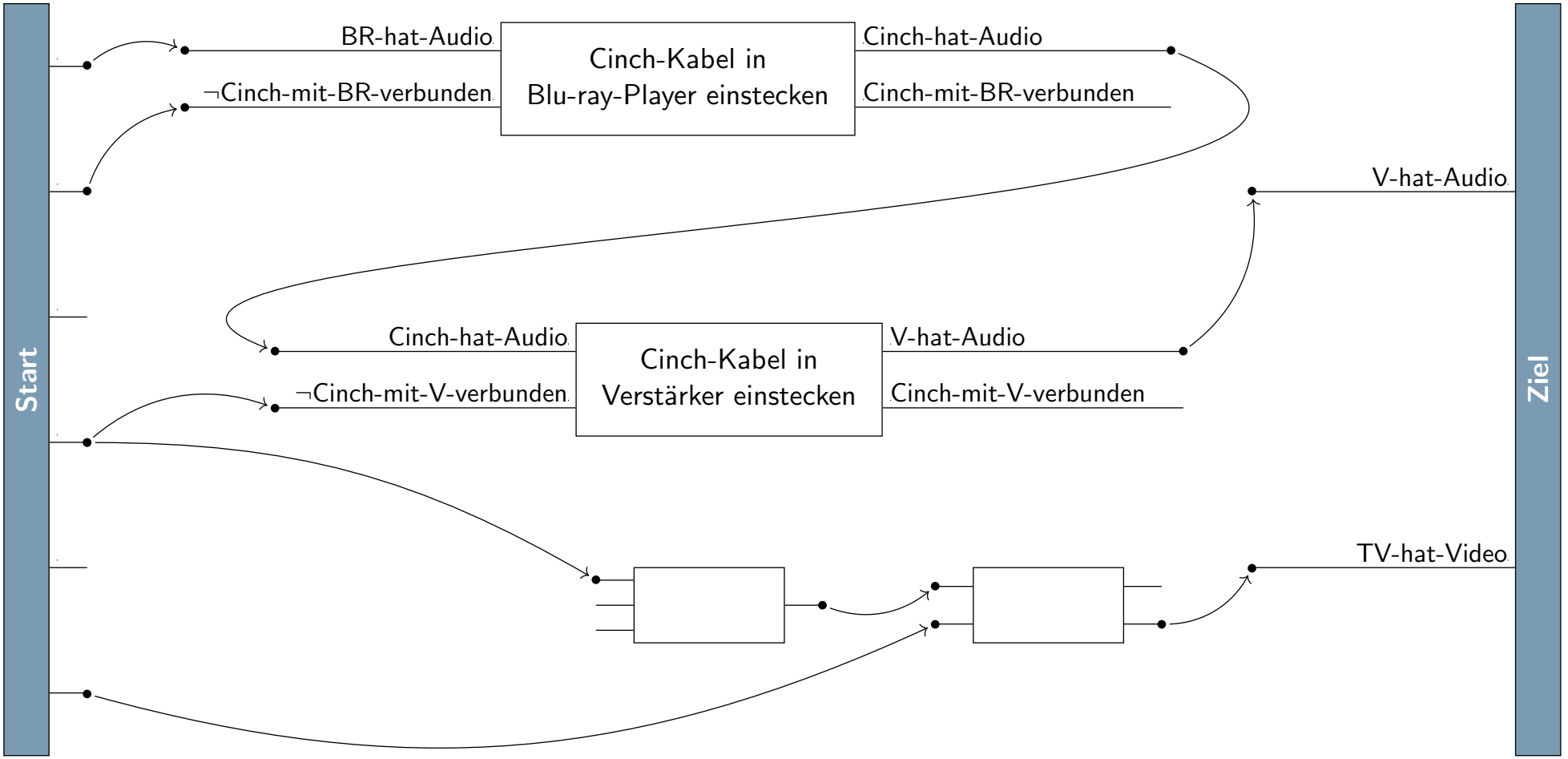
Hybrides Planen



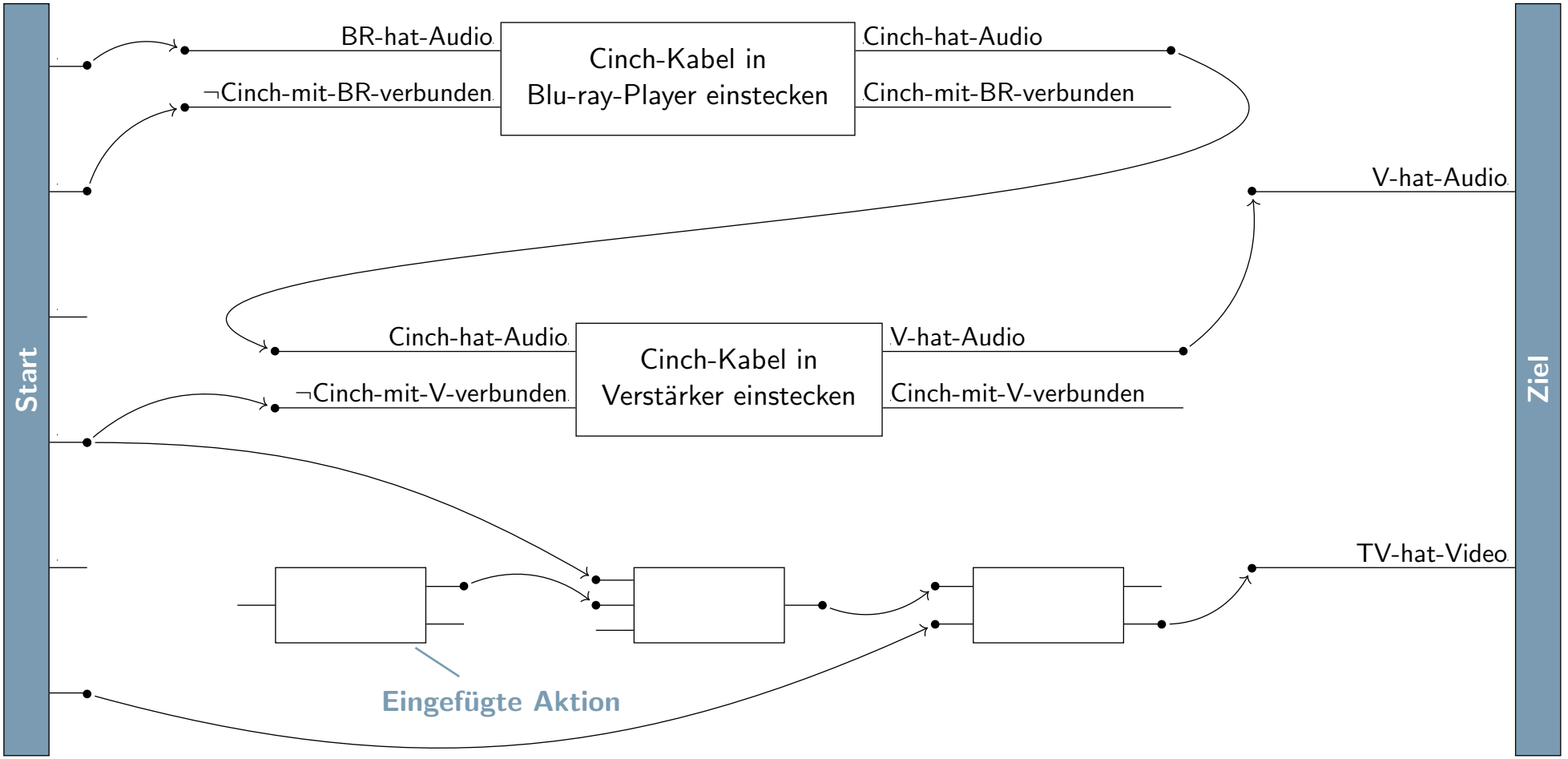
Hybrides Planen



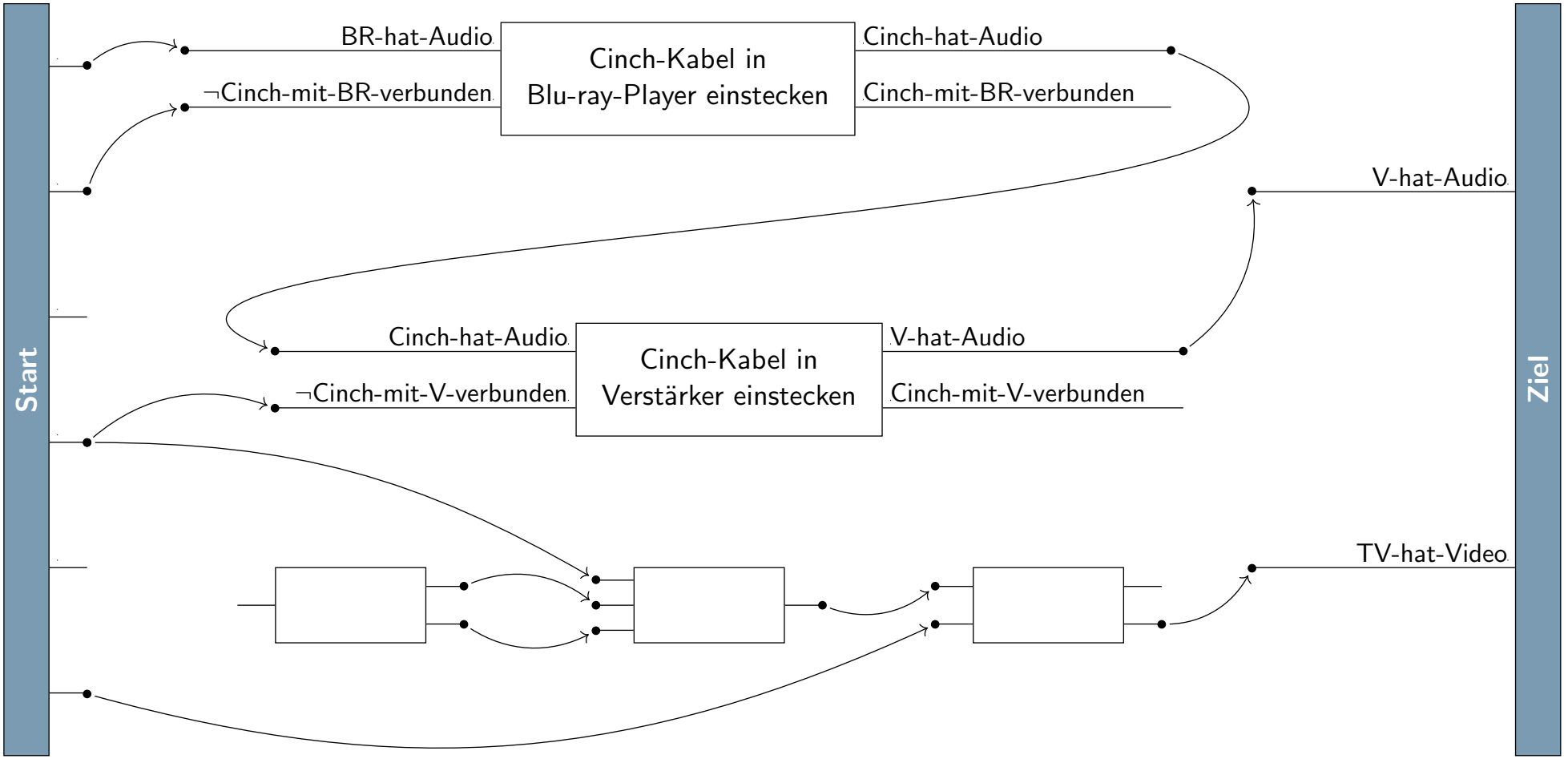
Hybrides Planen



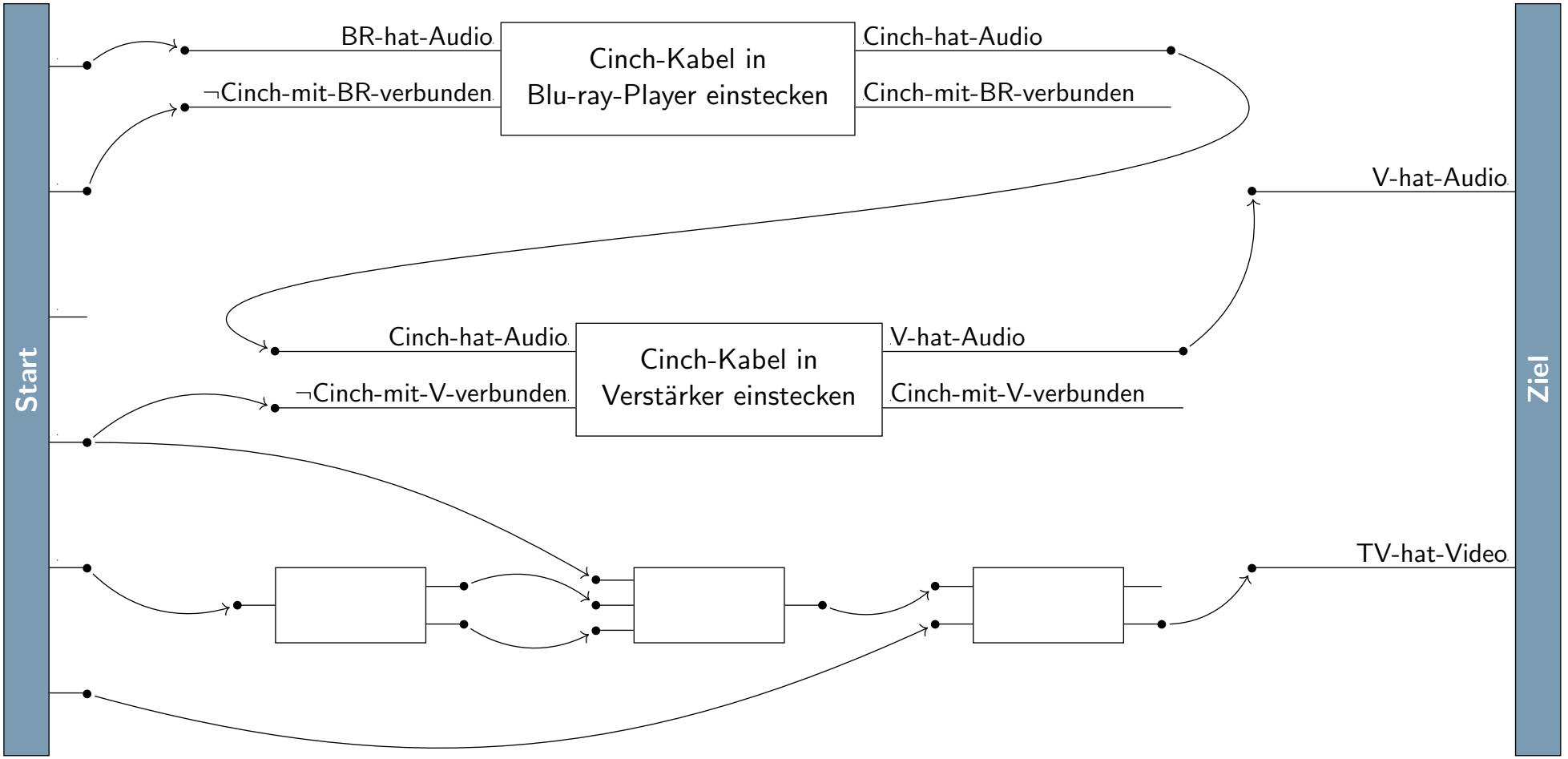
Hybrides Planen



Hybrides Planen



Hybrides Planen



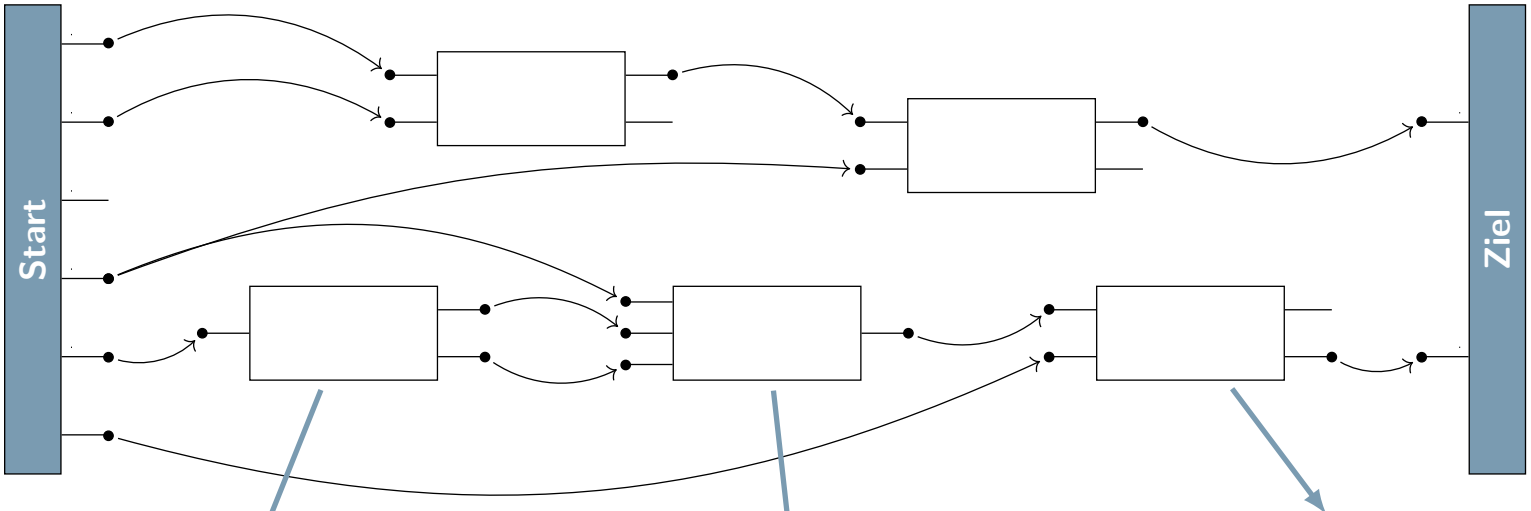
Hybrides Planen

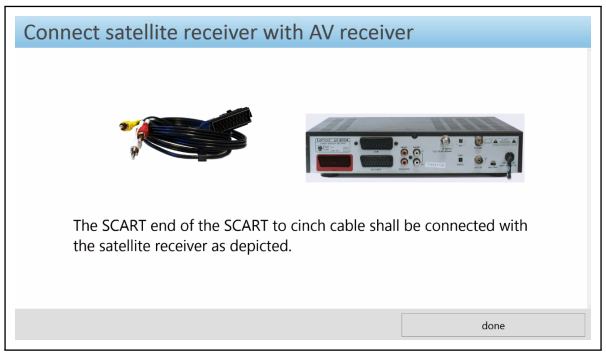
Zusammenfassung:

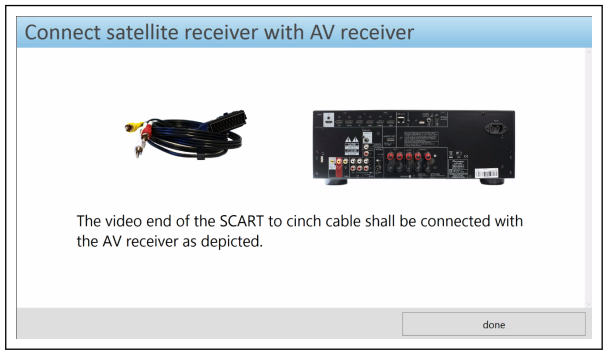
- Hybride Planungsprobleme sind spezifiziert durch:
  - Startzustand,
  - initiale abstrakte Aktionen,
  - Zielzustand.

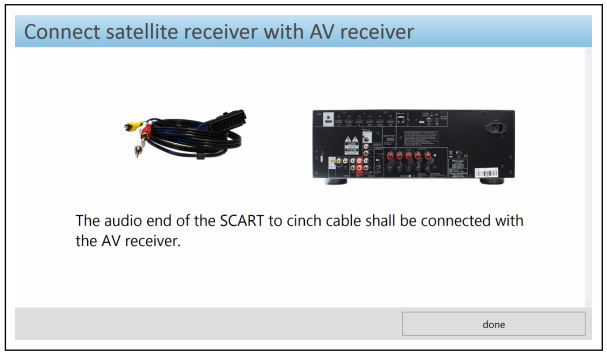
  } initialer Plan
- Lösungen haben folgende Eigenschaften:
  - wurden aus dem initialen Plan durch Verfeinerungen gewonnen,
  - sind partiell geordnet,
  - jede Linearisierung erfüllt die Ziele.
- Lösungen können als Grundlage von Assistenzsystemen verwendet werden.

Hybrides Planen

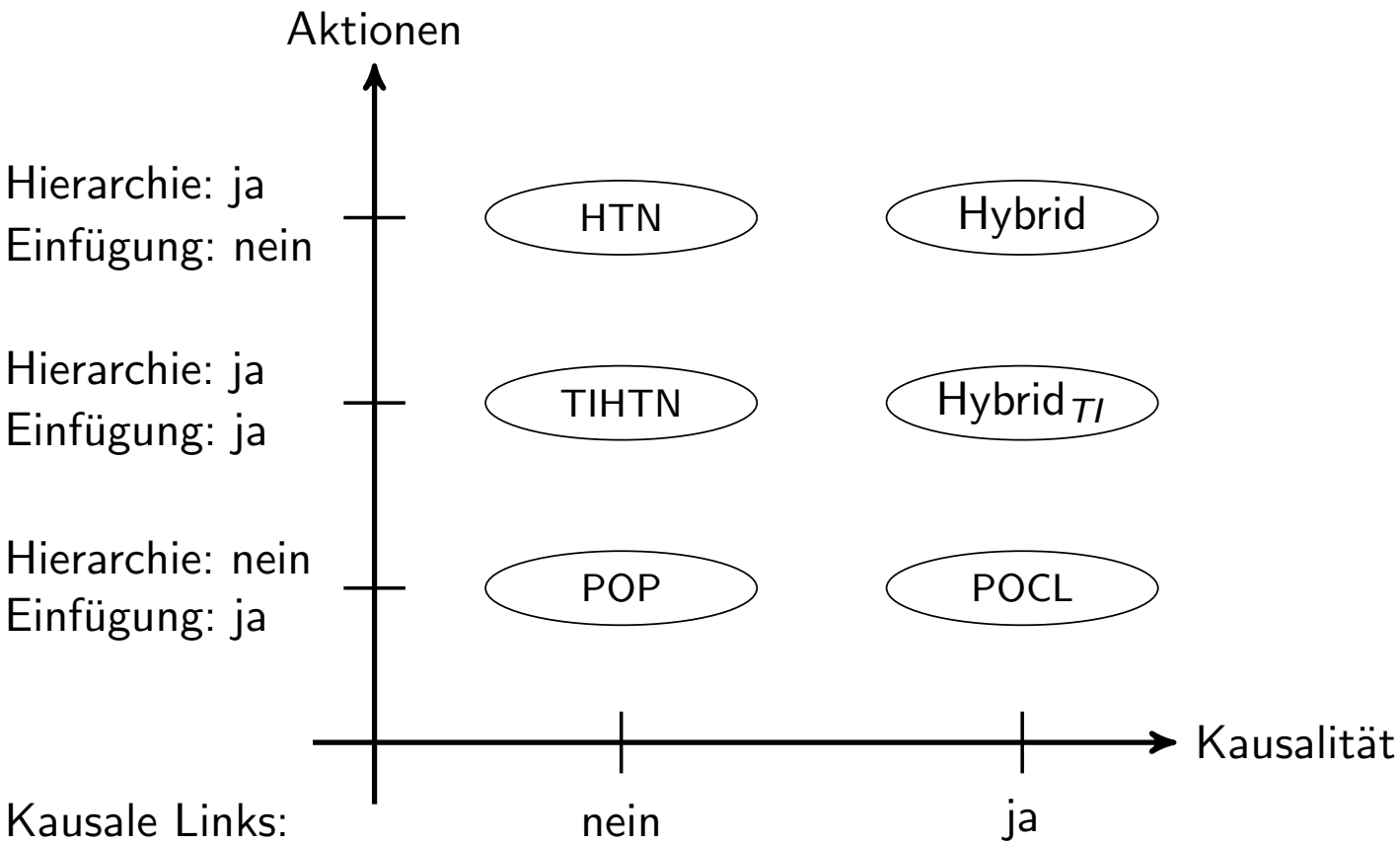




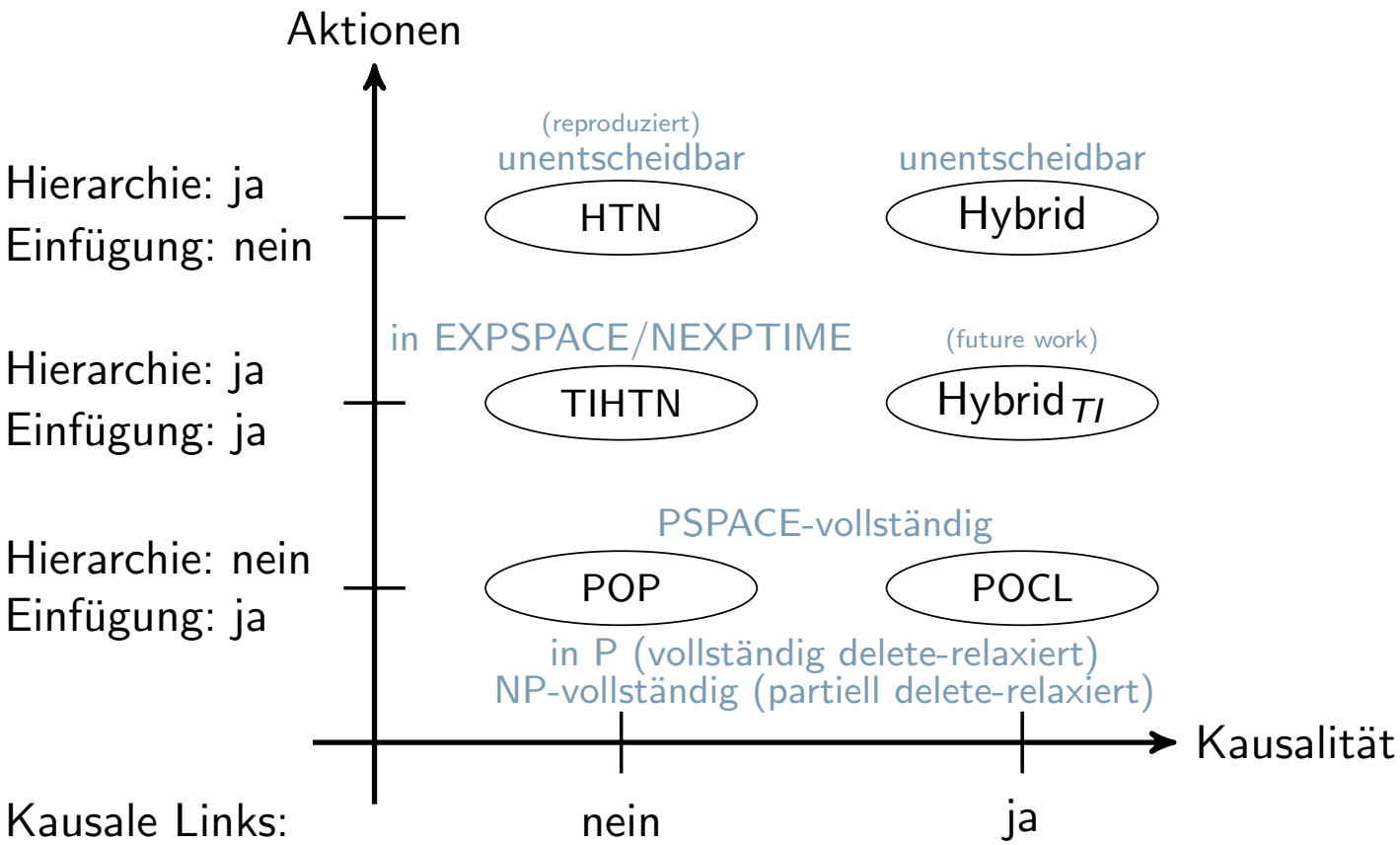




Kategorisierung der Problemklassen



Kategorisierung der Problemklassen

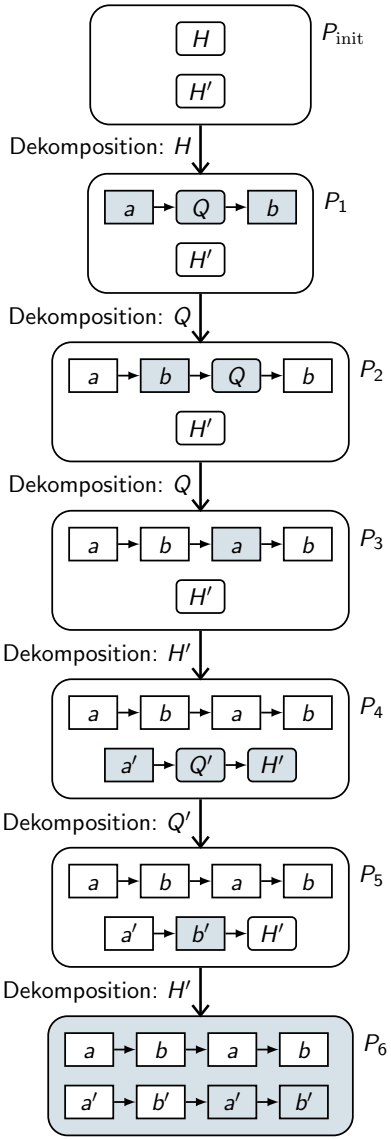


Das Planexistenzproblem für TIHTN-Probleme

Welchen Einfluss nimmt Task-Einfügung auf die Komplexität des Planexistenzproblems?

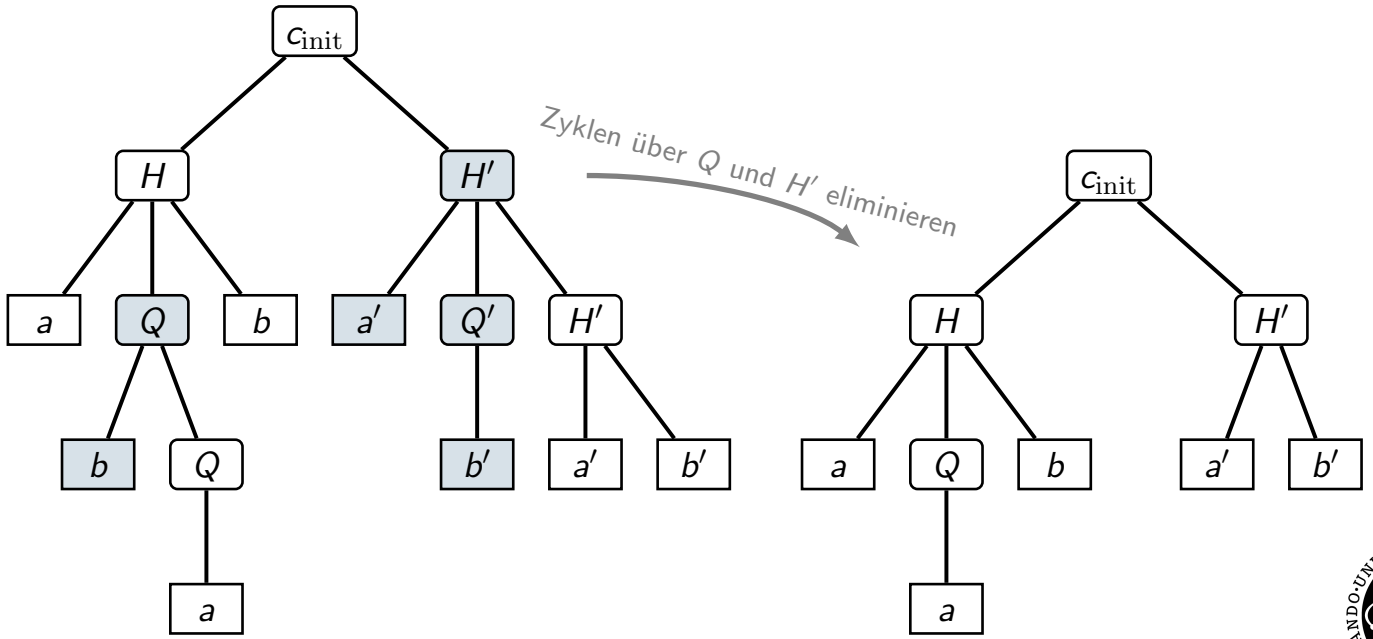
Theorem: Task-Einfügung macht das HTN-Problem entscheidbar

Das Planexistenzproblem für TIHTN-Probleme

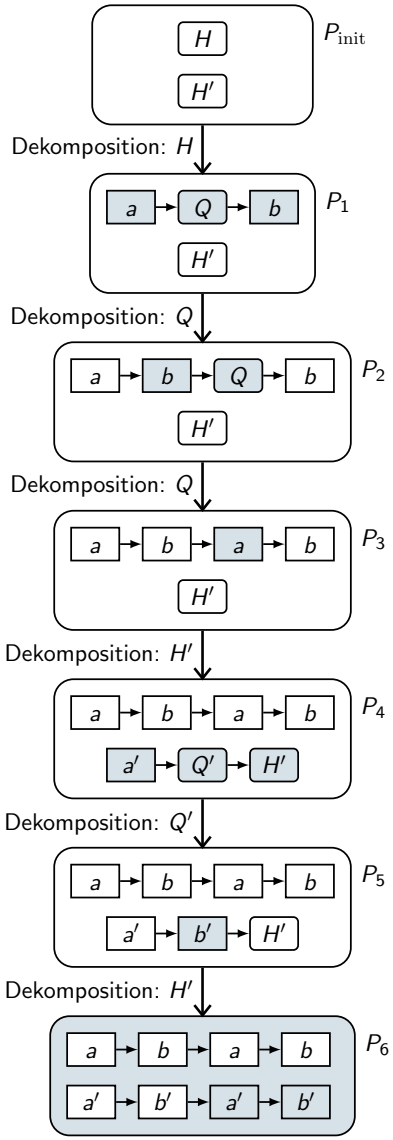


Theorem: TIHTN-Planen ist in **NEXPTIME**

Idee: Rate *azyklische* Dekompositionen und die einzufügenden Aktionen und prüfe Ausführbarkeit.



Das Planexistenzproblem für TIHTN-Probleme



Theorem: TIHTN-Planen ist in **NEXPTIME**

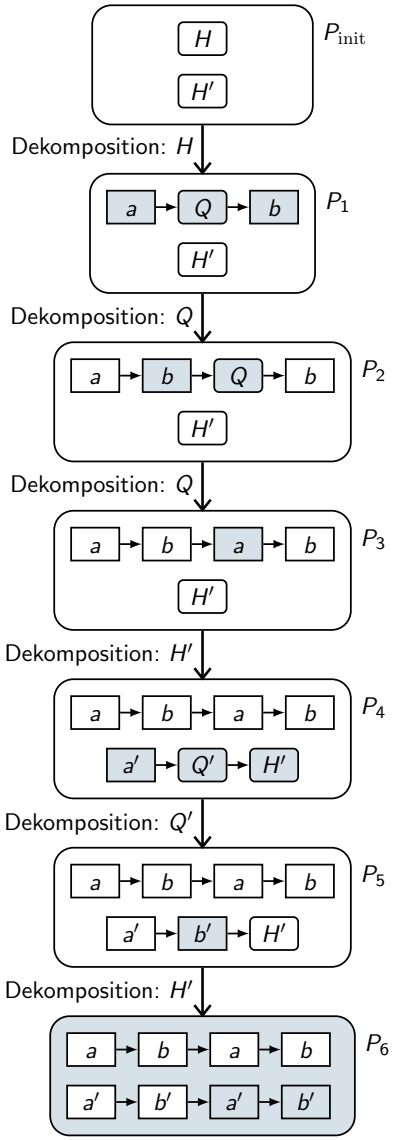
1. Schritt: Rate eine azyklische Dekomposition:

Der geratene Dekompositionsbaum beschreibt maximal $b^{|C|+1}$ Dekompositionen.

(C = Menge der abstrakten Tasks)
(b = Größe des größten Plans im Modell)

Prüfe in $O(b^{|C|+1})$, ob der Baum eine korrekte Folge von Dekompositionen beschreibt.

Das Planexistenzproblem für TIHTN-Probleme



Theorem: TIHTN-Planen ist in **NEXPTIME**

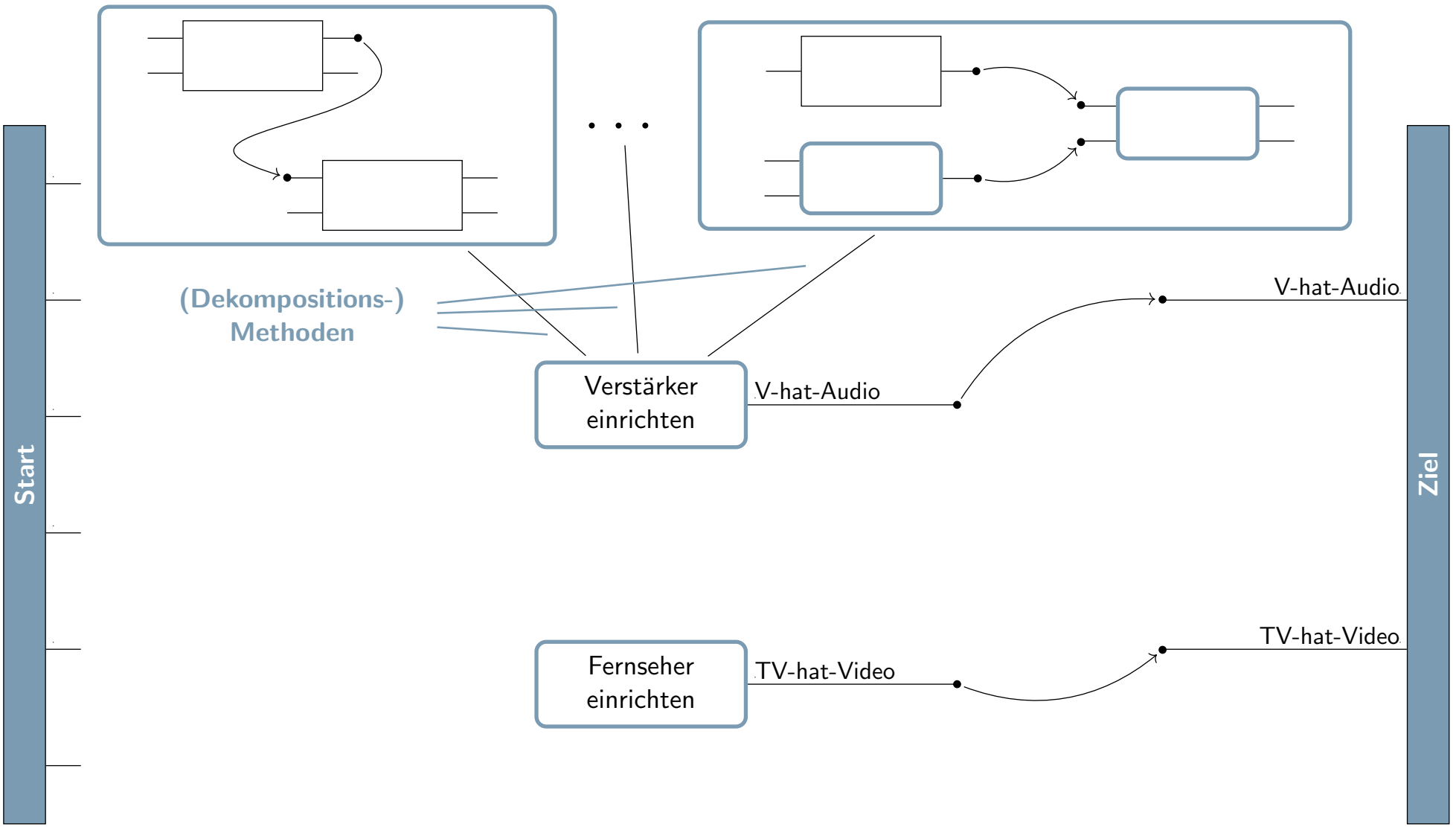
2. Schritt: Rate die einzufügenden Aktionen

Der (geratene) Dekompositionsbaum resultiert in einen Plan der Länge $\leq b^{|C|+1}$

Zwischen je zwei Aktionen müssen (nur) bis zu $2^{|V|}$ Aktionen eingefügt werden, um Ausführbarkeit zu erreichen.

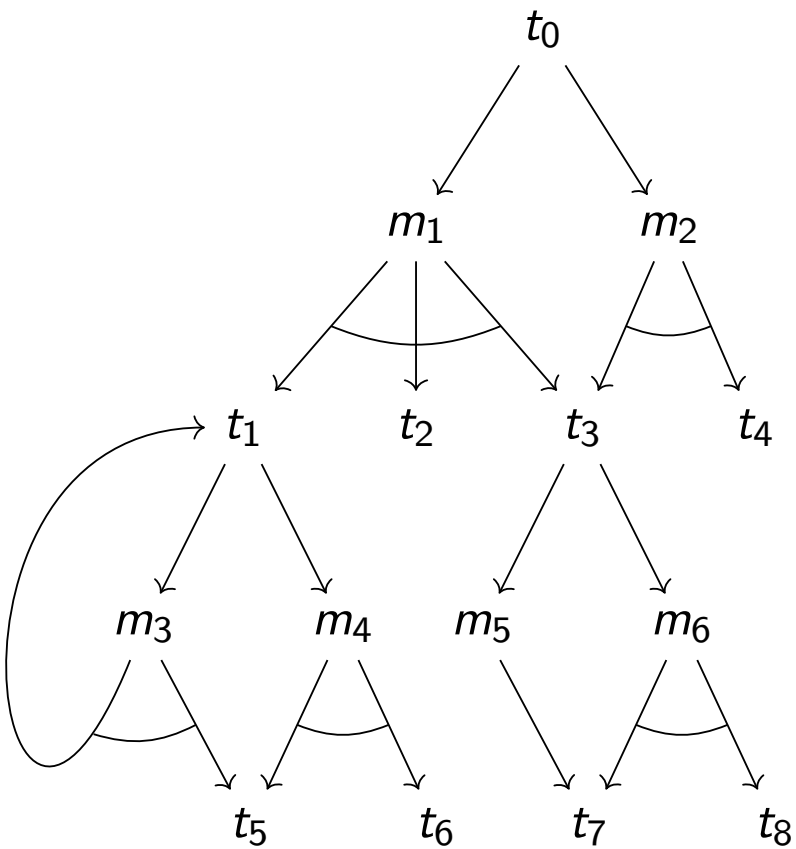
($|V|$ = Anzahl Zustandsvariablen)

Suchverfahren



TDG-basierte Heuristik

Der Task-Dekompositions-Graph (TDG) repräsentiert die Dekompositionsstruktur:



Nutzung des TDGs zur Berechnung von Heuristiken.

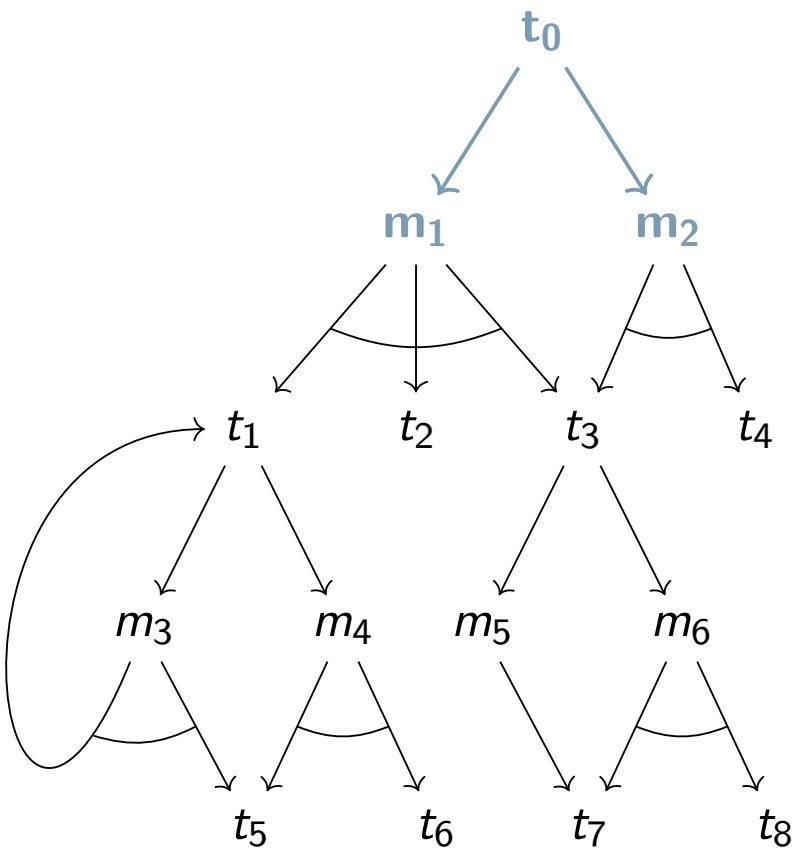
Schritt 1:
Berechnung des TDGs.

Schritt 2:
Berechne Heuristik $h(t)$ für jeden Task t im TDG (Preprocessing).

Schritt 3:
Für Suchknoten (Plan) P und seine Taskmenge T, berechne $h(P) := \sum_{t \in T} h(t)$.

TDG-basierte Heuristik

Der Task-Dekompositions-Graph (TDG) repräsentiert die Dekompositionsstruktur:

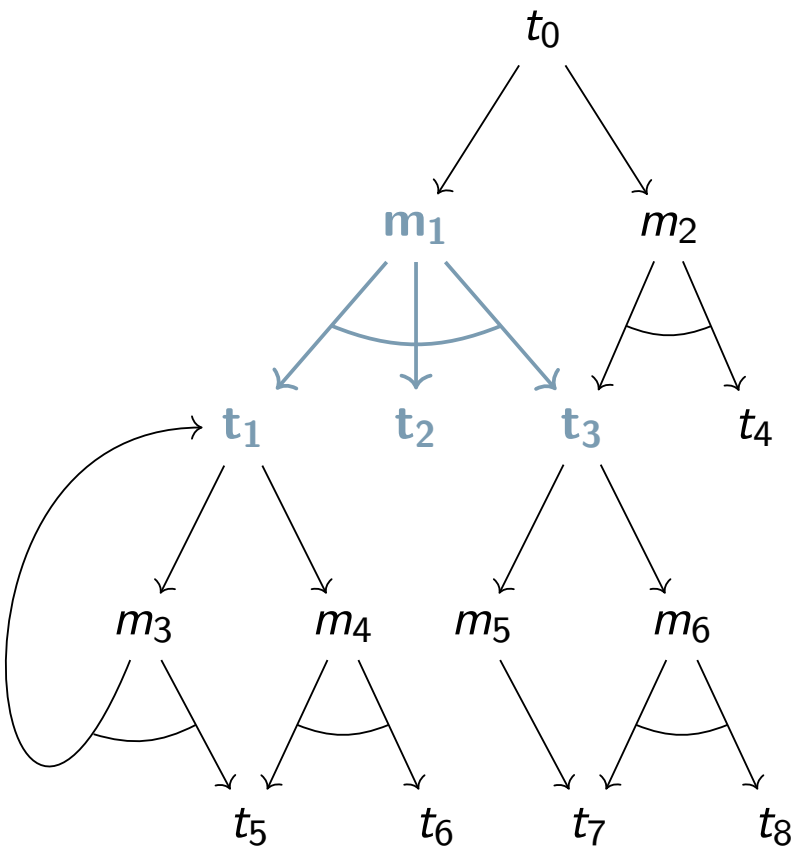


Beispiel:

$$h(t_0) = 1 + min\{h(m_1), h(m_2)\}$$

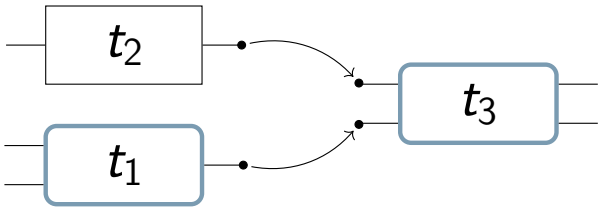
TDG-basierte Heuristik

Der Task-Dekompositions-Graph (TDG) repräsentiert die Dekompositionsstruktur:



Beispiel:

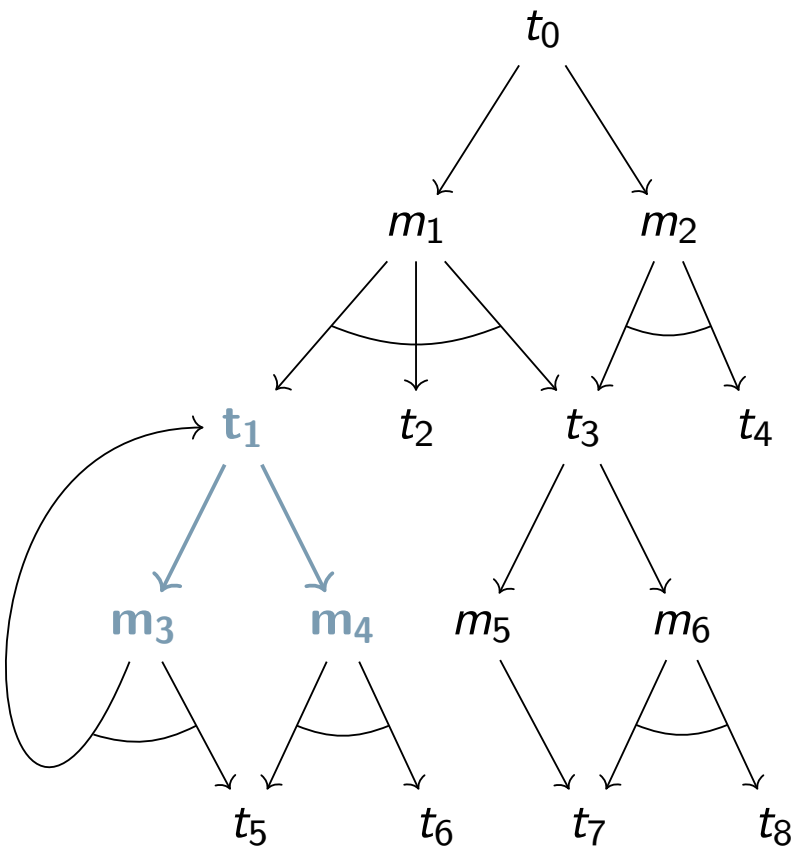
Methode $m_1 = (t_0, P)$ mit Plan P:



$$h(m_1) = \sum_{t_i \in \{t_1, t_2, t_3\}} h(t_i) - |CL|$$

TDG-basierte Heuristik

Der Task-Dekompositions-Graph (TDG) repräsentiert die Dekompositionsstruktur:

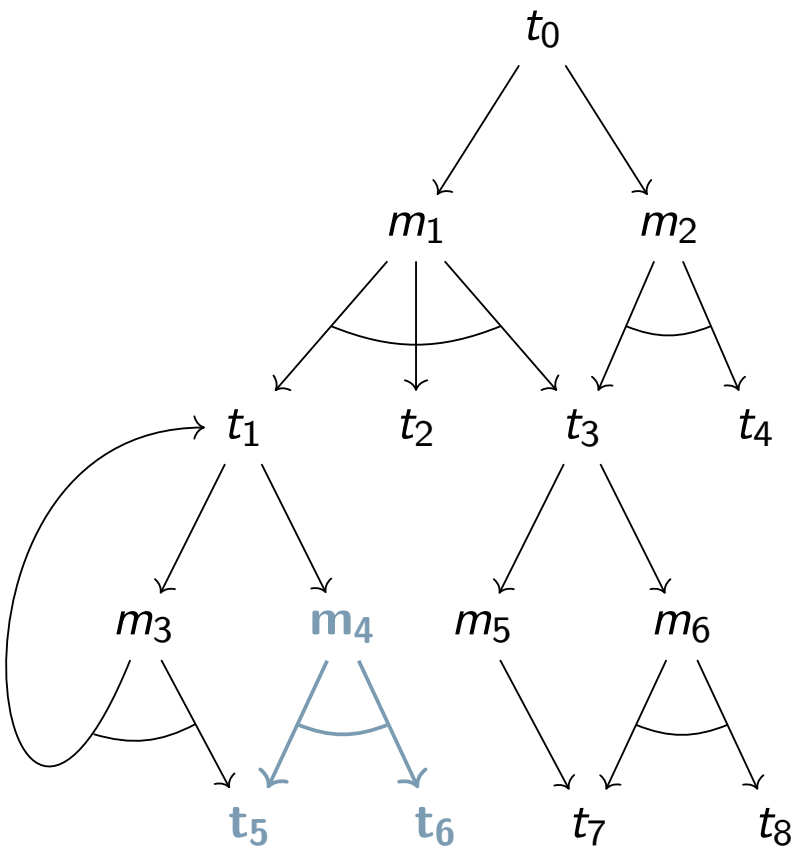


Beispiel:

$$h(t_1) = 1 + min\{h(m_3), h(m_4)\}$$

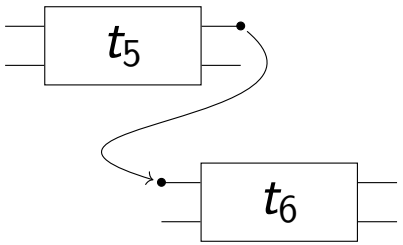
TDG-basierte Heuristik

Der Task-Dekompositions-Graph (TDG) repräsentiert die Dekompositionsstruktur:



Beispiel:

Methode $m_4 = (t_1, P)$ mit Plan P:



$$\begin{aligned} h(m_4) &= h(t_5) + h(t_6) - |CL| \\ &= |pre(t_5)| + |pre(t_6)| - 1 \\ &= 2 + 2 - 1 = 3 \end{aligned}$$

TDG-basierte Heuristik: Empirische Ergebnisse

- Vergleich mit anderen Heuristiken und Suchansätzen
- Zwei Varianten der vorgestellten Heuristik
  - darunter die *erste zulässige* $\rightarrow$ garantiert Finden optimaler Lösungen
  - nicht-zulässige Heuristik zeigt die besten Ergebnisse (Anzahl gelöster Probleminstanzen)

Entwicklung eines Verkabelungsassistenten

Die Herausforderung:

Entwicklung eines Verkabelungsassistenten

Unser Assistent besitzt folgende nutzerzentrierte Fähigkeiten:

- *Schritt-für-Schritt-Anleitung:*
  Automatische Assistenz basierend auf Modell der Hardware.
- *Planlinearisierung:*
  Anordnung der Anleitungsschritte in nutzerfreundlicher Weise.
- *Ausführungsüberwachung und Reparatur:*
  Behandlung von Ausführungsfehlern durch Anpassung des Plans.
- *Planerklärung:*
  Begründung der Notwendigkeit von hinterfragten Anleitungsschritten.

Entwicklung eines Verkabelungsassistenten

Beispielinteraktion mit dem System:

Zusammenfassung der Hauptbeiträge

Theoretische Grundlagen:

- Neue Formalisierung für HTN-Planen (wurde ein Standard)
- Erstmalige Komplexitätsergebnisse für *Task-Einfügung*, *hybrides Planen*, *nicht-lineares nicht-hierarchisches Planen*.

Suche und Heuristiken:

- Entwicklung des Task-Dekompositions-Graphen (TDGs).
- Erste informative, *zulässige* Ziel-Distanz-Schätzung für HTN- und hybride Planungsprobleme zum Finden *optimaler* Pläne.

Praktische Anwendung:

- Realisierung eines Assistenten zum Aufbau einer Heimkino-anlage basierend auf *hierarchischer Plangenerierung*, *nutzerfreundlicher Planlinearisierung*, *Planreparatur* und *Planerklärung*.

Zusammenfassung der Hauptbeiträge

Theoretische Grundlagen:

- Neue Formalisierung für HTN-Planen (wurde ein Standard)
- Erstmalige Komplexitätsergebnisse für *Task-Einfügung*, *hybrides Planen*, *nicht-lineares nicht-hierarchisches Planen*.

Suche und Heuristiken:

- Entwicklung des Task-Dekompositions-Graphen (TDGs).
- Erste informative, *zulässige* Ziel-Distanz-Schätzung für HTN- und hybride Planungsprobleme zum Finden *optimaler* Pläne.

Praktische Anwendung:

- Realisierung eines Assistenten zum Aufbau einer Heimkino-anlage basierend auf *hierarchischer Plangenerierung*, *nutzerfreundlicher Planlinearisierung*, *Planreparatur* und *Planerklärung*.

Zusammenfassung der Hauptbeiträge

Theoretische Grundlagen:

- Neue Formalisierung für HTN-Planen (wurde ein Standard)
- Erstmalige Komplexitätsergebnisse für *Task-Einfügung*, *hybrides Planen*, *nicht-lineares nicht-hierarchisches Planen*.

Suche und Heuristiken:

- Entwicklung des Task-Dekompositions-Graphen (TDGs).
- Erste informative, *zulässige* Ziel-Distanz-Schätzung für HTN- und hybride Planungsprobleme zum Finden *optimaler* Pläne.

Praktische Anwendung:

- Realisierung eines Assistenten zum Aufbau einer Heimkino-anlage basierend auf *hierarchischer Plangenerierung*, *nutzerfreundlicher Planlinearisierung*, *Planreparatur* und *Planerklärung*.

Hauptbeiträge

Hybrid Planning — From Theory to Practice

1. Theoretische Grundlagen
   Hauptbeiträge: 3 Publikationen
   weitere Beiträge: 5 Publikationen

2. Suche und Heuristiken
   Hauptbeiträge: 5 Publikationen
   weitere Beiträge: 12 Publikationen

3. Praktische Anwendung
   Hauptbeiträge: 5 Publikationen
   weitere Beiträge: 16 Publikationen

Hauptbeiträge: Theoretische Grundlagen

Zusammenfassung:

- Erste Formalisierung und theoretische Untersuchung der Task-Einfügung (Planexistenz)

Auch in der Dissertation behandelt:

- Neue Formalisierung für HTN-Planen (wurde ein Standard)
- Neue Formalisierung für hybrides Planen und Übertragung der HTN-Komplexitätsergebnisse
- Systematische Untersuchung von Legalitätseigenschaften hybrider Planungsmodelle
- Erste theoretische Untersuchungen (Planexistenz) für POP- und POCL-Planungsprobleme

Hauptbeiträge: Suche und Heuristiken

Zusammenfassung:

- Entwicklung des Task-Dekompositions-Graphen (TDGs).
- Erste informative, *zulässige* Ziel-Distanz-Schätzung für HTN- und hybride Planungsprobleme zum Finden optimaler Pläne.

Auch in der Dissertation behandelt:

- Weiterentwicklung eines planraumbasierten Ansatzes für hybrides Planen.
- (Mit-)Entwicklung von Suchstrategien basierend auf Landmarken.
- SampleFF-Heuristik für POP- & POCL-Planen.
- Generischer Ansatz, der zustandsbasierte Heuristiken für POP- & POCL-Heuristiken zugänglich macht.

Hauptbeiträge: Praktische Anwendung

Zusammenfassung:

- Realisierung eines Assistenten zum Aufbau einer Heimkinoanlage basierend auf:
  - hierarchischer Plangenerierung,
  - nutzerfreundliche Planlinearisierung,
  - Planreparatur und
  - Planerklärung.

Auch in der Dissertation behandelt:

- Empirische Evaluation von Planerklärungen im Kontext des Assistenten mit 59 Probanden.

Formale Problembeschreibung, (TI)HTN-Domäne

Definition (HTN/TIHTN-Planungsdomäne, grundiert)

Eine *Planungsdomäne* $\mathcal{D} = (V, N_C, N_P, \gamma, M)$ besteht aus:

- V, endlich vielen *Zustandsvariablen*,
- N_C, endlich vielen *abstrakten Aktionsnamen*,
- N_P, endlich vielen *primitiven Aktionsnamen*,
- $\gamma : N_P \to A$, bijektive Funktion, wobei A die *Aktionen* sind,
- $M \subseteq N_C \times P_{N_C \cup N_P}$, endlich vielen *Methoden*.

Definition (Zustand)

Ein *Zustand* ist eine Menge $s \in 2^V$.

Formale Problembeschreibung, (TI)HTN-Domäne

Definition (HTN/TIHTN-Planungsdomäne, grundiert)

Eine *Planungsdomäne* $\mathcal{D} = (V, N_C, N_P, \gamma, M)$ besteht aus:

- V, endlich vielen *Zustandsvariablen*,
- N_C, endlich vielen *abstrakten Aktionsnamen*,
- N_P, endlich vielen *primitiven Aktionsnamen*,
- $\gamma : N_P \to A$, bijektive Funktion, wobei A die *Aktionen* sind,
- $M \subseteq N_C \times P_{N_C \cup N_P}$, endlich vielen *Methoden*.

Formale Problembeschreibung, (TI)HTN-Domäne

Definition (HTN/TIHTN-Planungsdomäne, grundiert)

Eine *Planungsdomäne* $\mathcal{D} = (V, N_C, N_P, \gamma, M)$ besteht aus:

- V, endlich vielen *Zustandsvariablen*,
- N_C, endlich vielen *abstrakten Aktionsnamen*,
- N_P, endlich vielen *primitiven Aktionsnamen*,
- $\gamma : N_P \to A$, bijektive Funktion, wobei A die *Aktionen* sind,
- $M \subseteq N_C \times P_{N_C \cup N_P}$, endlich vielen *Methoden*.

Definition (Aktion, grundiert)

Eine *Aktion* $a = (pre^+, pre^-, eff^+, eff^-)$ besteht aus:

- $pre^+, pre^- \subseteq V$, den *Vorbedingungen* von a,
- $eff^+, eff^- \subseteq V$, den *Effekten* von a.

Formale Problembeschreibung, (TI)HTN-Domäne

Definition (HTN/TIHTN-Planungsdomäne, grundiert)

Eine *Planungsdomäne* $\mathcal{D} = (V, N_C, N_P, \gamma, M)$ besteht aus:

- V, endlich vielen *Zustandsvariablen*,
- N_C, endlich vielen *abstrakten Aktionsnamen*,
- N_P, endlich vielen *primitiven Aktionsnamen*,
- $\gamma : N_P \to A$, bijektive Funktion, wobei A die *Aktionen* sind,
- $M \subseteq N_C \times P_{N_C \cup N_P}$, endlich vielen *Methoden*.

Definition (Plan, grundiert)

Ein *Plan* $P = (L, \prec, \alpha)$ besteht aus:

- L den Labels,
- $\prec \subseteq L \times L$, der partiellen Ordnung auf L und
- $\alpha : L \to N_C \cup N_P$, der Labelfunktion.

Formale Problembeschreibung, (TI)HTN-Problem

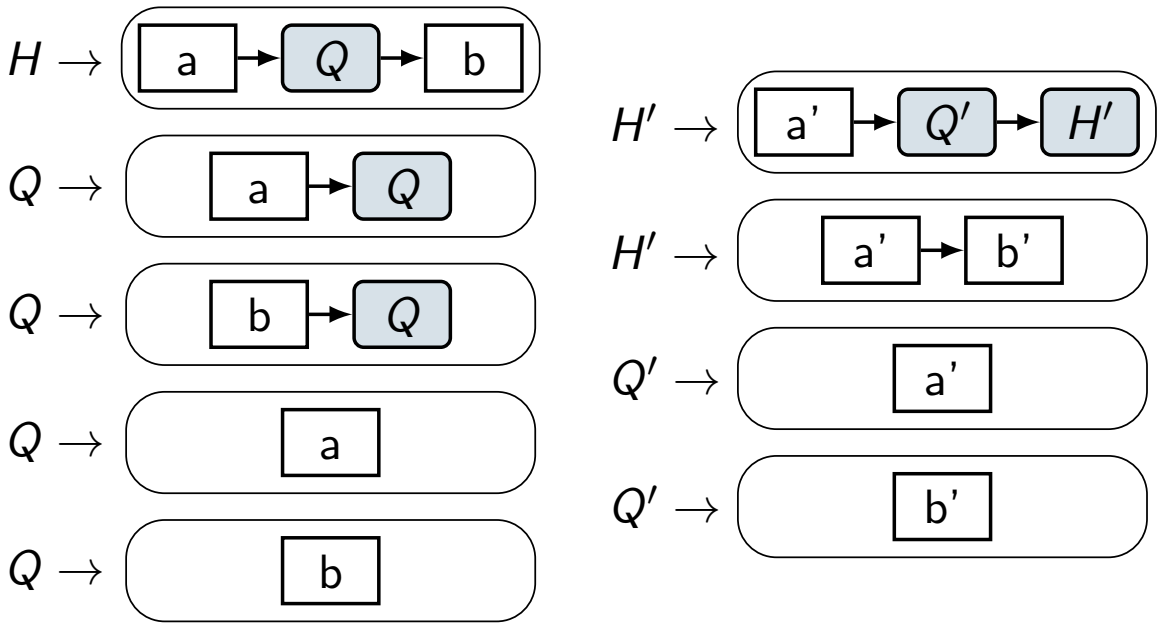
Definition (Planungsproblem, grundiert)

Ein *Planungsproblem* $\mathcal{P} = (\mathcal{D}, s_{\text{init}}, P_{\text{init}}, g)$ besteht aus:

- $\mathcal{D}$, der *Planungsdomäne*,
- $s_{\text{init}} \in 2^V$ dem *Anfangszustand*,
- $P_{\text{init}} \subseteq P_{N_C \cup N_P}$ dem *Anfangsplan* und
- $g = (g^+, g^-)$ mit $g^+, g^- \subseteq V$ der *Zielzustandsbeschreibung*.

Hierarchisches Planen und formale Sprachen I

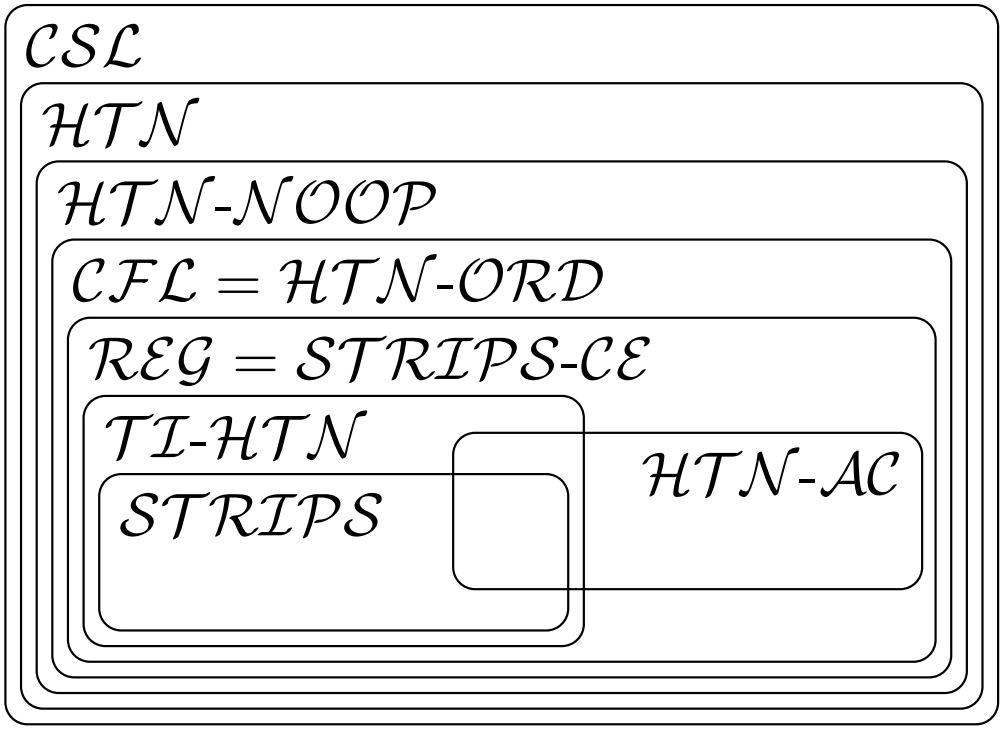
Mit Methoden kann man Grammatiken beschreiben; hier G und G':



Das Startsymbol von G ist H; damit ist $L(G) = a(a|b)^+b$.
Das Startsymbol von G' ist H'; damit ist $L(G') = (a'(a'|b'))^*a'b'$.

Hierarchisches Planen und formale Sprachen II

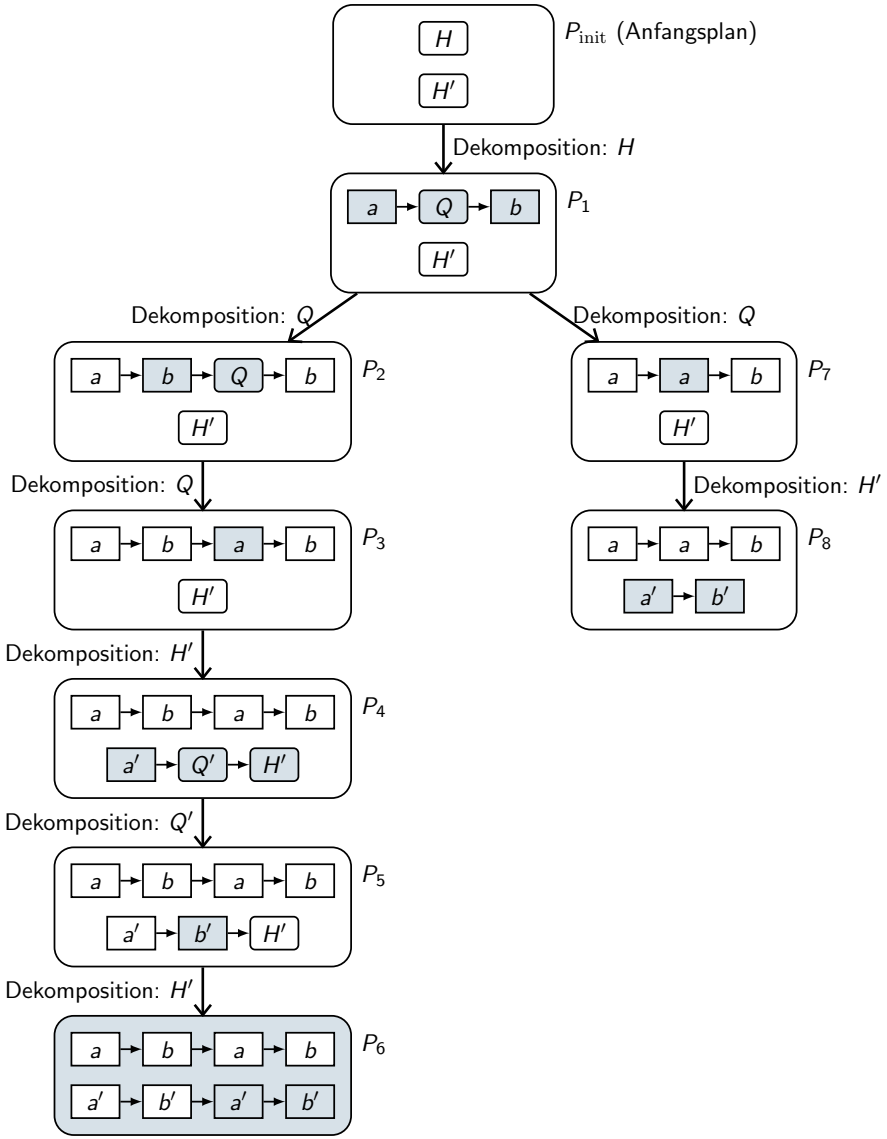
Ausdrucksmächtigkeit der verschiedenen Formalismen:



Komplexitäten im HTN- und TIHTN-Planen

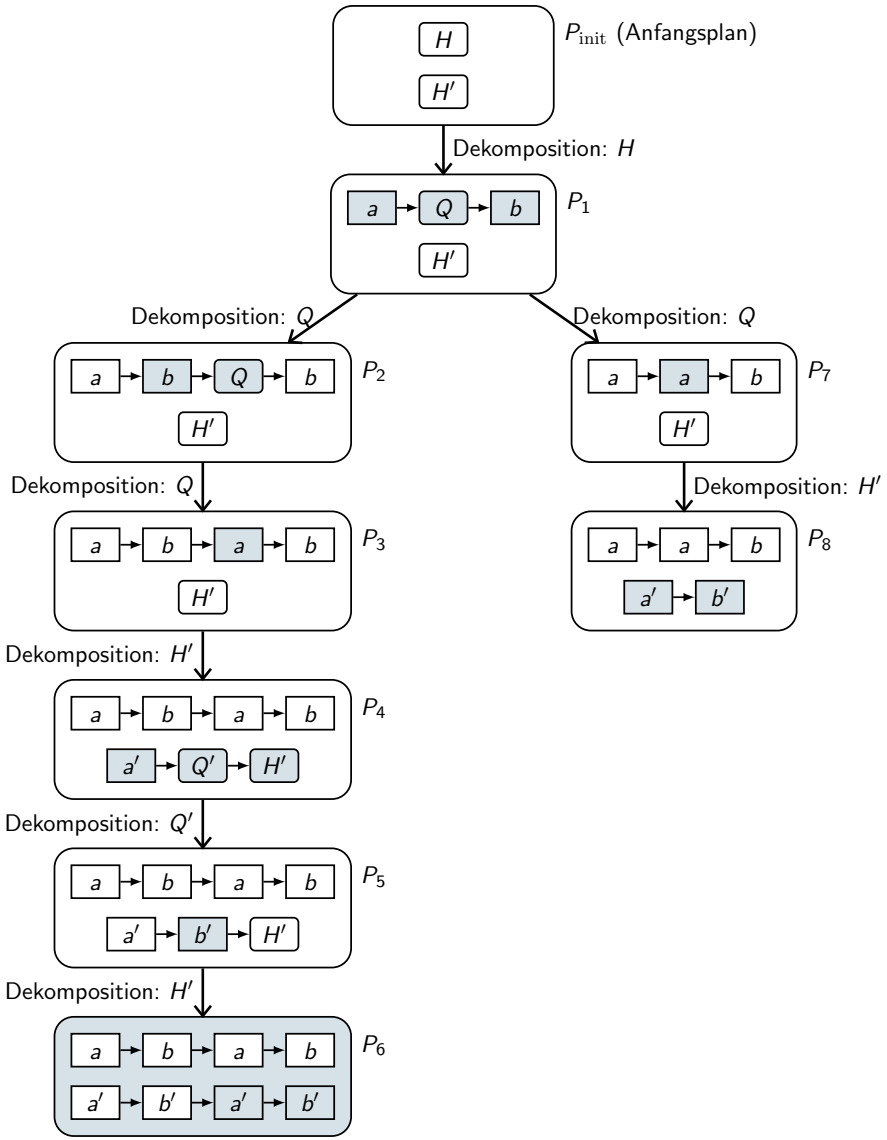
Restrictions on Hierarchy	Task Networks Totally Ordered?	Task Insertion?	Complexity Class
Arbitrary	yes	yes	**PSPACE**
Arbitrary	yes	no	**EXPTIME**
Arbitrary	no	yes	**NEXPTIME**
Arbitrary	no	no	**undecidable** and **semi−decidable**
Tail-Recursive	yes	yes	**PSPACE**
Tail-Recursive	yes	no	**PSPACE**
Tail-Recursive	no	yes	**NEXPTIME**
Tail-Recursive	no	no	**EXPSPACE**
Regular	yes	yes	**PSPACE**
Regular	yes	no	**PSPACE**
Regular	no	yes	**PSPACE**
Regular	no	no	**PSPACE**
Acyclic	yes	yes	**PSPACE**
Acyclic	yes	no	**PSPACE**
Acyclic	no	yes	**NEXPTIME**
Acyclic	no	no	**NEXPTIME**
No Hierarchy	yes	yes	**PSPACE**
No Hierarchy	yes	no	in **P**
No Hierarchy	no	yes	**PSPACE**
No Hierarchy	no	no	**NP**

Das Planexistenzproblem für TIHTN-Probleme



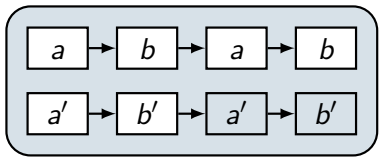
In diesem Beispielproblem ist ein Plan eine Lösung, wenn er zwei namensgleiche Aktionssequenzen enthält.

Das Planexistenzproblem für TIHTN-Probleme

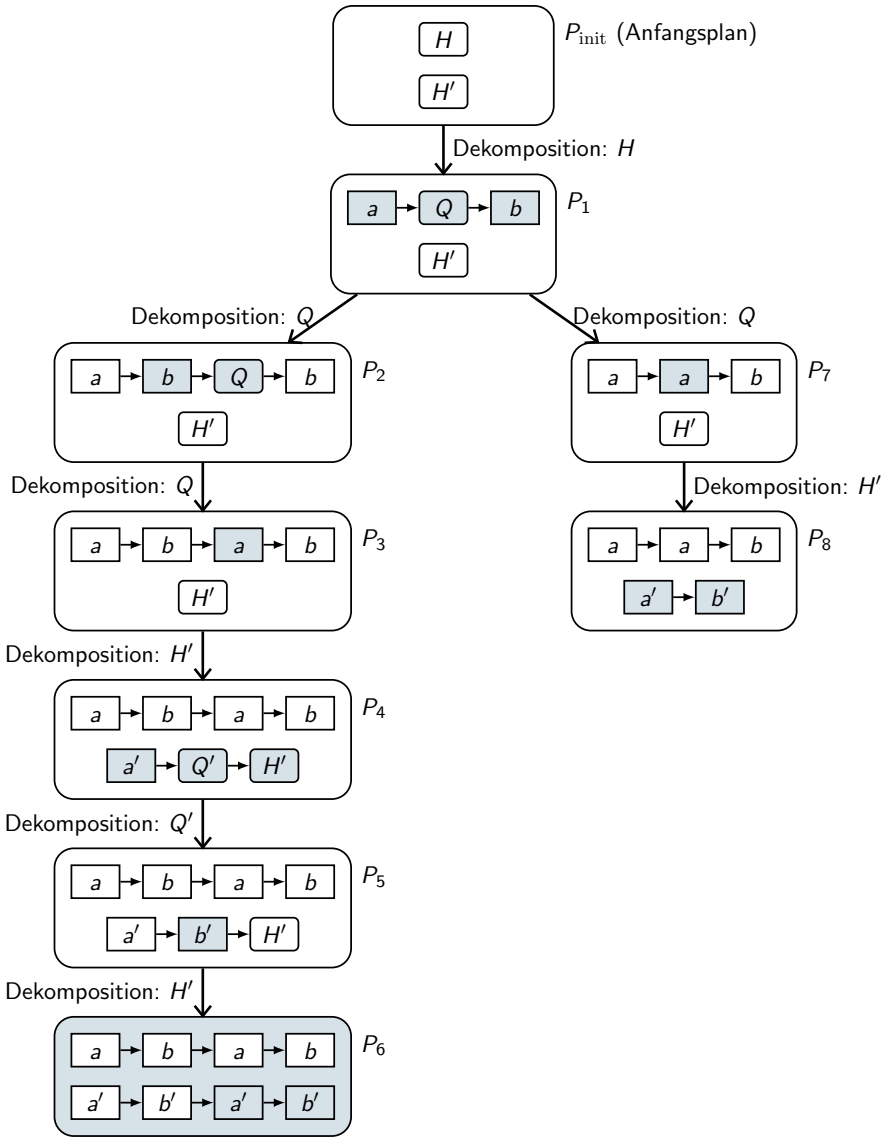


In diesem Beispielproblem ist ein Plan eine Lösung, wenn er zwei namensgleiche Aktionssequenzen enthält.

Plan P_6 ist eine Lösung!

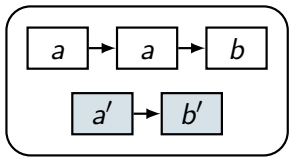


Das Planexistenzproblem für TIHTN-Probleme

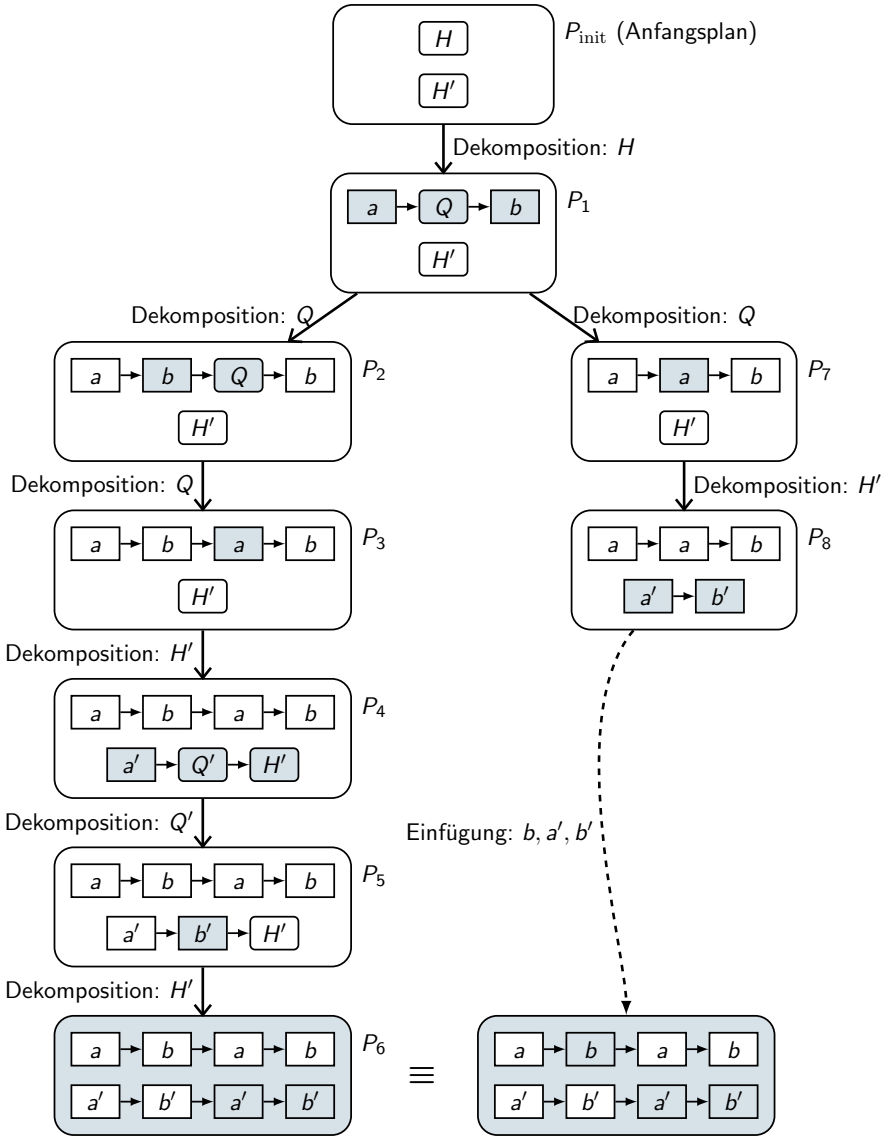


In diesem Beispielproblem ist ein Plan eine Lösung, wenn er zwei namensgleiche Aktionssequenzen enthält.

Plan P_8 ist keine Lösung!

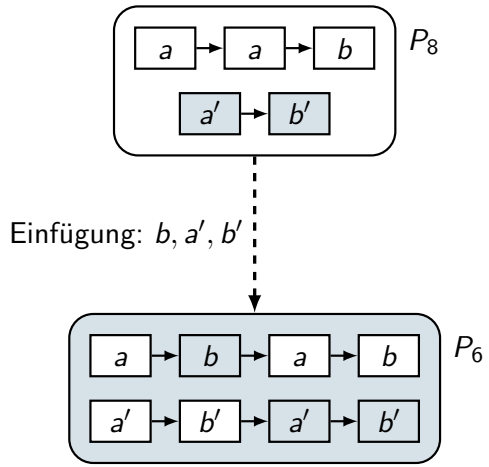


Das Planexistenzproblem für TIHTN-Probleme



In diesem Beispielproblem ist ein Plan eine Lösung, wenn er zwei namensgleiche Aktionssequenzen enthält.

Einfluss der Task-Einfügung:



Unentscheidbarkeitsbeweis hybrides Planen

- *Problem:* die Legalitätskriterien schränken die ausdrückbaren Modelle ein? ... wirklich?
- Wir zeigen, dass jedes HTN-Problem π in ein hybrides Problem π' überführt werden kann, so dass gilt:
  - π hat eine Lösung genau dann wenn π' eine besitzt und
  - π' erfüllt alle Legalitätskriterien aus der Literatur.

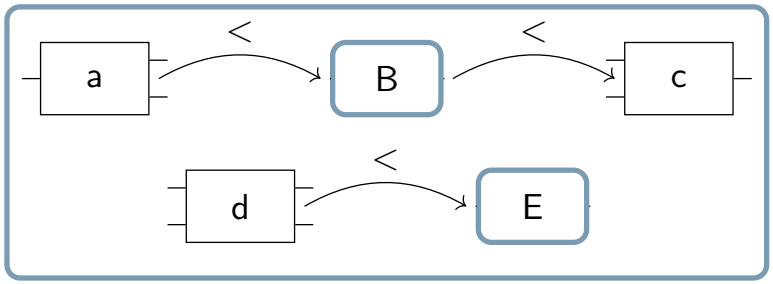
Unentscheidbarkeitsbeweis hybrides Planen

- *Problem:* die Legalitätskriterien schränken die ausdrückbaren Modelle ein? ... wirklich?
- Wir zeigen, dass jedes HTN-Problem π in ein hybrides Problem π' überführt werden kann, so dass gilt:
  - π hat eine Lösung genau dann wenn π' eine besitzt und
  - π' erfüllt alle Legalitätskriterien aus der Literatur.

Kodierung: HTN-Problem in legales hybrides Problem

Für jeden primitiven Task t, erstelle eine abstrakte Kopie T ohne Vorbedingungen und Effekte. Dann:

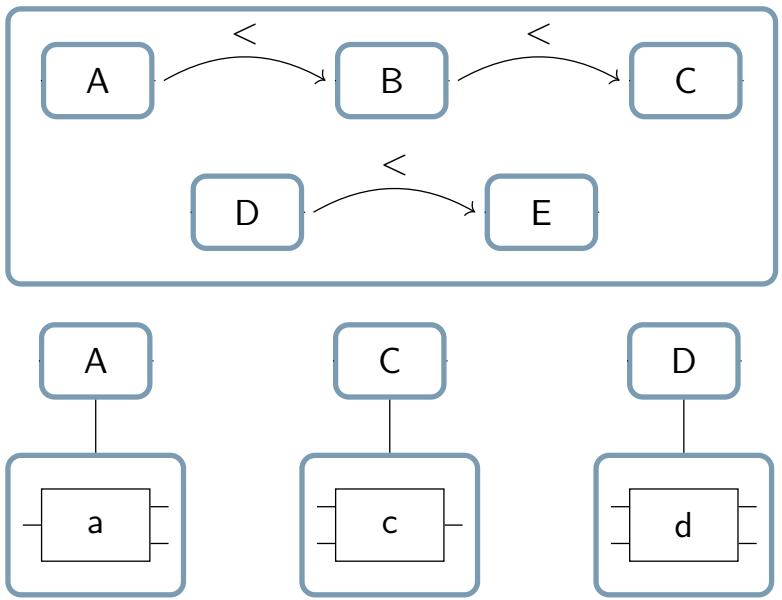
- Füge eine Methode hinzu, deren Plan P exakt t enthält.
- Ersetze in jedem Plan t durch T.



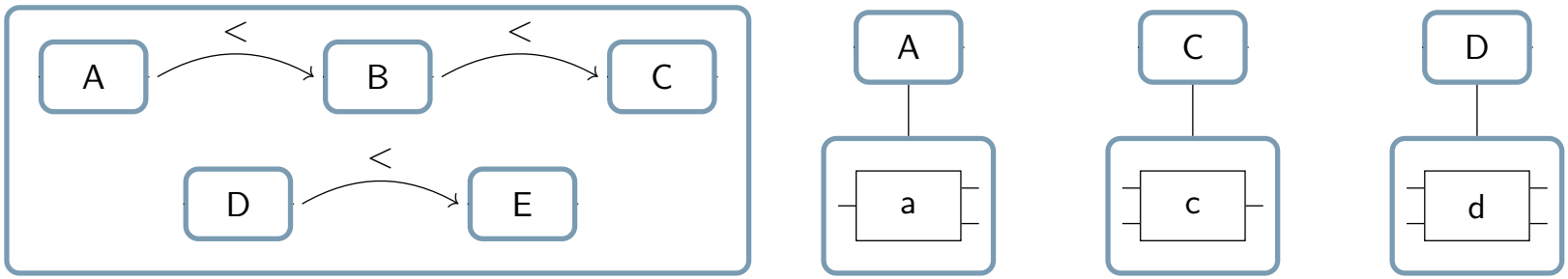
Kodierung: HTN-Problem in legales hybrides Problem

Für jeden primitiven Task t, erstelle eine abstrakte Kopie T ohne Vorbedingungen und Effekte. Dann:

- Füge eine Methode hinzu, deren Plan P exakt t enthält.
- Ersetze in jedem Plan t durch T.



Kodierung: HTN-Problem in legales hybrides Problem



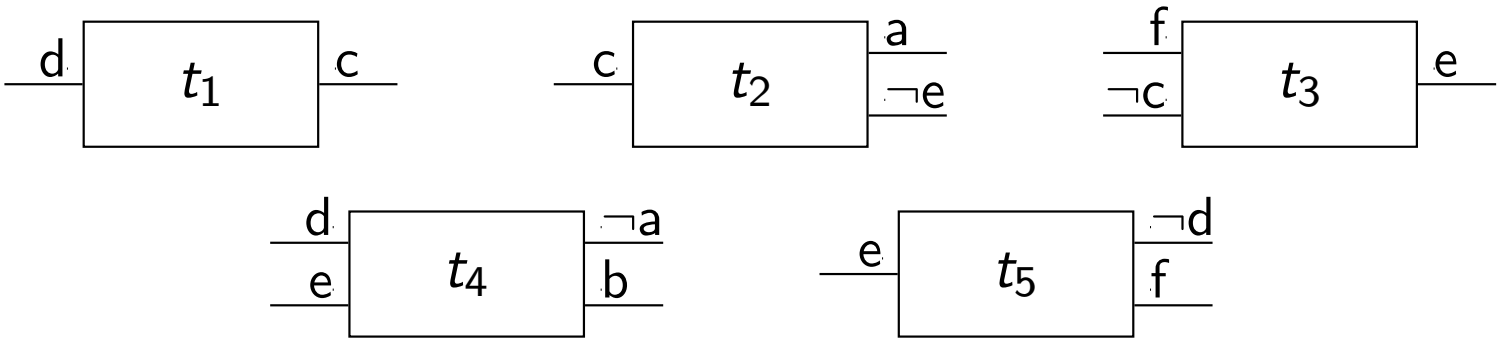
Modelleigenschaften:

- Keine abstrakte Aktion hat Vorbedingungen oder Effekte.
- Für alle Pläne gilt:
  - entweder gibt es nur abstrakte Aktionen
  - oder höchstens eine.
- Daraus folgt, dass alle Methoden alle Legalitätskriterien erfüllen!

Delete-Relaxierung im klassischen Planen

Idee

Problemvereinfachung durch Delete-Relaxierung

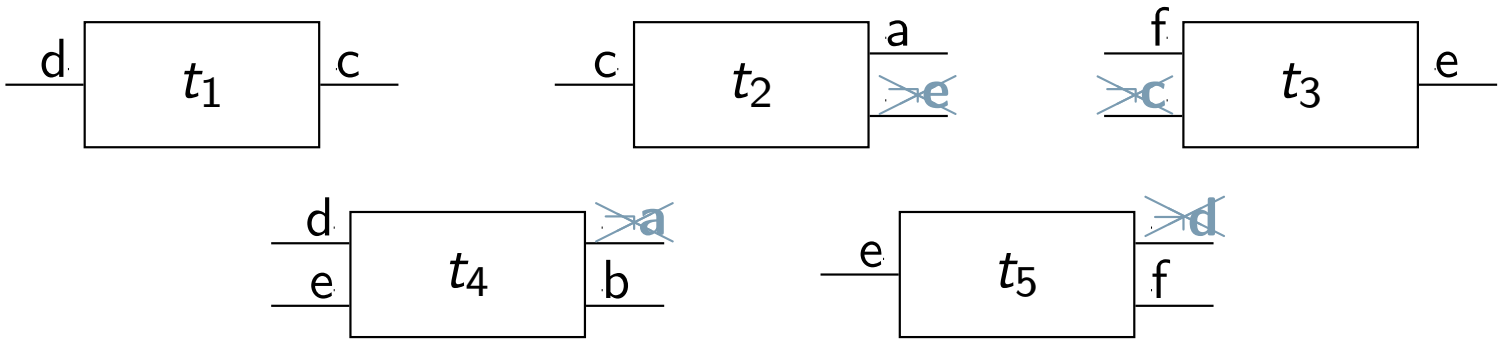


Klassisches Planungsproblem:

Delete-Relaxierung im klassischen Planen

Idee

Problemvereinfachung durch Delete-Relaxierung

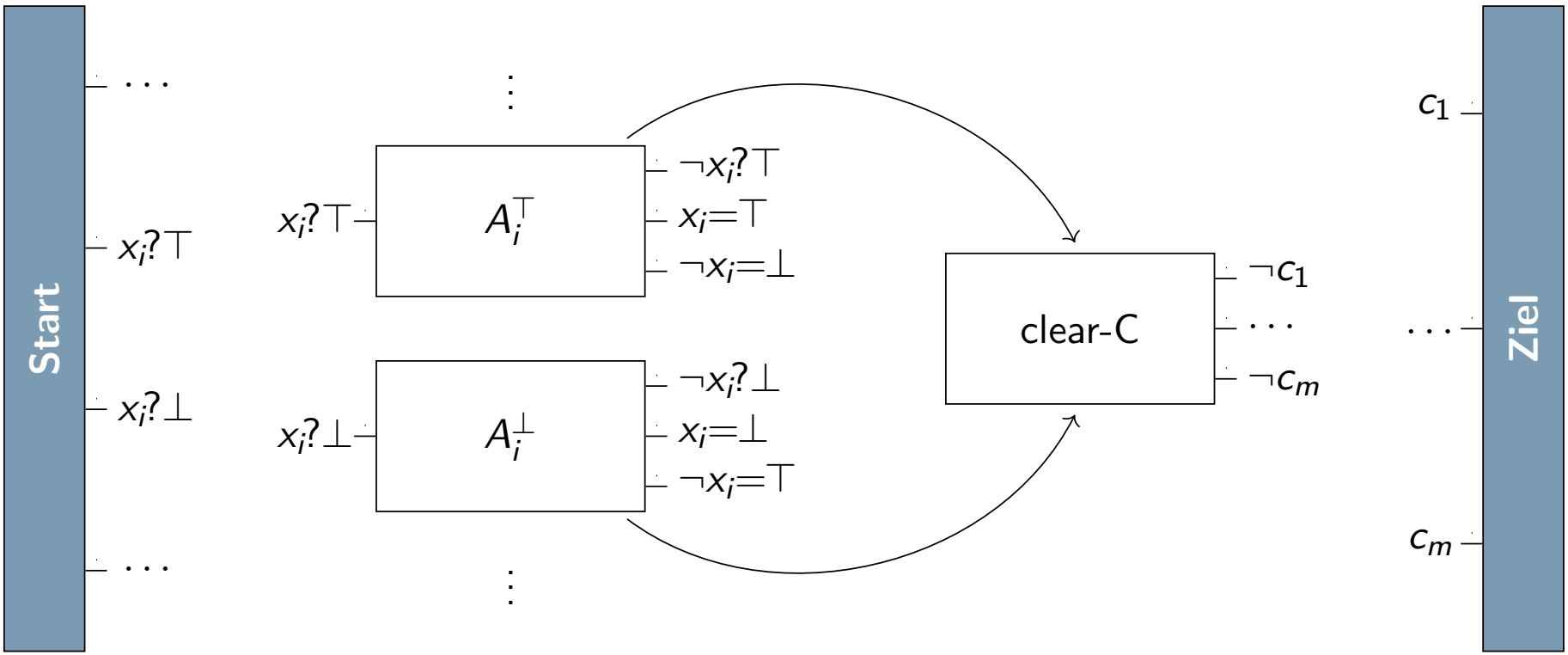


Klassisches Planen: Relaxierung reduziert Komplexität von PSPACE-vollständig auf NP bzw. P:

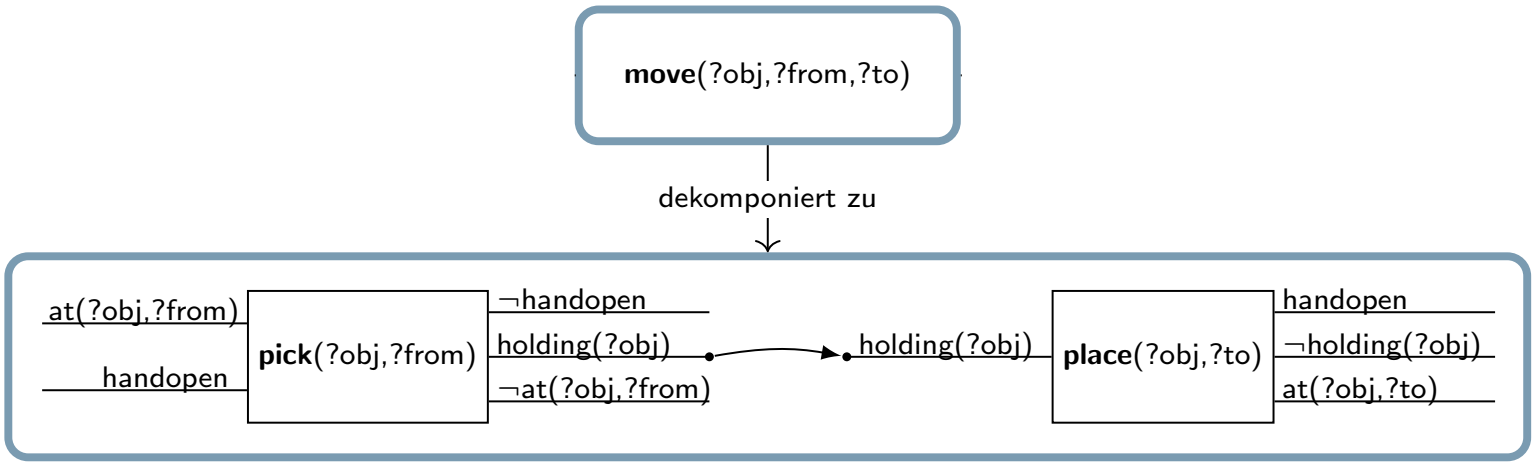
- positive Effekte, positive Vorbedingungen: P
- positive Effekte, beliebige Vorbedingungen: NP-vollständig

Partielle Delete-Relaxierung: NP-Härte

$x^1, \ldots, x^n$ sind Bool'sche Variablen. $c^1, \ldots, c^m$ sind Klauseln. Jedes c^j entspricht einer Disjunktion. Für $x^i \in c^j$ gibt es eine Aktion $\langle \{x_i = \top\}, \{c_j\}, \emptyset \rangle$.



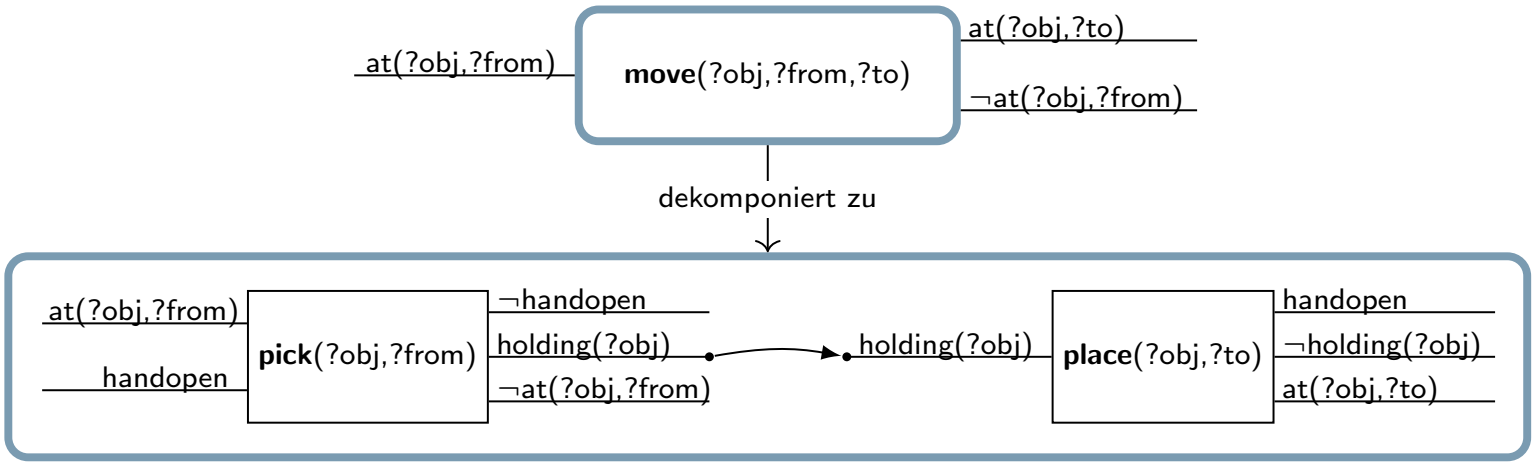
Legalitätskriterien: Motivation



Der hybride Planungsformalismus erlaubt es, Vorbedingungen und Effekte für abstrakte Aktionen anzugeben. Aber wozu?

- Um abstrakte Lösungen zu generieren,
- zur Ausnutzung während der Suche und
- *für Modellierungsassistenz* durch Einschränkung auf legale Modelle (Test aller Dekompositionsmethoden).

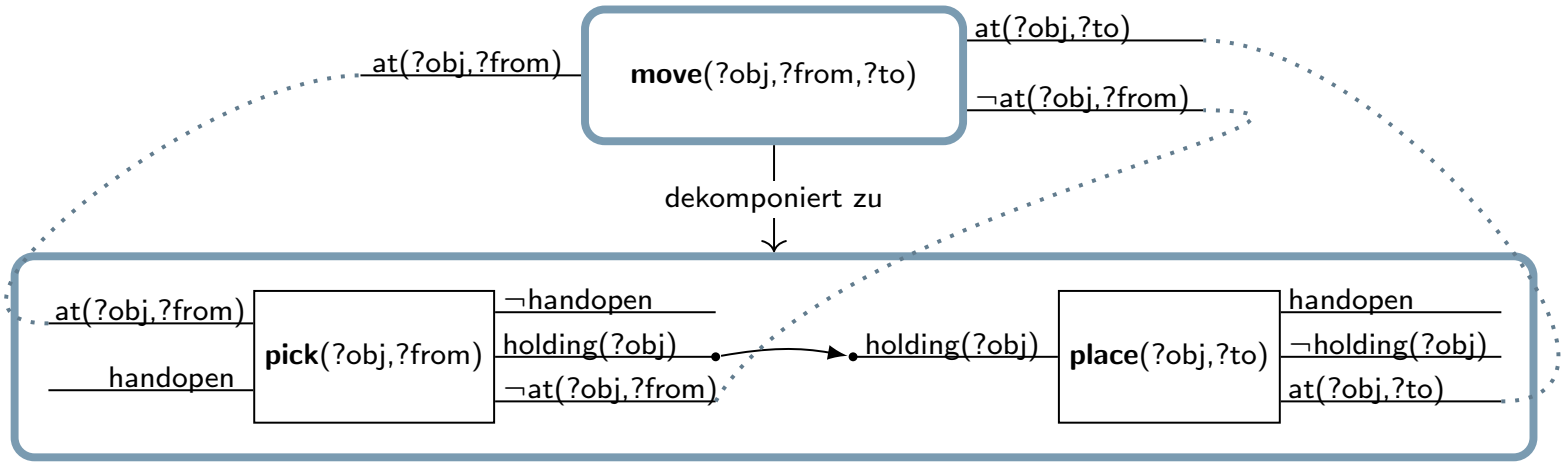
Legalitätskriterien: Motivation



Der hybride Planungsformalismus erlaubt es, Vorbedingungen und Effekte für abstrakte Aktionen anzugeben. Aber wozu?

- Um abstrakte Lösungen zu generieren,
- zur Ausnutzung während der Suche und
- *für Modellierungsassistenz* durch Einschränkung auf legale Modelle (Test aller Dekompositionsmethoden).

Legalitätskriterien: Motivation



Der hybride Planungsformalismus erlaubt es, Vorbedingungen und Effekte für abstrakte Aktionen anzugeben. Aber wozu?

- Um abstrakte Lösungen zu generieren,
- zur Ausnutzung während der Suche und
- *für Modellierungsassistenz* durch Einschränkung auf legale Modelle (Test aller Dekompositionsmethoden).

Legalitätskriterien: Übersicht I

Definition (Downward Compatible, Bercher et al., ECAI-2016)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein Plan.

- Falls $\varphi \in pre$, dann existiert φ als Vorbedingung eines Tasks in P ohne kausalen Link, der darauf zeigt.
- Falls $\varphi \in eff$, dann existiert φ als Effekt eines Tasks in P.

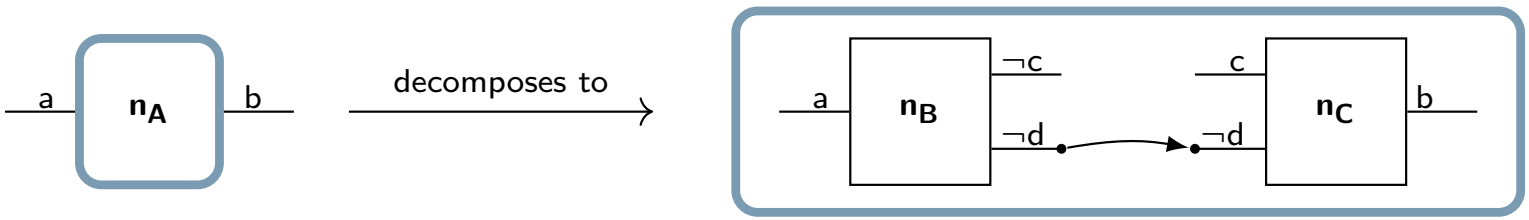
- Verhindert/detektiert die offensichtlichsten Modellierungsfehler.
- Abstrakte Aktionen können immer dekomponiert werden (wie im HTN-Planen).

Legalitätskriterien: Übersicht I

Definition (Downward Compatible, Bercher et al., ECAI-2016)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein Plan.

- Falls $\varphi \in pre$, dann existiert φ als Vorbedingung eines Tasks in P ohne kausalen Link, der darauf zeigt.
- Falls $\varphi \in eff$, dann existiert φ als Effekt eines Tasks in P.



(Diese Methode erfüllt das Kriterium.)

Legalitätskriterien: Übersicht II

Definition (Biundo and Schattenberg, 2001)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein total geordneter Plan.

- Es muss ein Zustand s existieren, der pre erfüllt, $s \models pre$, so dass P's Tasksequenz $\bar{t}$ in s ausführbar ist.
- Für alle Zustände, die das erste Kriterium erfüllen, generiert $\bar{t}$ einen Zustand, der eff erfüllt, $s \models eff$.

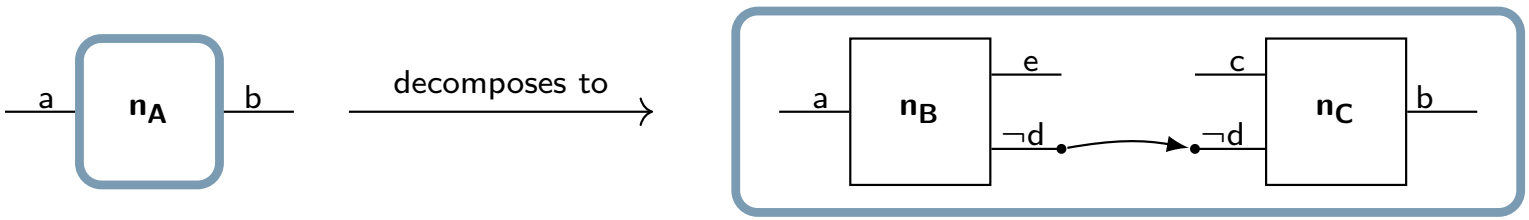
- Zustandstransitionssemantik adaptiert auf abstrakte Aktionen.
- Starke Annahme: Offene Vorbedingungen des Plans müssen durch einen singulären Zustand erfüllt werden.

Legalitätskriterien: Übersicht II

Definition (Biundo and Schattenberg, 2001)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein total geordneter Plan.

- Es muss ein Zustand s existieren, der pre erfüllt, $s \models pre$, so dass P's Tasksequenz $\bar{t}$ in s ausführbar ist.
- Für alle Zustände, die das erste Kriterium erfüllen, generiert $\bar{t}$ einen Zustand, der eff erfüllt, $s \models eff$.



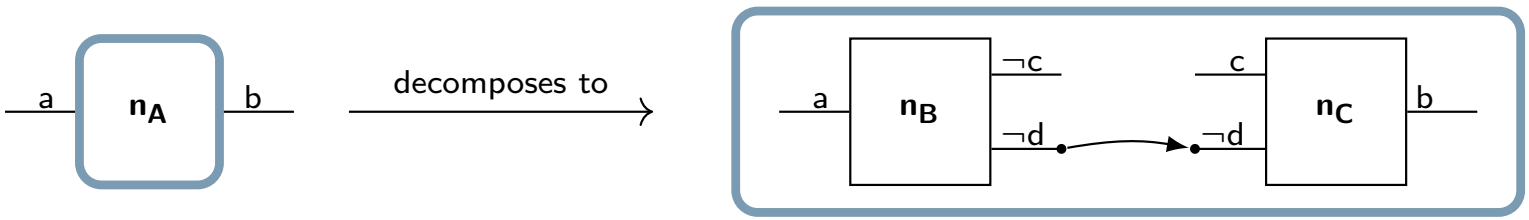
(Diese Methode erfüllt das Kriterium.)

Legalitätskriterien: Übersicht II

Definition (Biundo and Schattenberg, 2001)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein total geordneter Plan.

- Es muss ein Zustand s existieren, der pre erfüllt, $s \models pre$, so dass P's Tasksequenz $\bar{t}$ in s ausführbar ist.
- Für alle Zustände, die das erste Kriterium erfüllen, generiert $\bar{t}$ einen Zustand, der eff erfüllt, $s \models eff$.



(Diese Methode erfüllt das Kriterium nicht.)

Legalitätskriterien: Übersicht III

Definition (Yang, 1990)

Let $m = (n_c, P)$ be a method, $n_c = (pre, eff)$ an abstract task, and P a plan.

- *pre* und *eff* sind Vorbedingungen und Effekte in P.
- Es gibt keine kausalen Bedrohungen.

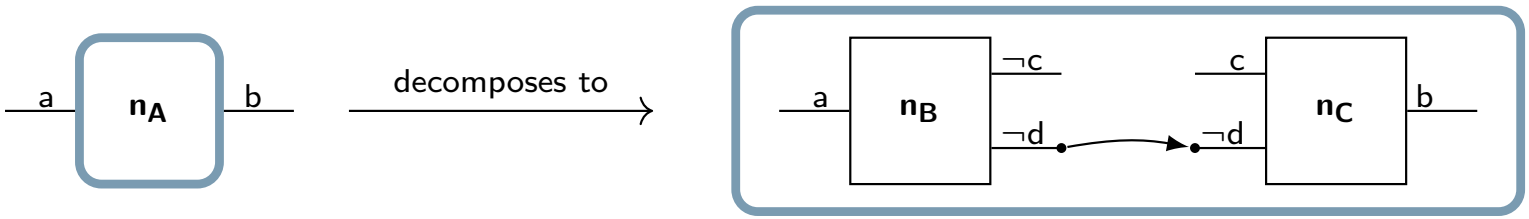
- Etwas schwächer als das vorherige Kriterium: Offene Vorbedingungen müssen nicht durch einen singulären Zustand wahr gemacht werden.

Legalitätskriterien: Übersicht III

Definition (Yang, 1990)

Let $m = (n_c, P)$ be a method, $n_c = (pre, eff)$ an abstract task, and P a plan.

- *pre* und *eff* sind Vorbedingungen und Effekte in P.
- Es gibt keine kausalen Bedrohungen.



(Diese Methode erfüllt das Kriterium.)

Legalitätskriterien: Übersicht IV

Definition (Young et al., 1994)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein Plan.

- Jede Vorbedingung aus *pre* trägt zu mindestens einem Effekt in *eff* durch eine Folge von kausalen Links bei.
- ... und umgekehrt.

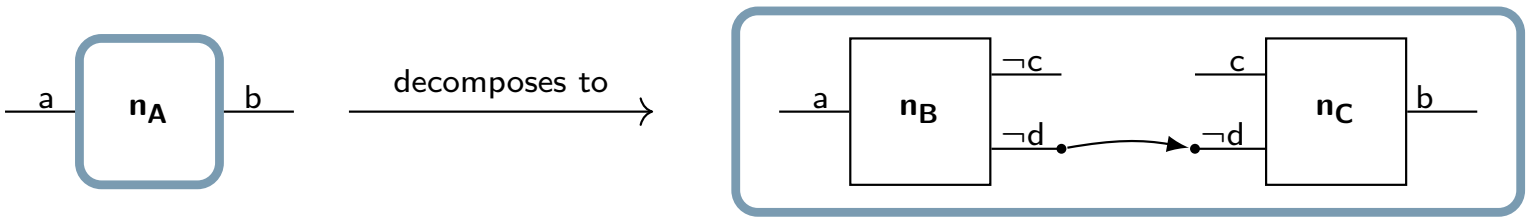
- Weder stärker noch schwächer als das andere Kriterium.

Legalitätskriterien: Übersicht IV

Definition (Young et al., 1994)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein Plan.

- Jede Vorbedingung aus *pre* trägt zu mindestens einem Effekt in *eff* durch eine Folge von kausalen Links bei.
- ... und umgekehrt.



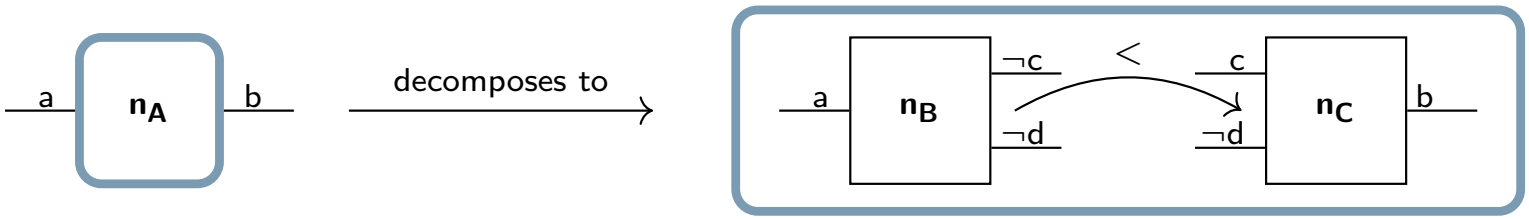
(Diese Methode erfüllt das Kriterium.)

Legalitätskriterien: Übersicht IV

Definition (Young et al., 1994)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein Plan.

- Jede Vorbedingung aus *pre* trägt zu mindestens einem Effekt in *eff* durch eine Folge von kausalen Links bei.
- ... und umgekehrt.



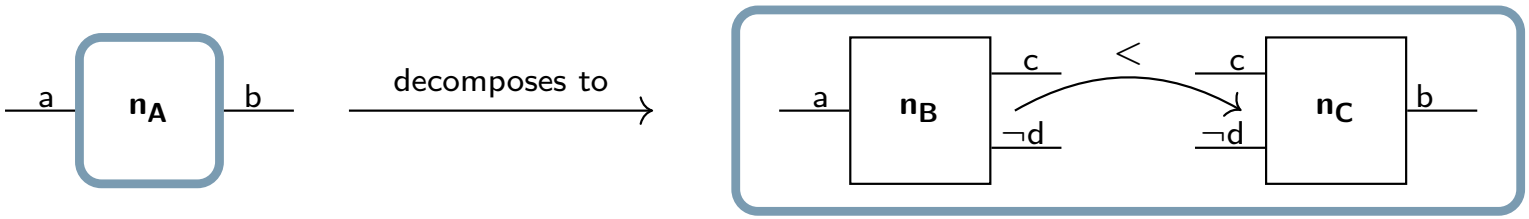
(Diese Methode erfüllt das Kriterium nicht.)

Legalitätskriterien: Übersicht IV

Definition (Young et al., 1994)

Sei $m = (n_c, P)$ eine Methode, $n_c = (pre, eff)$ ein abstrakter Task und P ein Plan.

- Jede Vorbedingung aus *pre* trägt zu mindestens einem Effekt in *eff* durch eine Folge von kausalen Links bei.
- ... und umgekehrt.



(Diese Methode erfüllt das Kriterium nicht.)

Pseudocode von PANDA

Input : $\texttt{Fringe} = \{P_{\text{init}}\}$
Output : Eine Lösung oder `fail`.

1 **while** $\texttt{Fringe} \neq \emptyset$ **do**
2 $\quad P := \textbf{PlanSel}(\texttt{Fringe})$
3 $\quad F := \mathit{FlawDet}(P)$
4 $\quad$ **if** $F = \emptyset$ **then return** P
5 $\quad f := \textbf{FlawSel}(F)$
6 $\quad \texttt{Fringe} := (\texttt{Fringe} \setminus \{P\}) \cup \mathit{Successors}(P, f)$
7 **return** `fail`

Hybrider Planungsalgorithmus: Flaws und Modifikationen

Mögliche Flaws und ihre Modifikationen:

Flaw	**Modifikation**
Abstrakter Task	Dekomposition
Offene Vorbedingung	Einfügung eines kausalen Links mittels: • Bestehender Aktion • Aktion aus Domäne (Aktionseinfügung) • Aktion aus Methode (Dekomposition)
Kausale Bedrohung	• Einfügung eines Ordnungsconstraints • Einfügung einer Variablenbindung • Dekomposition

Kodierung von POCL-Plänen: Motivation

Wir benötigen:

- Heuristik für Pläne.

In der Literatur bereits vorhanden:

- eine Vielzahl informativer Heuristiken für Zustände.

Idee:
Überführe Plan in einen Zustand!

Kodierung von POCL-Plänen: Ansatz

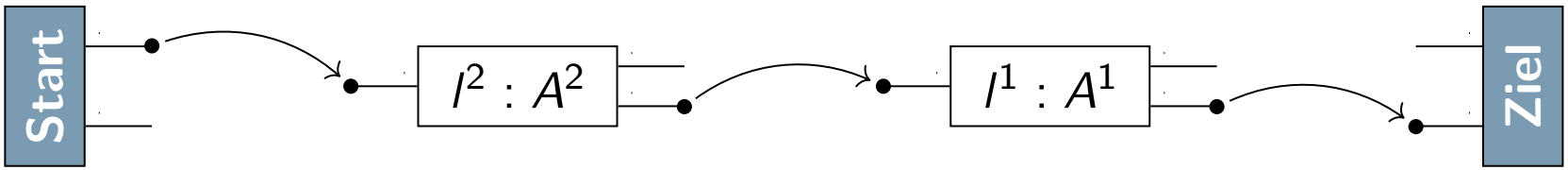
Überführe Planungsproblem $\mathcal{P} = \langle \mathcal{D}, P_{\text{init}} \rangle$ in $\mathcal{P}' = \langle \mathcal{D}', s_{\text{init}} \rangle$, s.d.:

Lösungen für $\mathcal{P}$	$\equiv$	Lösungen für $\mathcal{P}'$
Zieldistanz für P_{init}	$\equiv$	Zieldistanz für s_{init}

Dann:
Für (bereits existierende) zustandsbasierte Heuristik h' setze
$h(P) := h'(s_{\text{init}})$

Kodierung von POCL-Plänen: Beispiel

Sei $\mathcal{P} = \langle \mathcal{D}, s_{init}, g \rangle$ mit $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ und P gegeben durch:

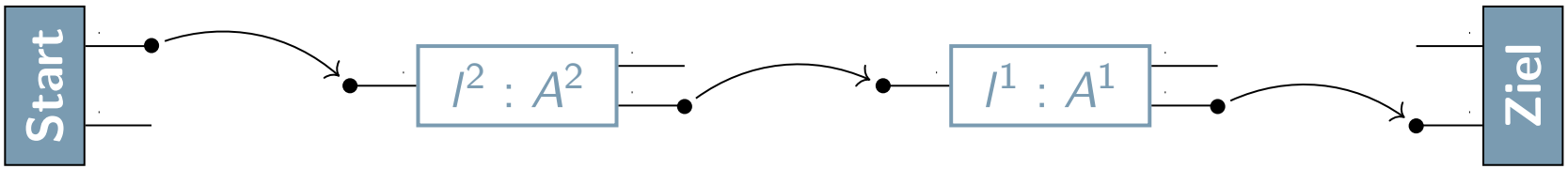


P ist kodiert in $\mathcal{P}' = \langle \mathcal{D}', s'_{init}, g' \rangle$ mit $\mathcal{D}' = \langle \mathcal{V}', \mathcal{A}' \rangle$:

- $\mathcal{V}' := \mathcal{V} \cup \{l_1, l_2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l_1 : A_1), enc(l_2 : A_2)\}$ with
  - $enc(l_1 : A_1) = \langle pre(A_1) \wedge \neg l_1 \wedge l_2, eff(A_1) \wedge l_1 \rangle$
  - $enc(l_2 : A_2) = \langle pre(A_2) \wedge \neg l_2, eff(A_2) \wedge l_2 \rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l_1, l_2\}$

Kodierung von POCL-Plänen: Beispiel

Sei $\mathcal{P} = \langle \mathcal{D}, s_{init}, g \rangle$ mit $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ und P gegeben durch:

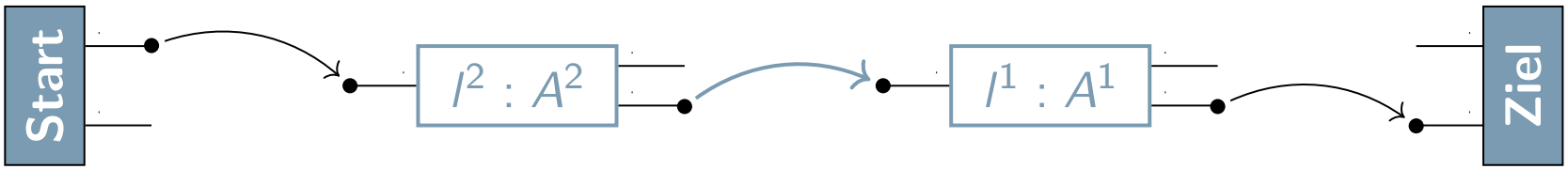


P ist kodiert in $\mathcal{P}' = \langle \mathcal{D}', s'_{init}, g' \rangle$ mit $\mathcal{D}' = \langle \mathcal{V}', \mathcal{A}' \rangle$:

- $\mathcal{V}' := \mathcal{V} \cup \{l_1, l_2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l_1 : A_1), enc(l_2 : A_2)\}$ with
  - $enc(l_1 : A_1) = \langle pre(A_1) \wedge \neg l_1 \wedge l_2, eff(A_1) \wedge l_1 \rangle$
  - $enc(l_2 : A_2) = \langle pre(A_2) \wedge \neg l_2, eff(A_2) \wedge l_2 \rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l_1, l_2\}$

Kodierung von POCL-Plänen: Beispiel

Sei $\mathcal{P} = \langle \mathcal{D}, s_{init}, g \rangle$ mit $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ und P gegeben durch:

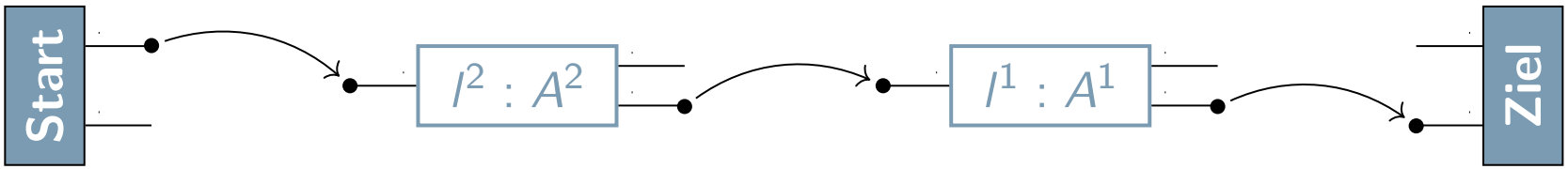


P ist kodiert in $\mathcal{P}' = \langle \mathcal{D}', s'_{init}, g' \rangle$ mit $\mathcal{D}' = \langle \mathcal{V}', \mathcal{A}' \rangle$:

- $\mathcal{V}' := \mathcal{V} \cup \{l_1, l_2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l_1 : A_1), enc(l_2 : A_2)\}$ with
  - $enc(l_1 : A_1) = \langle pre(A_1) \wedge \neg l_1 \wedge l_2, eff(A_1) \wedge l_1 \rangle$
  - $enc(l_2 : A_2) = \langle pre(A_2) \wedge \neg l_2, eff(A_2) \wedge l_2 \rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l_1, l_2\}$

Kodierung von POCL-Plänen: Beispiel

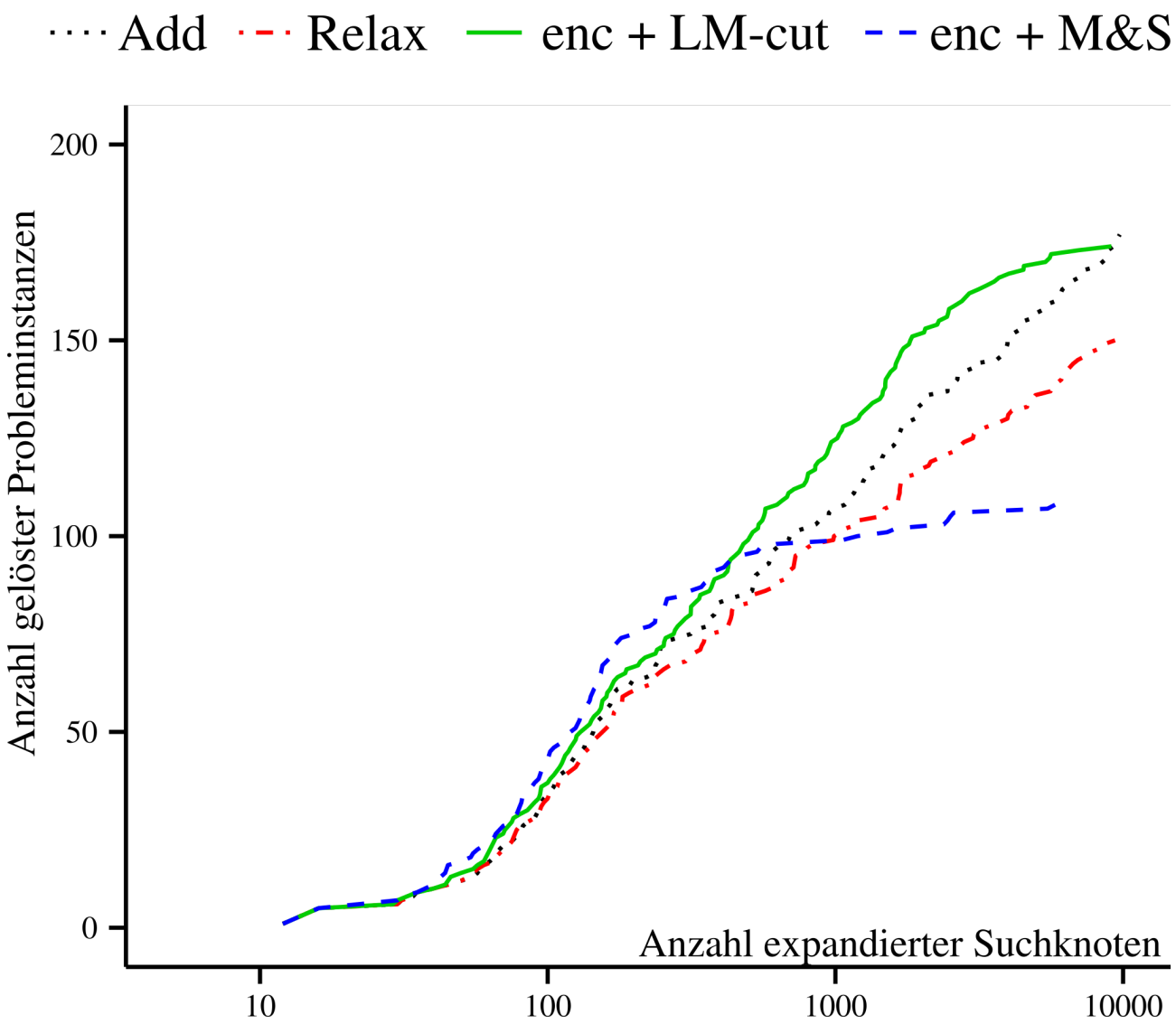
Sei $\mathcal{P} = \langle \mathcal{D}, s_{init}, g \rangle$ mit $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ und P gegeben durch:



P ist kodiert in $\mathcal{P}' = \langle \mathcal{D}', s'_{init}, g' \rangle$ mit $\mathcal{D}' = \langle \mathcal{V}', \mathcal{A}' \rangle$:

- $\mathcal{V}' := \mathcal{V} \cup \{l_1, l_2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l_1 : A_1), enc(l_2 : A_2)\}$ with
  - $enc(l_1 : A_1) = \langle pre(A_1) \wedge \neg l_1 \wedge l_2, eff(A_1) \wedge l_1 \rangle$
  - $enc(l_2 : A_2) = \langle pre(A_2) \wedge \neg l_2, eff(A_2) \wedge l_2 \rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l_1, l_2\}$

Kodierung von POCL-Plänen: Empirische Evaluation I



Kodierung von POCL-Plänen: Empirische Evaluation II

Domöne	n	Add	Relax	enc + LM-Cut	enc + M&S	FD + LM-Cut	FD + M&S
grid	5	**0**	**0**	**0**	**0**	2	2
gripper	20	14	**20**	1	1	20	8
logistics	20	**16**	15	6	0	16	1
movie	30	**30**	**30**	**30**	**30**	30	30
mystery	20	8	**10**	5	5	13	13
mystery-prime	20	3	**4**	2	1	12	12
blocks	21	2	3	3	**5**	21	21
logistics	28	**28**	**28**	27	5	28	15
miconic	100	**100**	53	65	29	100	68
depot	22	**2**	**2**	1	1	11	7
driverlog	20	7	**9**	3	3	15	12
freecell	20	**0**	**0**	**0**	**0**	6	6
rover	20	**20**	19	9	5	18	8
zeno-travel	20	4	4	3	**5**	16	13
airport	20	**18**	15	6	5	20	18
pipesworld-noTankage	20	**8**	5	1	1	18	19
pipesworld-Tankage	20	**1**	**1**	**1**	**1**	11	14
satellite	20	**18**	**18**	4	3	15	7
pipesworld	20	**1**	**1**	**1**	**1**	11	14
rover	20	**0**	**0**	**0**	**0**	18	8
storage	20	7	**9**	5	5	17	15
tpp	20	**9**	8	5	5	9	7
total	526	296	254	178	111	427	318

Heuristik Sample-FF

Basiert auf partieller Delete-Relaxierung (NP-vollständig)

Erinnerung: Aktueller Plan wird *nicht* relaxiert!
D.h. es wird berücksichtigt:

- Ordnung auf den Aktionen
- Kausale Links(!), d.h. pruning!
- Alle Vorbedingungen und Effekte der vorhandenen Aktionen

Einzige Relaxierung:

- Delete-Relaxierung der Domäne: Nur positive Vorbedingungen und Effekte sind vorhanden

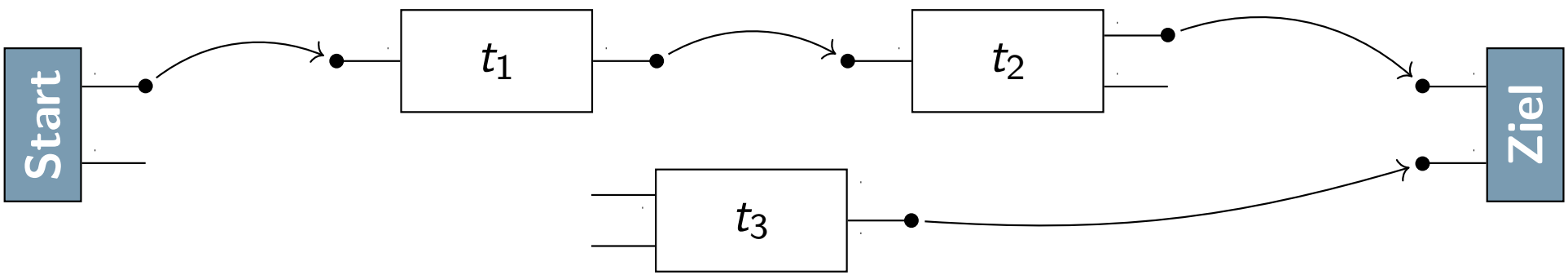
Heuristik Sample-FF

Quelle der NP-Härte: die partielle Ordnung. (Eine) Lösung:

- Sample n Linearisierungen
- Löse die Teilprobleme der Linearisierungen mit der FF-Heuristik

Linearisierungen:

- $t_3 \rightarrow t_1 \rightarrow t_2$
- $t_1 \rightarrow t_3 \rightarrow t_2$
- $t_1 \rightarrow t_2 \rightarrow t_3$



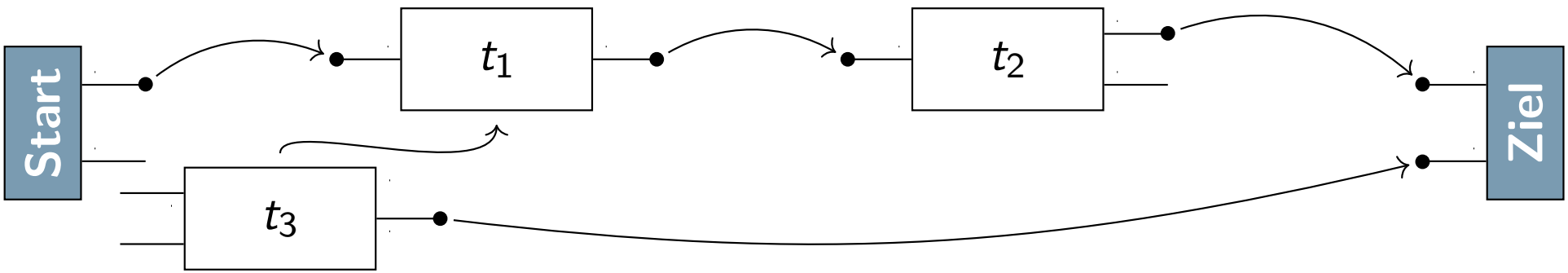
Heuristik Sample-FF

Quelle der NP-Härte: die partielle Ordnung. (Eine) Lösung:

- Sample n Linearisierungen
- Löse die Teilprobleme der Linearisierungen mit der FF-Heuristik

Linearisierungen:

- $t_3 \rightarrow t_1 \rightarrow t_2$
- $t_1 \rightarrow t_3 \rightarrow t_2$
- $t_1 \rightarrow t_2 \rightarrow t_3$



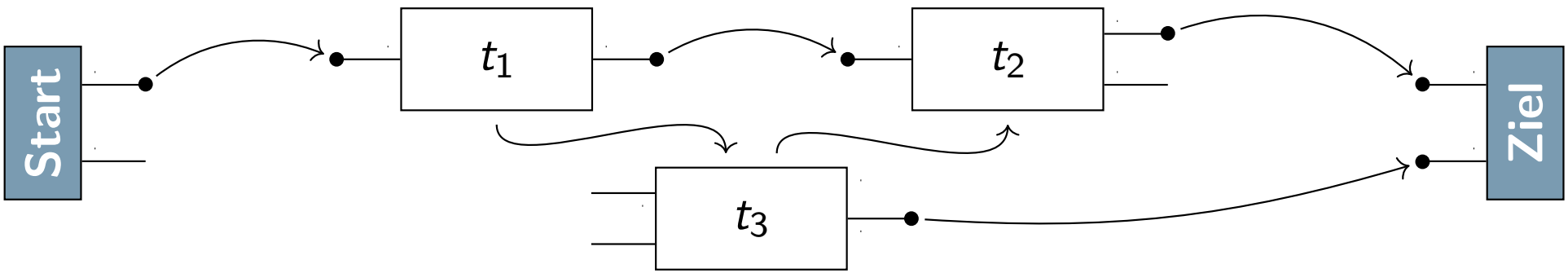
Heuristik Sample-FF

Quelle der NP-Härte: die partielle Ordnung. (Eine) Lösung:

- Sample n Linearisierungen
- Löse die Teilprobleme der Linearisierungen mit der FF-Heuristik

Linearisierungen:

- $t_3 \rightarrow t_1 \rightarrow t_2$
- $t_1 \rightarrow t_3 \rightarrow t_2$
- $t_1 \rightarrow t_2 \rightarrow t_3$



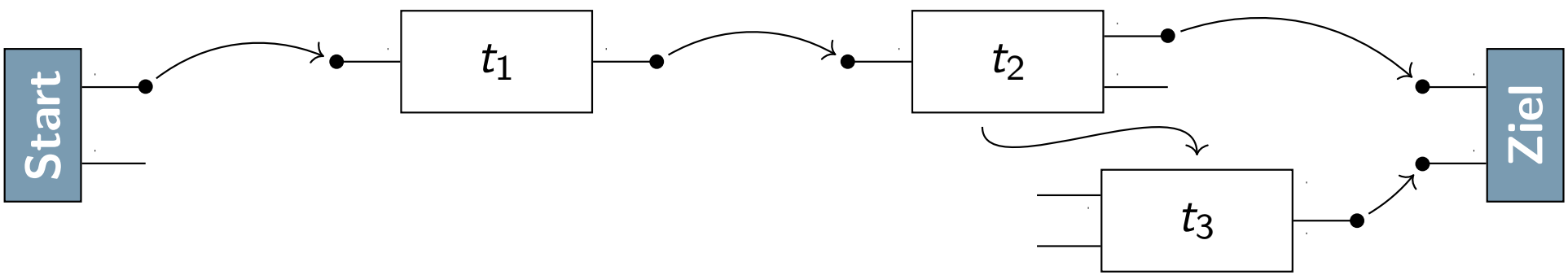
Heuristik Sample-FF

Quelle der NP-Härte: die partielle Ordnung. (Eine) Lösung:

- Sample n Linearisierungen
- Löse die Teilprobleme der Linearisierungen mit der FF-Heuristik

Linearisierungen:

- $t_3 \rightarrow t_1 \rightarrow t_2$
- $t_1 \rightarrow t_3 \rightarrow t_2$
- $t_1 \rightarrow t_2 \rightarrow t_3$



Heuristik Sample-FF: Empirische Evaluation I

Evaluation mit (fast allen) Planungsproblemen der IPC 1 bis IPC 5.

Vergleich von PANDA mit A* und:

- Der **Add-Heuristik für POCL-Planen** (Younes & Simmons)
- Der **Relax-Heuristik** (Nguyen & Kambhampati)
- 12 Varianten der **Sample-FF-Heuristik**

Heuristik Sample-FF: Empirische Evaluation II

Ergebnisse:

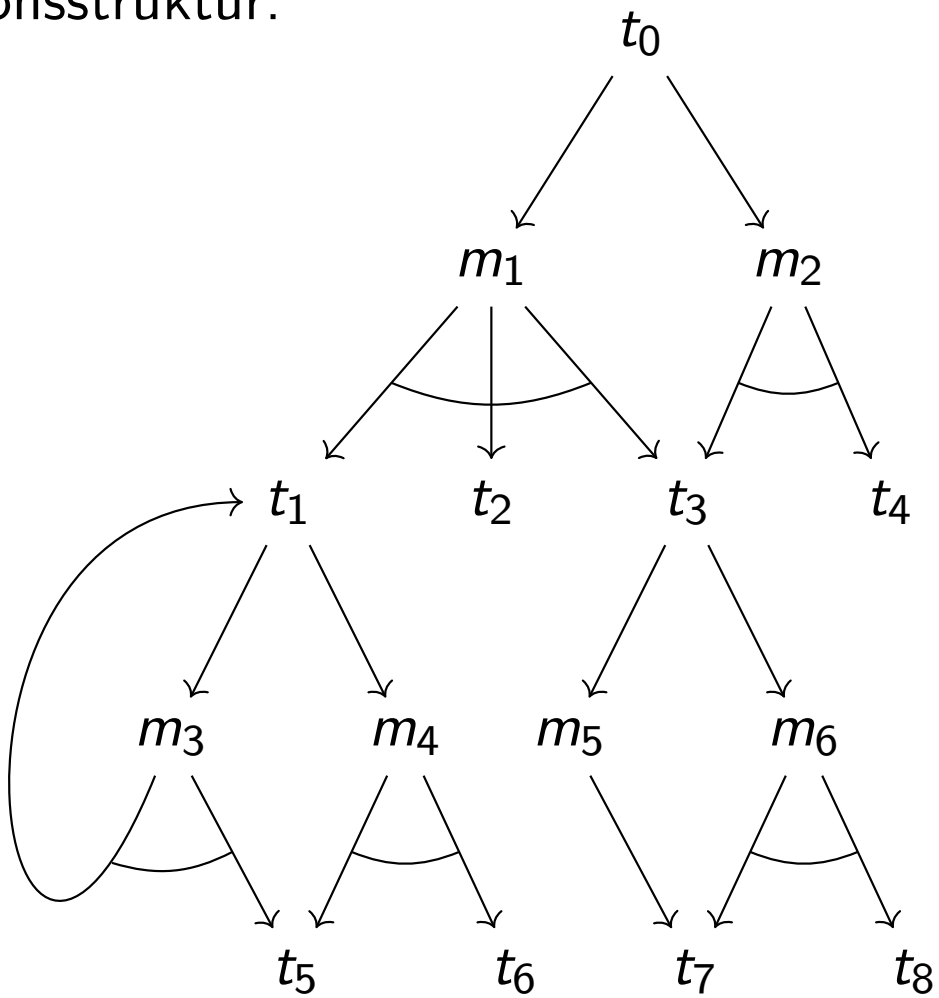
- Anzahl gelöster Probleminstanzen in 15 Minuten (von 446)
  Add: 292, Relax: 194, Sample-FF: zwischen 127 und 187
- Bei einem Planungsproblem konnte dessen Unlösbarkeit nur durch die Sample-FF bewiesen werden
- Es existiert keine (relaxierte) Lösung unter 30 gesampelten Linearisierungen pro Plan für 36 % aller Suchknoten

Heuristik Sample-FF: Empirische Evaluation III

Domäne	n	**Add**	**Relax**	**Sample-FF** front: $\bot$ end: $\bot$ 1	3	10	30	front: $\bot$ end: $\top$ 1	3	10	30	front: $\top$ end: $\top$ 1	3	10	30
grid	5	0	0	0	0	0	0	0	0	0	0	1	**1**	**1**	**1**
gripper	20	14	7	1	1	1	1	1	2	1	1	2	**3**	**3**	2
logistics	20	12	7	**8**	5	6	6	6	7	6	5	0	0	1	1
movie	30	30	30	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**
mystery	20	8	9	10	11	9	9	10	11	10	9	**12**	**12**	11	11
mystery-prime	20	3	3	3	4	**6**	5	5	4	4	4	**6**	**6**	**6**	**6**
blocks	21	4	7	5	5	**6**	**6**	4	3	3	3	5	3	2	0
logistics	28	28	27	22	23	23	**24**	21	19	20	21	15	13	14	15
miconic	100	100	39	**40**	**40**	37	35	39	41	37	32	15	16	18	20
depot	22	2	1	1	1	1	1	0	1	1	1	2	2	**3**	2
driverlog	20	7	7	**11**	9	10	9	9	10	9	8	8	7	9	7
rover	20	20	19	13	14	**15**	**15**	11	11	12	11	7	9	9	9
zeno-travel	10	4	3	3	**5**	4	**5**	4	3	4	4	1	1	1	1
airport	20	18	9	10	**11**	**11**	**11**	7	10	10	10	7	8	6	4
pipesworld-noTankage	10	8	2	2	3	**5**	3	2	1	2	1	1	4	3	**5**
pipesworld-Tankage	10	1	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0
satellite	20	16	7	**7**	5	6	6	5	5	7	5	1	2	3	3
pipesworld	10	1	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0
storage	20	7	4	6	7	9	8	6	7	7	6	9	9	**10**	**10**
tpp	20	19	11	**8**	7	6	7	6	6	6	6	5	6	6	6
total	446	292	194	182	183	187	183	168	173	171	159	127	132	136	133

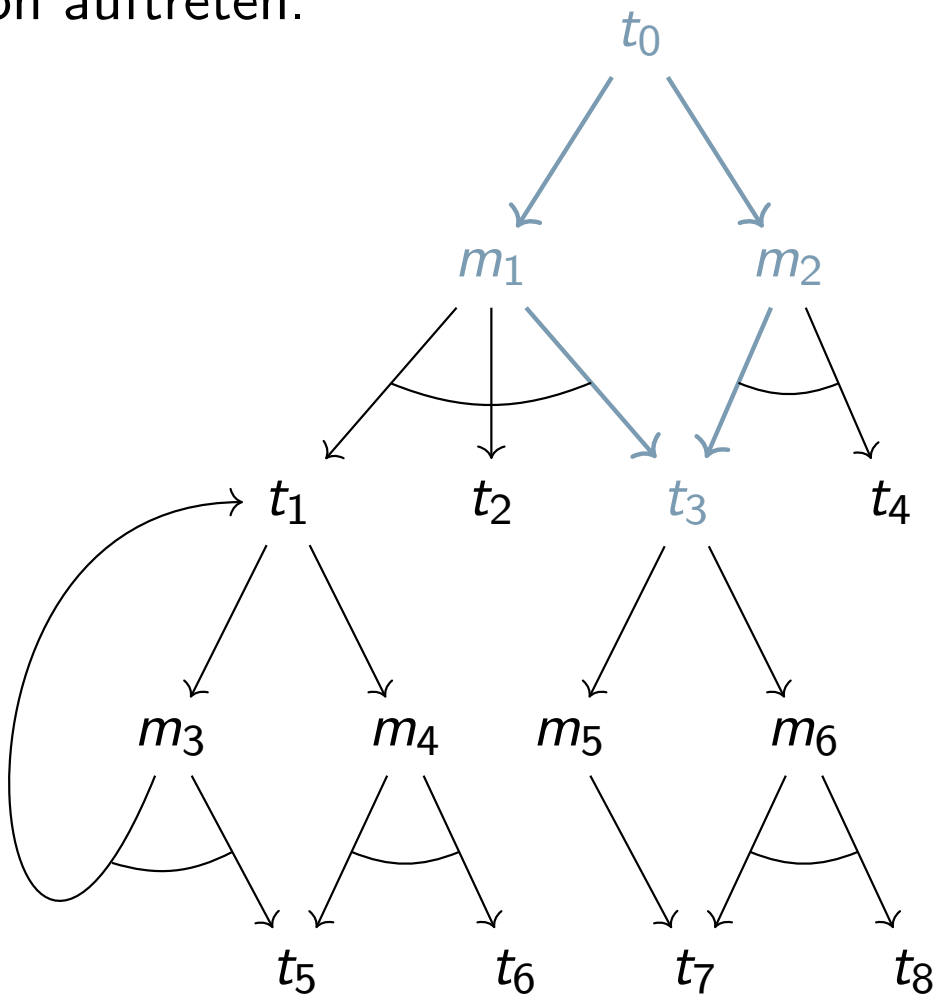
Der Task-Dekompositions-Graph

Der Task-Dekompositions-Graph (TDG) repräsentiert die Dekompositionsstruktur:



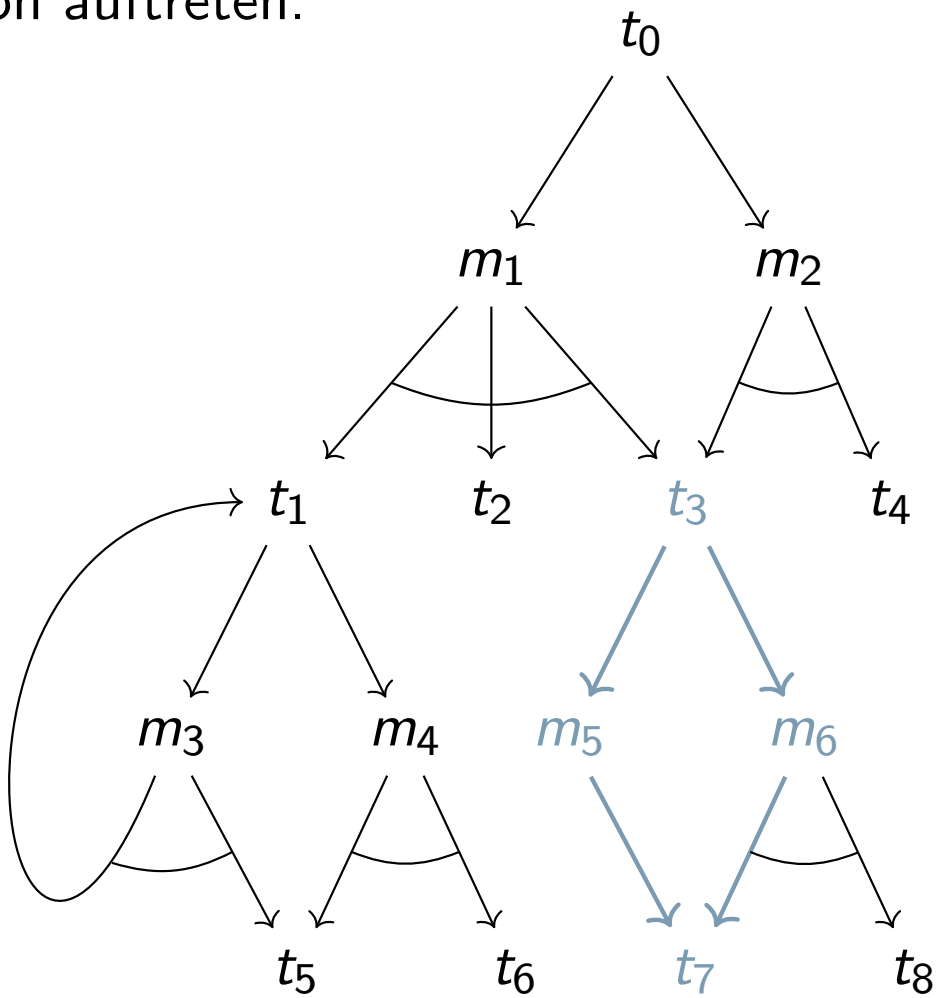
Hierarchische Landmarken

Hierarchische Landmarken sind Aktionen, die bei jeder Dekomposition auftreten.



Hierarchische Landmarken

Hierarchische Landmarken sind Aktionen, die bei jeder Dekomposition auftreten.



Heuristik TDG-c

Sei $\langle V_T, V_M, E_{T\to M}, E_{M\to T}\rangle$ ein TDG.

Die Heuristikwerte (TDG-c) der Taskknoten sind definiert durch:

$$h_T(v_t) := \begin{cases} cost(v_t) & \text{falls } v_t \text{ primitiv} \\ \min\limits_{(v_t,v_m)\in E_{T\to M}} h_M(v_m) & \text{sonst} \end{cases}$$

Für Methodenknoten $v_m = \langle PS, \prec, CL\rangle$:

$$h_M(v_m) := \sum_{(v_m,v_t)\in E_{M\to T}} h_T(v_t)$$

Heuristik TDG-m

Sei $\langle V_T, V_M, E_{T\to M}, E_{M\to T}\rangle$ ein TDG.

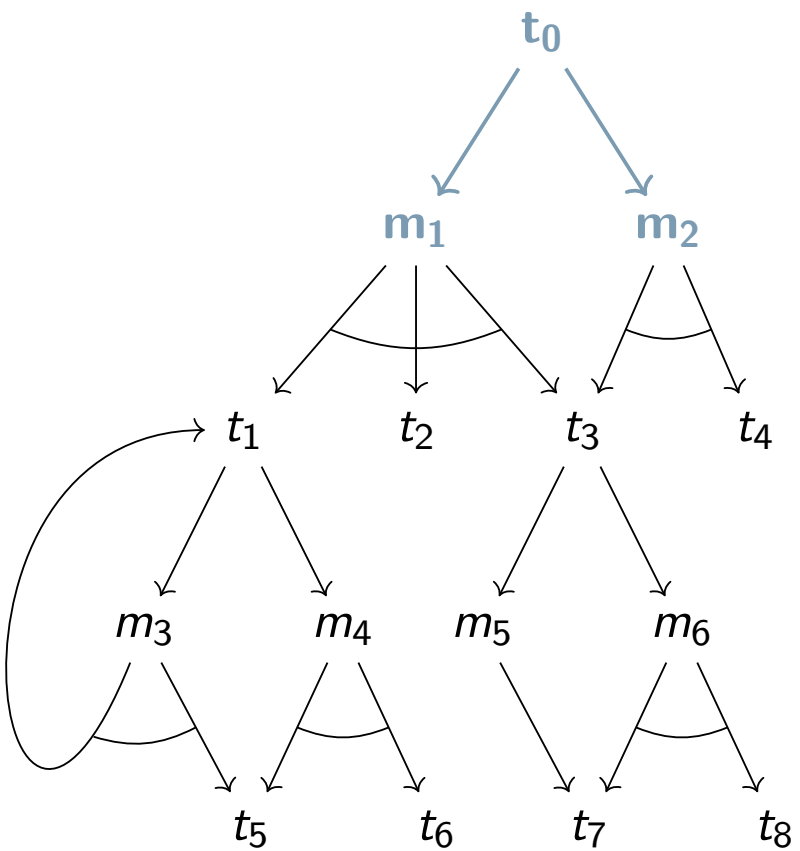
Die Heuristikwerte (TDG-m) der Taskknoten sind definiert durch:

$$h_T(v_t) := \begin{cases} |pre(v_t)| & \text{falls } v_t \text{ primitiv} \\ 1 + \min\limits_{(v_t,v_m)\in E_{T\to M}} h_M(v_m) & \text{sonst} \end{cases}$$

Für Methodenknoten $v_m = \langle PS, \prec, CL\rangle$:

$$h_M(v_m) := \sum_{(v_m,v_t)\in E_{M\to T}} h_T(v_t) - |CL|$$

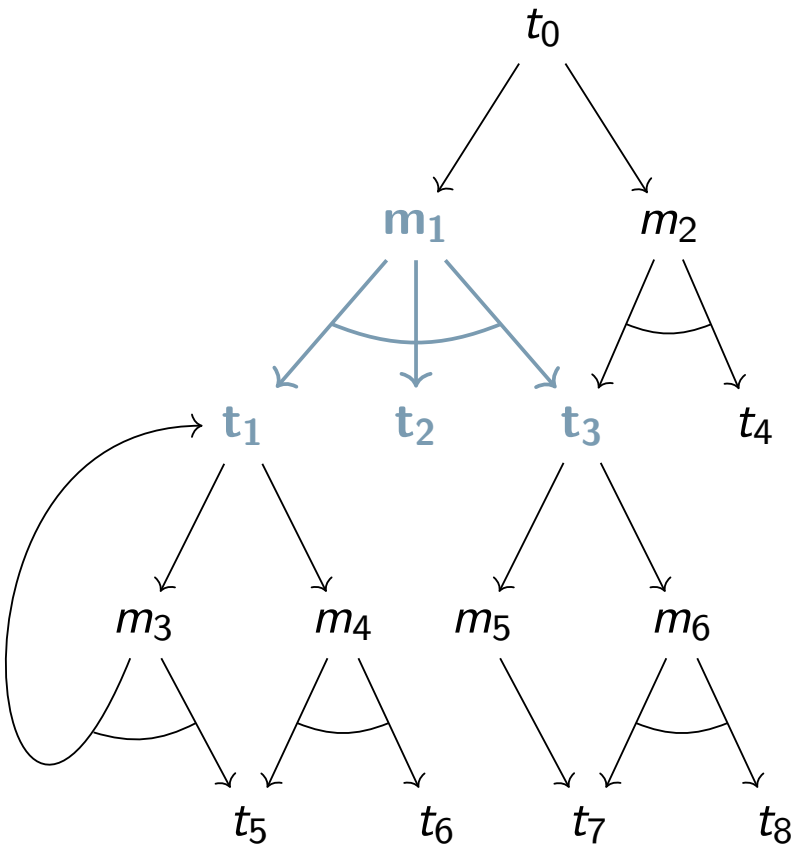
Heuristik TDG-m: Beispiel



Beispiel:

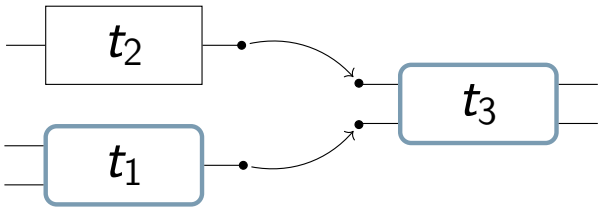
$$h(t_0) = 1 + min\{h(m_1), h(m_2)\}$$

Heuristik TDG-m: Beispiel



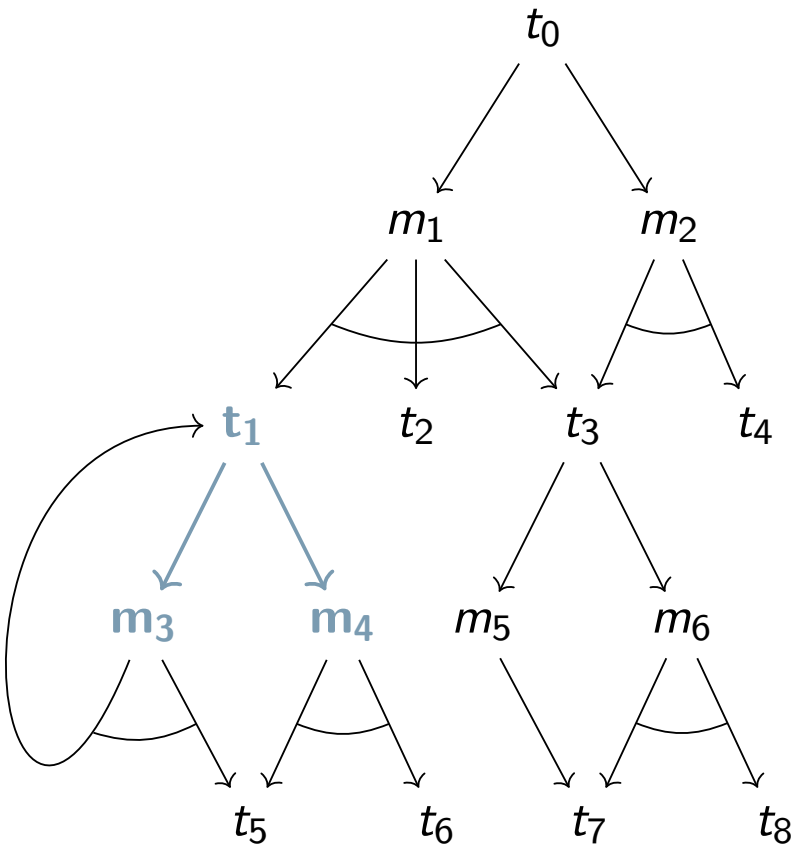
Beispiel:

Methode $m_1 = (t_0, P)$ mit Plan P:



$$h(m_1) = \sum_{t_i \in \{t_1, t_2, t_3\}} h(t_i) - |CL|$$

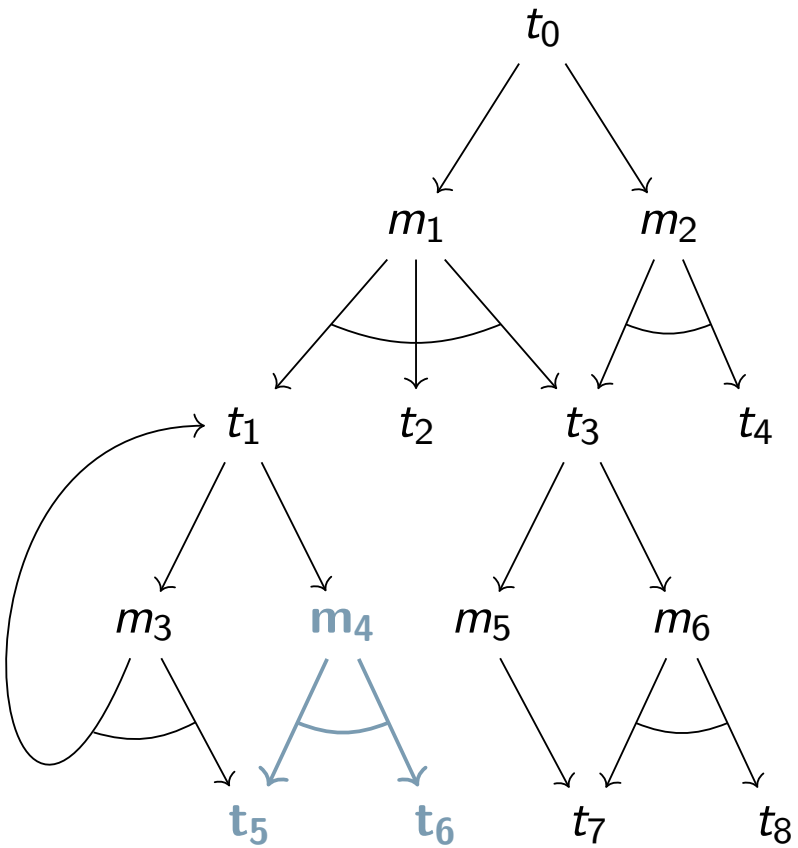
Heuristik TDG-m: Beispiel



Beispiel:

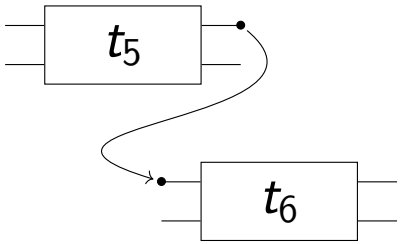
$$h(t_1) = 1 + min\{h(m_3), h(m_4)\}$$

Heuristik TDG-m: Beispiel



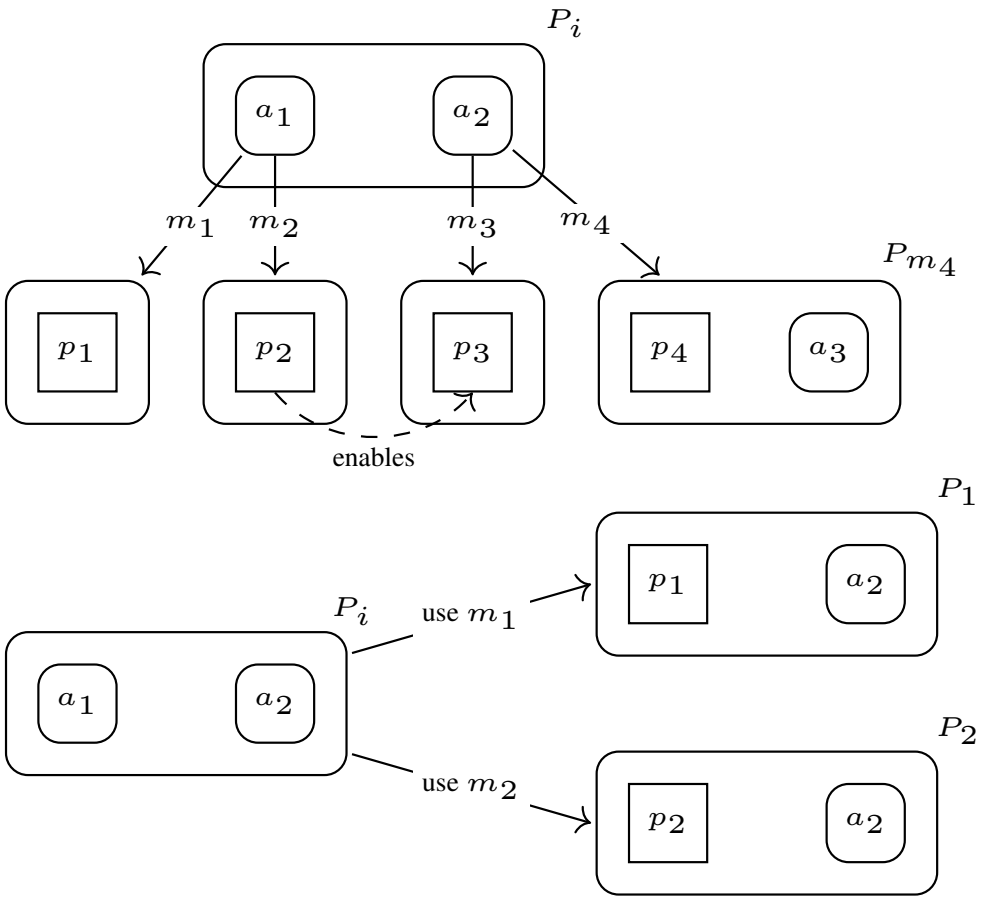
Beispiel:

Methode $m_4 = (t_1, P)$ mit Plan P:



$$\begin{aligned} h(m_4) &= h(t_5) + h(t_6) - |CL| \\ &= |pre(t_5)| + |pre(t_6)| - 1 \\ &= 2 + 2 - 1 = 3 \end{aligned}$$

TDG-Heuristiken: Neuberechnung des TDGs



Oben: TDG mit folgenden Werten:

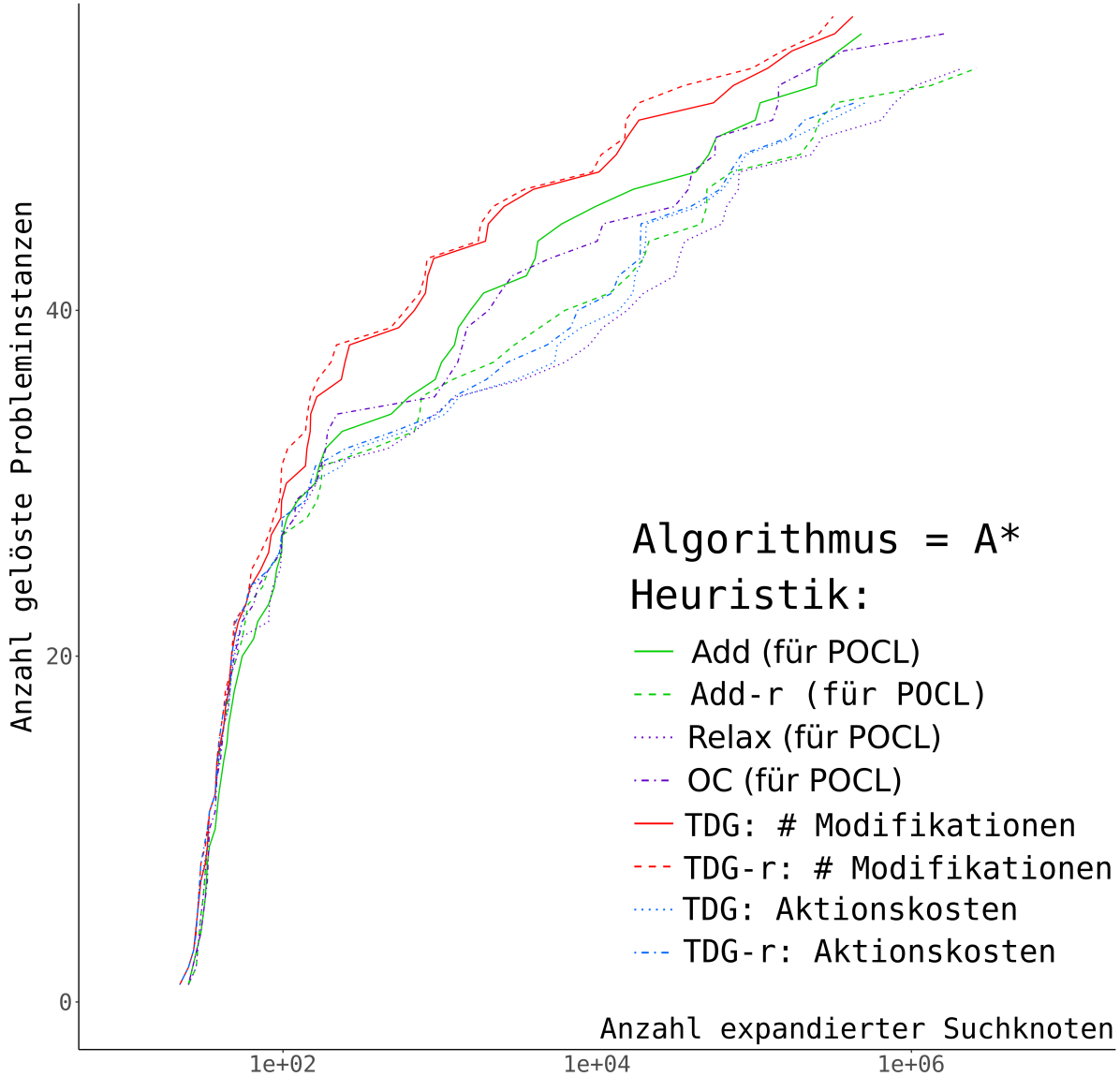
$$cost(p_3) = i$$
$$h_M(P_{m_4}) = h_T(p_4) + h_T(a_3) = j$$
$$j > i$$

Unten: Pläne und deren Heuristiken.

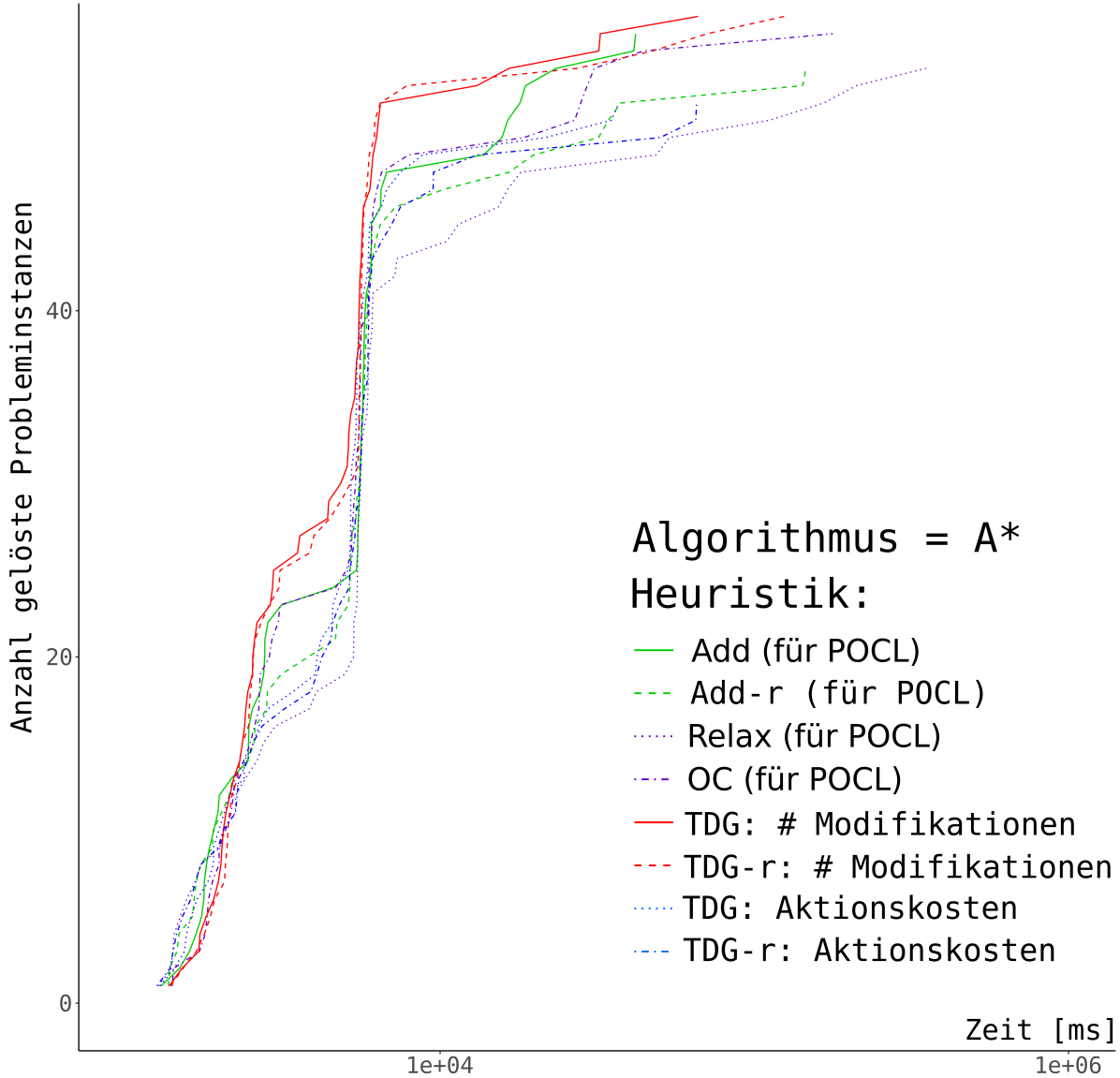
Ohne Neuberechnung gilt:
$h(P_1) = h(P_2) = i$.

Nach Neuberechnung gilt:
$h(P_1) = j$ und $h(P_2) = i$.

TDG-Heuristiken: Empirische Evaluation I



TDG-Heuristiken: Empirische Evaluation I



TDG-Heuristiken: Empirische Evaluation II

	Strategie	UM-Tr. (21 inst.)			SmartPh. (5 inst.)			Satellite (22 inst.)			Woodw. (11 inst.)			Summary (59 inst.)		
		#s	#o	cost	#s	#o	cost	#s	#o	cost	#s	#o	cost	#s	#o	cost
blind	Uniform	21	21	1.00	4	4	1.00	17	17	1.00	8	8	1.00	50	50	**1.00**
	BF	21	21	1.00	4	4	1.00	15	15	1.00	7	7	1.00	47	47	**1.00**
	DF	21	21	1.00	5	1	1.60	19	7	2.09	8	4	1.44	53	33	2.09
Systeme	UMCP$_{BF}$	21	21	1.00	4	4	1.00	15	15	1.00	7	7	1.00	47	47	**1.00**
	UMCP$_{DF}$	21	21	1.00	4	1	1.60	17	6	2.09	6	4	1.29	48	32	2.09
	UMCP$_{h}$	21	21	1.00	5	4	1.40	19	11	1.50	7	7	1.00	52	43	1.50
	Compile	18	18	1.00	5	5	1.00	21	18	1.10	5	5	1.00	49	46	1.10
	Compile$_{opt}$	16	16	1.00	5	5	1.00	9	9	1.00	5	5	1.00	35	35	**1.00**
A*	ADD	21	21	1.00	4	1	1.20	21	21	1.00	10	9	1.17	56	52	1.20
	ADD-r	21	21	1.00	5	5	1.00	19	18	1.08	9	4	1.25	54	48	1.25
	Relax	21	21	1.00	5	5	1.00	18	18	1.00	10	8	1.17	54	52	1.17
	OC	21	21	1.00	4	4	1.00	21	21	1.00	10	7	1.17	56	53	1.17
	TDG-m/(rec)	21	21	1.00	5	5	1.00	22	21	1.31	9	9	1.00	**57**	56	1.31
	TDG-c/(rec)	21	21	1.00	5	5	1.00	18	18	1.00	8	8	1.00	52	52	**1.00**
Greedy-A*	ADD	21	21	1.00	4	0	1.20	21	20	1.09	10	9	1.17	56	50	1.20
	ADD-r	21	21	1.00	5	5	1.00	20	17	1.10	10	4	1.25	56	47	1.25
	Relax	21	21	1.00	5	5	1.00	18	15	1.10	10	4	1.25	54	45	1.25
	OC	21	21	1.00	4	4	1.00	22	21	1.09	10	7	1.22	**57**	53	1.22
	TDG-m/(rec)	21	21	1.00	5	5	1.00	22	17	1.31	9	8	1.08	**57**	51	1.31
	TDG-c	21	21	1.00	5	5	1.00	20	20	1.00	10	10	1.00	56	56	**1.00**
	TDG-c (rec)	21	21	1.00	5	5	1.00	20	20	1.00	11	11	1.00	**57**	**57**	**1.00**

TDG-Heuristiken: Empirische Evaluation III

Einfluss des TDG-Neubauens auf Suchzeit und Suchraum:

Strategy		space			time			space			time			space			time			space			time		
		<	=	>	<	=	>	<	=	>	<	=	>	<	=	>	<	=	>	<	=	>	<	=	>
		UM-Translog						SmartPhone						Satellite						Woodworking					
A^*	TDG-m	2	19	0	1	15	5	1	4	0	1	4	0	22	0	0	0	17	5	7	2	0	1	6	2
	TDG-c	2	19	0	0	13	8	1	4	0	0	4	1	18	0	0	0	10	8	6	2	0	4	3	1
A^*_2	TDG-m	2	19	0	3	16	2	1	4	0	0	4	1	22	0	0	0	13	9	4	5	0	1	6	2
	TDG-c	2	19	0	4	11	6	1	4	0	1	4	0	20	0	0	0	11	9	5	5	0	4	3	3

Beispiel für zeitlich gewinnbringende Reduktionen:

- Suchraum: von 86.935 auf 15.213 (83%) und
  15 Sek auf 8 Sek (46%) für TDG-m (in Woodworking)
- Suchraum: von 2.360 auf 158 (by 93%) und
  5 Sek auf 3 Sek (40%) für TDG-m (selbe Instanz wie oben)

TDG-Heuristiken: Empirische Evaluation IV

Einfluss des Zufalls bei 50 Runs in der Smartphone-Domäne:

Problem / **Strategie**	**#1** μ_s	σ_s/μ_s	μ_t	**#2** μ_s	σ_s/μ_s	μ_t
BF	30	0.14	0	—	—	—
DF	20	0.06	0	164	1.04	1
UMCP-BF	58	0.24	0	375530	0.04	55
UMCP-DF	19	0.08	0	15863	1.45	6
UMCP-H	18	0.00	0	15964	1.21	7
Greedy mit OC	19	0.05	0	1608	0.61	9
Greedy mit TDG-m	18	0.00	0	164	0.30	2

μ_s: Mittelwert explorierter Suchraum
σ_s/μ_s: Relative Standardabweichung
μ_t: Mittelwert benötigte Zeit

TDG-Heuristiken: Empirische Evaluation IV

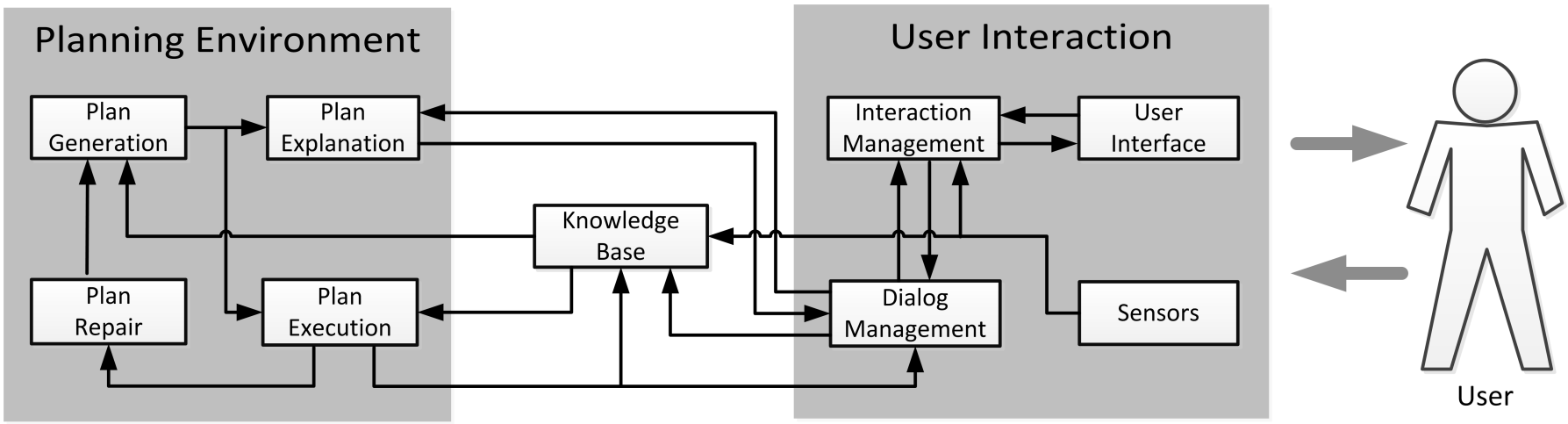
Einfluss des Zufalls bei 50 Runs in der Satellite-Domäne:

Problem / **Strategie**	**2–2–2** μ_s	σ_s/μ_s	μ_t	**3–3–3** μ_s	σ_s/μ_s	μ_t
BF	3012	0.31	8	(16) 1122746	0.25	338
DF	442	0.74	2	239418	0.89	48
UMCP-BF	2165	0.26	5	—	—	—
UMCP-DF	273	0.62	1	581613	0.68	133
UMCP-H	443	0.47	2	513303	0.58	122
Greedy mit OC	620	0.43	3	163326	0.93	37
Greedy mit TDG-m	333	0.83	2	85274	1.45	26

μ_s: Mittelwert explorierter Suchraum
σ_s/μ_s: Relative Standardabweichung
μ_t: Mittelwert benötigte Zeit

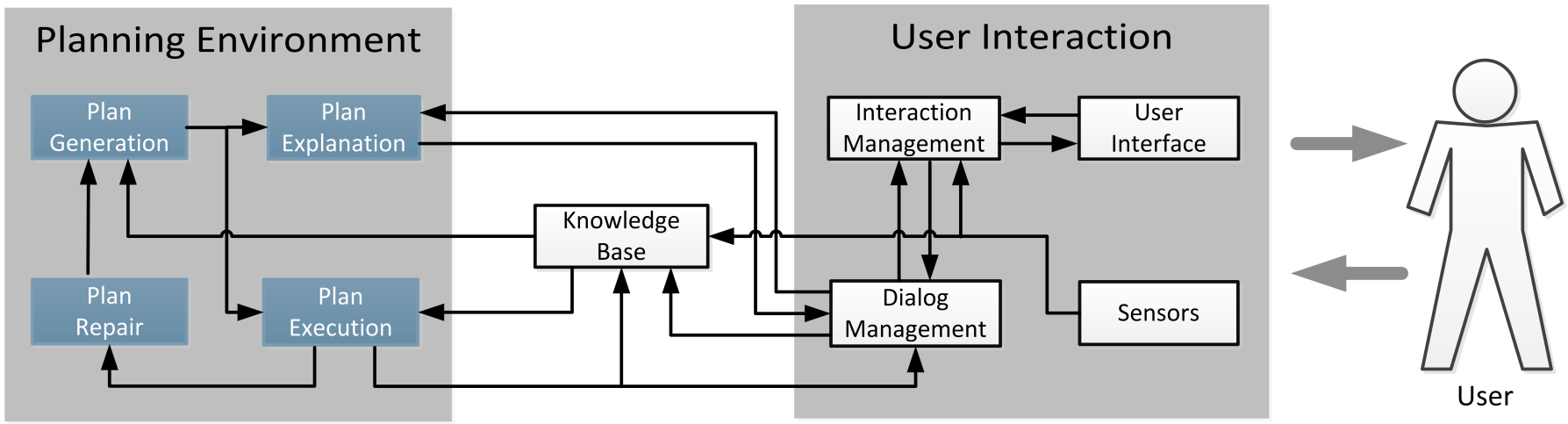
Systemarchitektur zur Nutzerassistenz

Generischer Ansatz basiert auf Zusammenspiel einer Vielzahl von Modulen domänenunabhängiger Technologien:



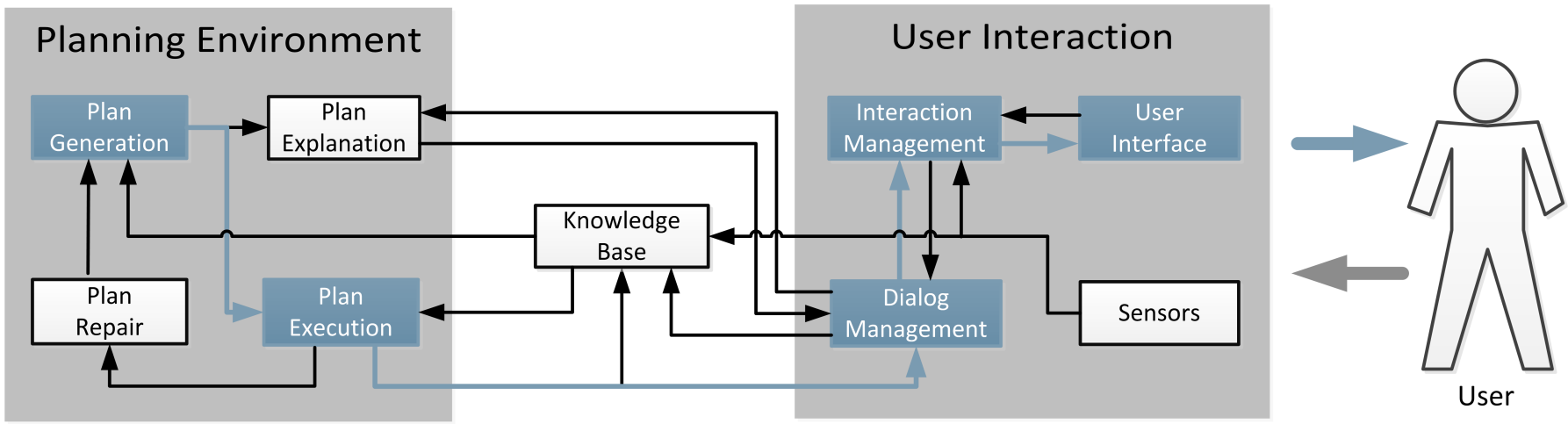
Systemarchitektur zur Nutzerassistenz

Generischer Ansatz basiert auf Zusammenspiel einer Vielzahl von Modulen domänenunabhängiger Technologien:



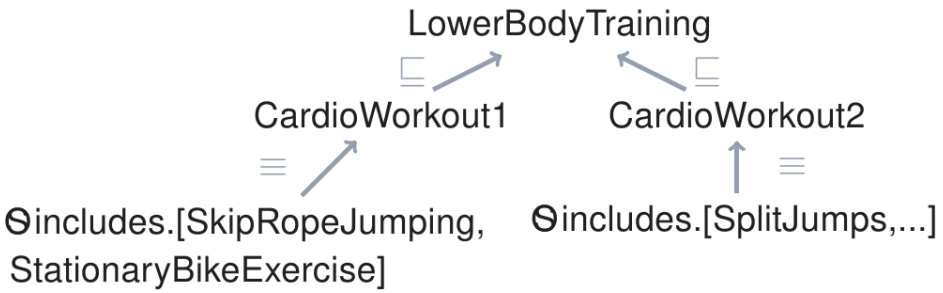
Systemarchitektur zur Nutzerassistenz

Generischer Ansatz basiert auf Zusammenspiel einer Vielzahl von Modulen domänenunabhängiger Technologien:

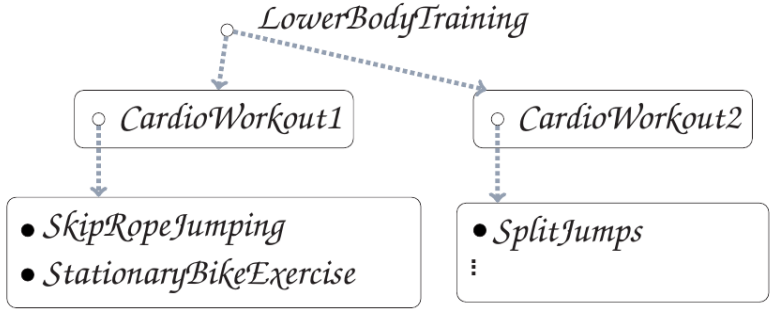


Kohärente Sicht auf die Modelle

Ontologie:



Planungsmodell:



Klassendefinitionen:

$$LowerBodyTraining \equiv \ni includes.[\exists trains.\exists partOf.LowerBody]$$

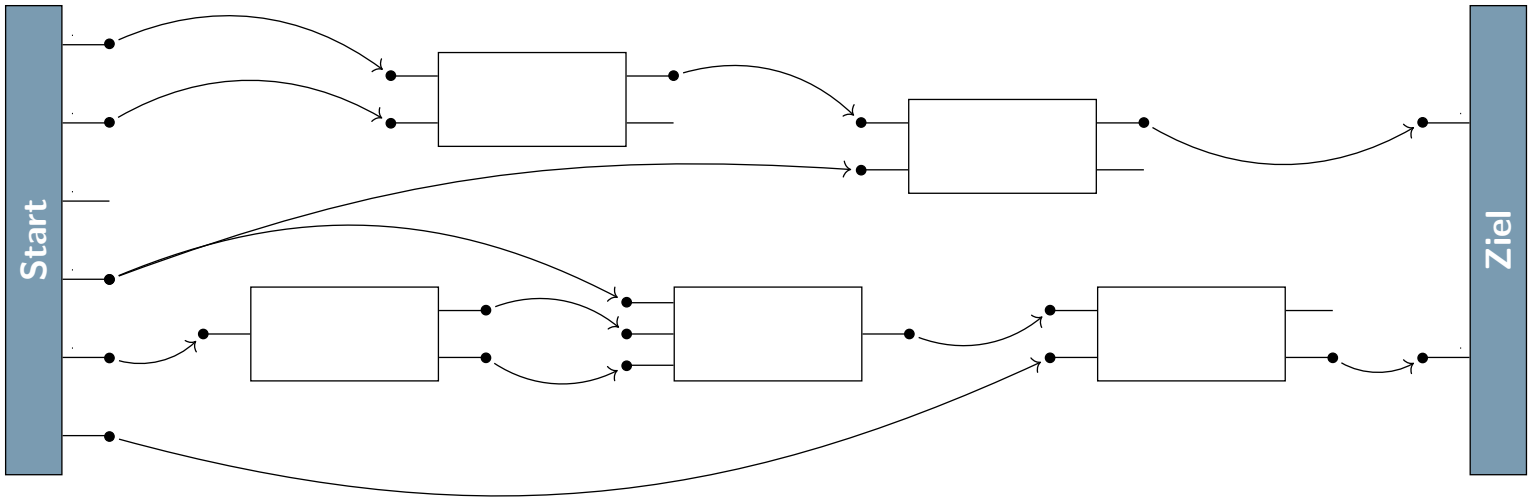
$$CardioWorkout1 \equiv \ni includes.[SkipRopeJumping, StationaryBikeExercise]$$

$$CardioWorkout2 \equiv \ni includes.[SplitJumps, \ldots]$$

$$SkipRopeJumping \sqsubseteq \exists trains.GastrocnemiusMuscle \sqcap \exists engages.QuadricepsFemorisMuscle \sqcap \exists engages.Hamstring$$

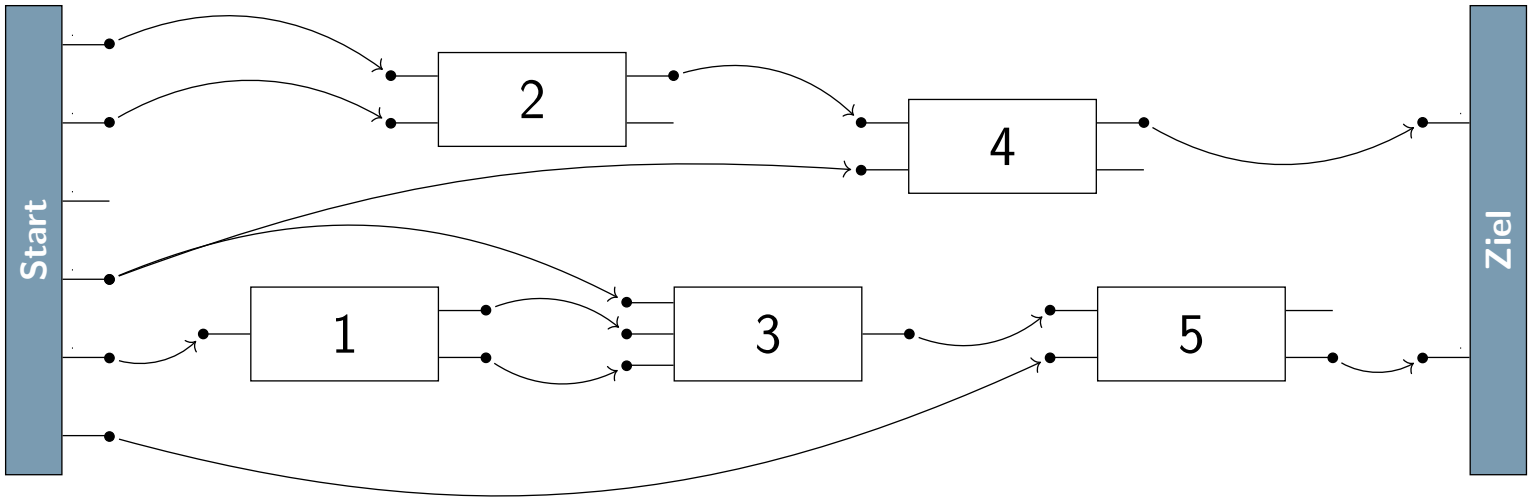
Nutzerfreundliche Planlinearisierung I

Welche Linearisierungen sind für menschliche Nutzer geeignet?



Nutzerfreundliche Planlinearisierung I

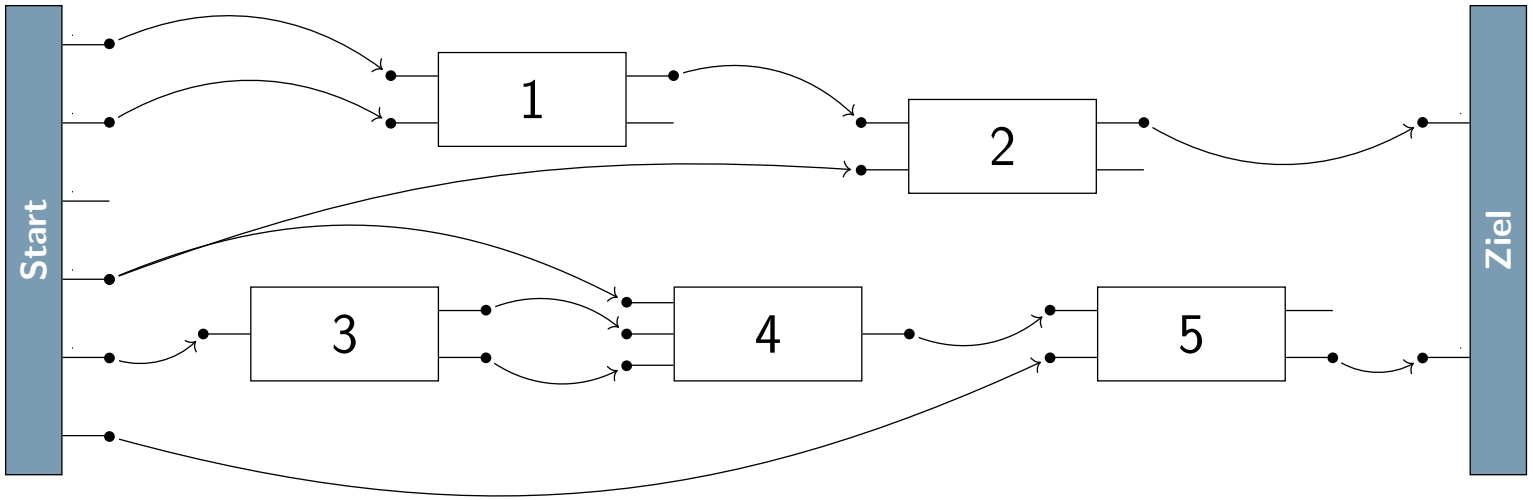
Welche Linearisierungen sind für menschliche Nutzer geeignet?



1: verbinde ...
2: verbinde CINCH-Kabel (erstes Ende) mit Blu-ray-Player
3: verbinde ...
4: verbinde CINCH-Kabel (anderes Ende) mit AV-Verstärker
5: verbinde ...

Nutzerfreundliche Planlinearisierung I

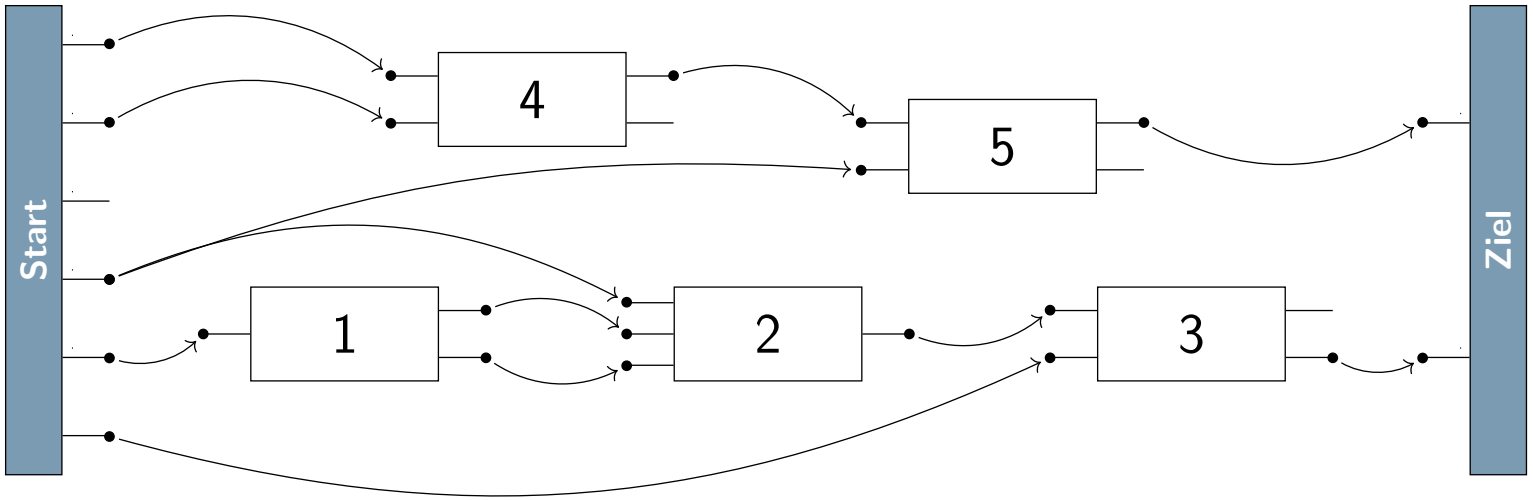
Welche Linearisierungen sind für menschliche Nutzer geeignet?



1: verbinde CINCH-Kabel (erstes Ende) mit Blu-ray-Player
2: verbinde CINCH-Kabel (anderes Ende) mit AV-Verstärker
3: verbinde ...
4: verbinde ...
5: verbinde ...

Nutzerfreundliche Planlinearisierung I

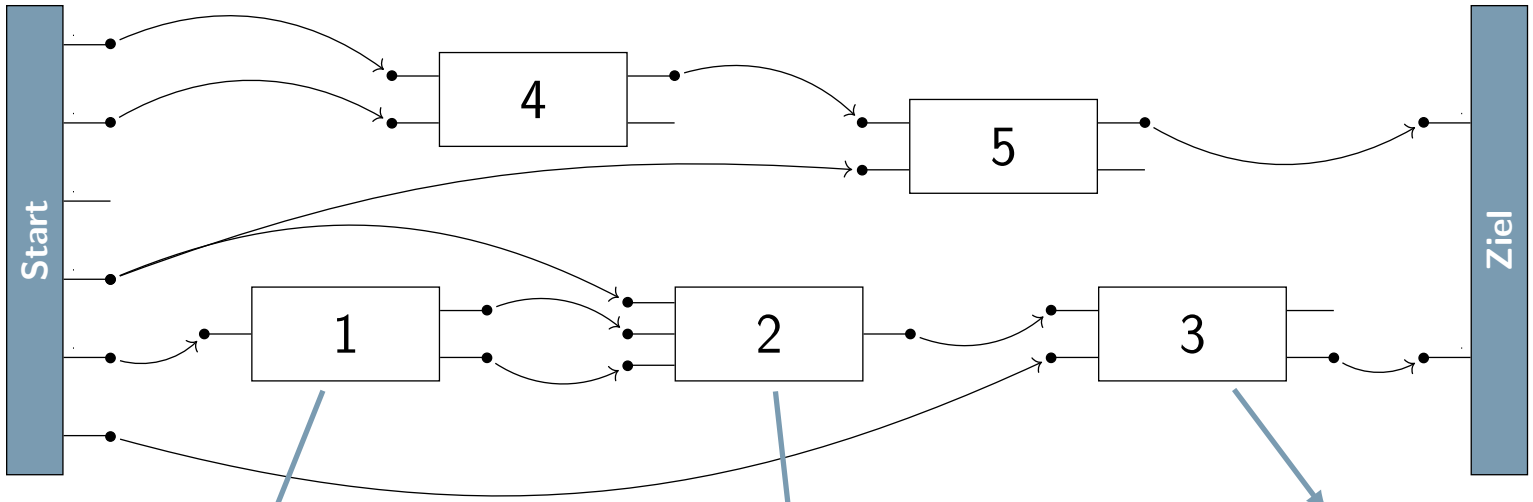
Welche Linearisierungen sind für menschliche Nutzer geeignet?

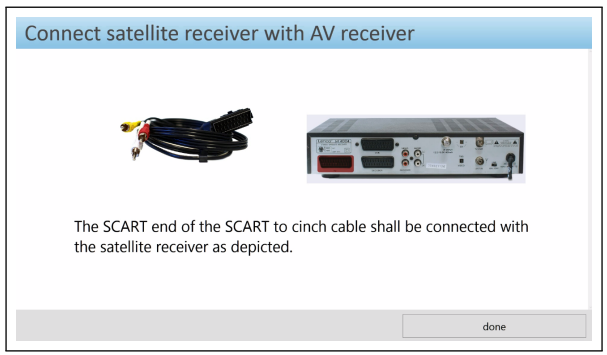


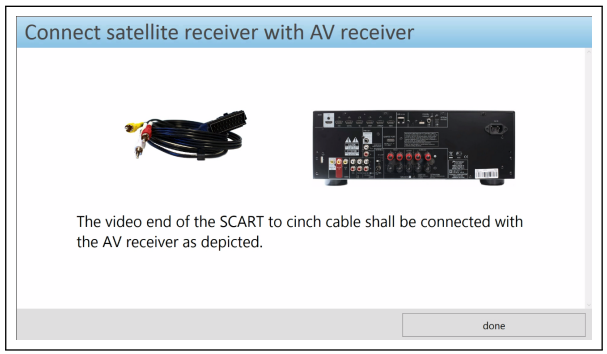
1: verbinde ...
2: verbinde ...
3: verbinde ...
4: verbinde CINCH-Kabel (erstes Ende) mit Blu-ray-Player
5: verbinde CINCH-Kabel (anderes Ende) mit AV-Verstärker

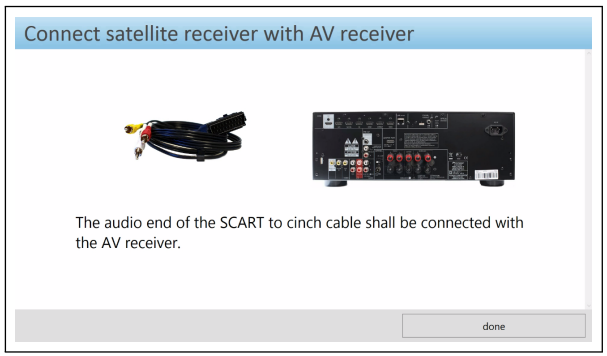
Nutzerfreundliche Planlinearisierung I

Welche Linearisierungen sind für menschliche Nutzer geeignet?









Nutzerfreundliche Planlinearisierung II

Wir stellen drei Planlinearisierungen vor, basierend auf:

- Abstand in der Hierarchie des Modells: Methoden enthalten „zusammengehörige“ Aktionen.
- Anzahl identischer Konstanten: Arbeite Objekte (z.B. Kabel, Geräte) sukzessive ab.
- Anzahl gemeinsamer kausaler Links: Arbeite Aktionen gemäß ihrer Kausalzusammenhänge ab.

Generierung von Planerklärungen I

Nutzerfrage: Welchem Zweck dient diese Aktion? / Wieso sollte ich sie ausführen?

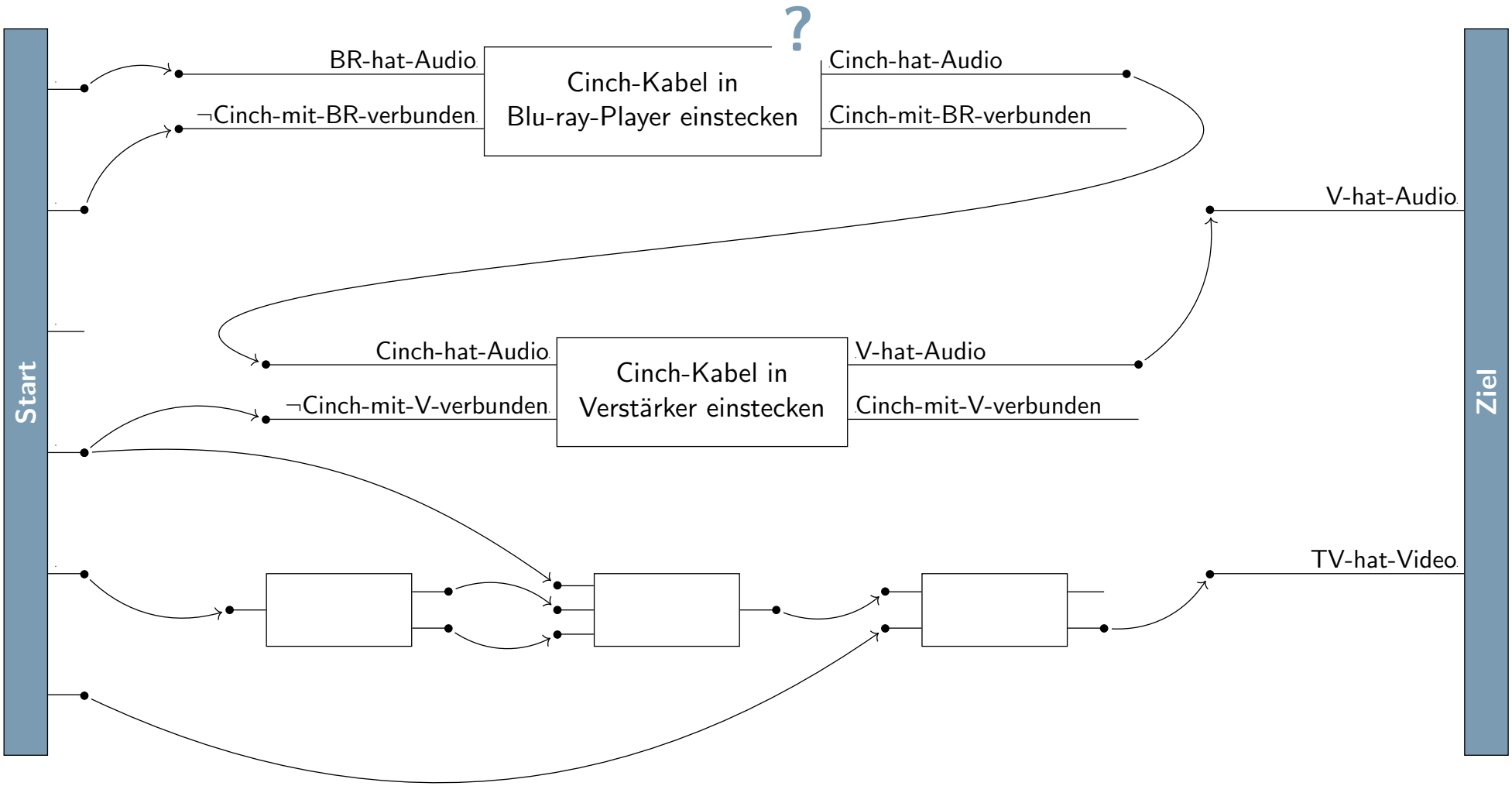
Erklärungen in natürlicher Sprache werden aus formalen Beweisen generiert basierend auf:

- der Kausalstruktur (kausale Links) des Lösungsplans und/oder
- der Dekompositionshierarchie.

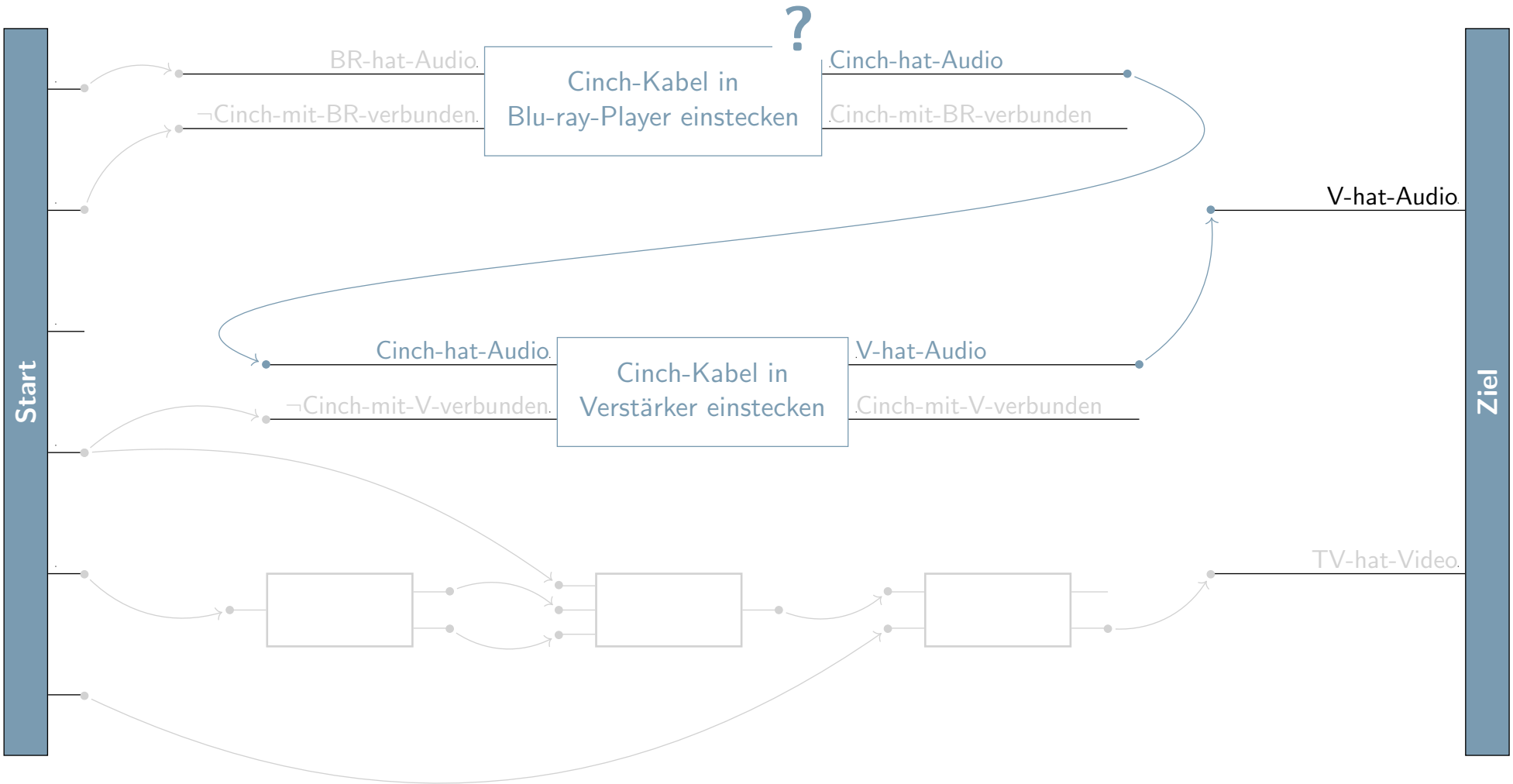
Beispielaxiome:

- $\forall$ actions $a_1, a_2 : CR(a_1, a_2) \wedge N(a_2) \Rightarrow N(a_1)$
- $N(\textbf{Ziel})$

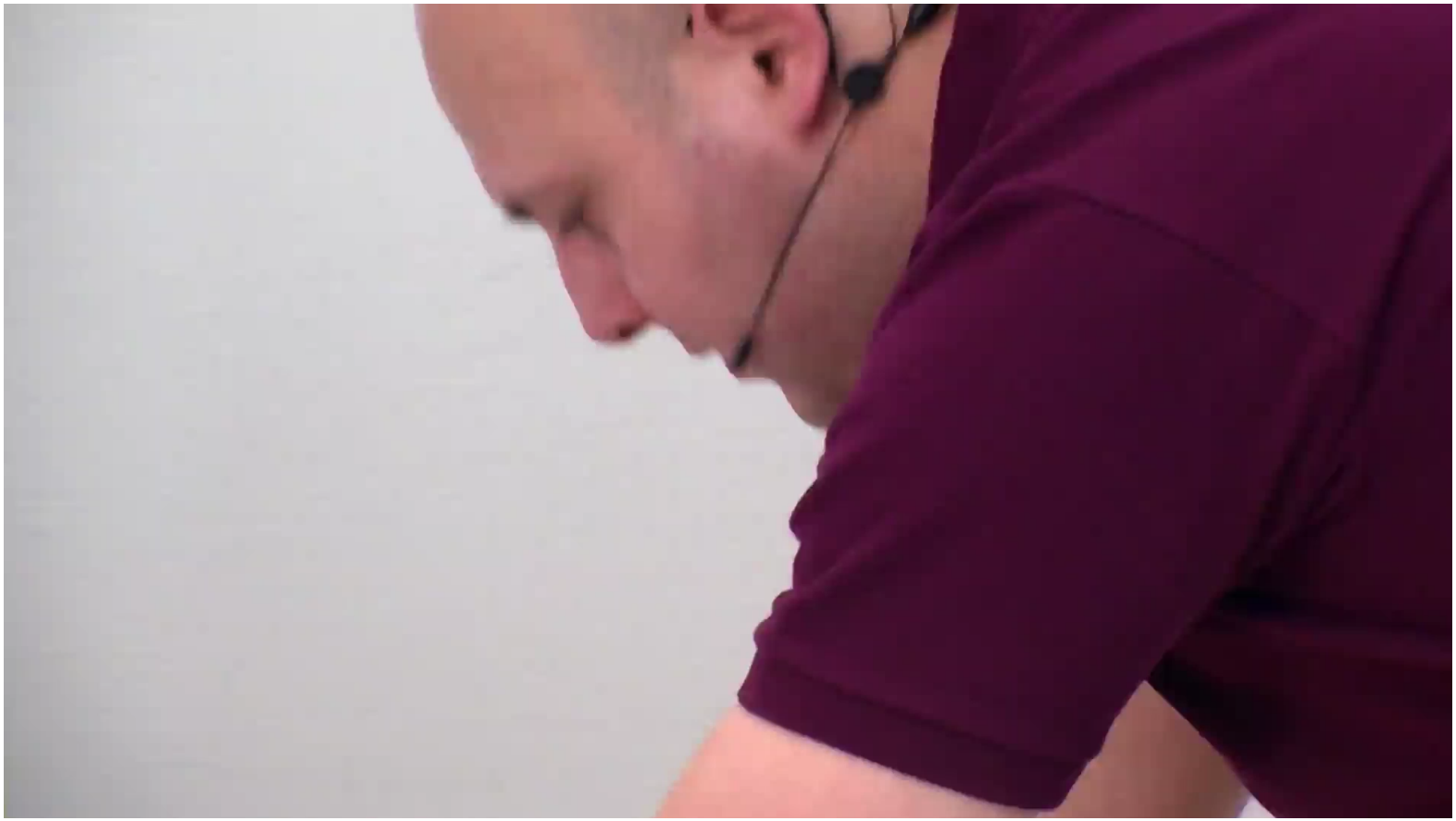
Generierung von Planerklärungen II



Generierung von Planerklärungen II



Generierung von Planerklärungen III



Nutzerstudie des Assistenz-Systems I

Wir haben untersucht:

- Die generelle Akzeptanz eines solchen Assistenzsystems (in der Beispielanwendung Verkabelungsassistent).
- Den Einfluss von Planerklärungen.

  Die Hypothese: *Planerklärungen stärken die Konfidenz der Nutzer, dass die präsentierte Lösung korrekt ist.*

Nutzerstudie des Assistenz-Systems: Experimentaldesign

Versuchsaufbau:

- Verkabelungsaufgabe:
  - Fernseher benötigt Videosignal
  - Audio/Video-Verstärker benötigt Audiosignal
- Probanden bekamen eine Schritt-für-Schritt-Anleitung auf einem iPad präsentiert. Diesen mussten sie folgen.
- Die Studie war als kontrolliertes, randomisiertes Experiment mit 59 Probanden konzipiert.
  - „Treatment“-Gruppe bekam für zwei Schritte eine Planerklärung präsentiert.
  - Die Kontrollgruppe bekam nur die Instruktionen.

Nutzerstudie des Assistenz-Systems: Ergebnisse I

Probanden sollten beurteilen, wie sehr sie sicher sind, dass die präsentierte Lösung korrekt ist.

- Durchschnitt/Standardabweichung auf 5-Punkte-Likert-Skala: 4.50/0.82 (Treatment), 4.66/0.55 (Kontrollgruppe)
  (Der Unterschied war statistisch nicht signifikant)
- Mögliche Ursachen:
  - Sicherheit war bereits sehr hoch (4.66 von 5, Kontrollgruppe)
  - Ungeeigneter Versuchsaufbau: Erklärungen kamen ungefragt; waren nicht zwingend nötig.

Nutzerstudie des Assistenz-Systems: Ergebnisse II

- Die animierten Bilder wurden sehr oft positiv erwähnt.
- Hohe Korrelation zwischen Gesamteindruck und der Beurteilung der eigenen Fähigkeiten: Schlechte Selbsteinschätzung sagt höheres Gefallen voraus.
- Frauen mögen das System mehr als Männer.
- Probanden mit höherem Bildungsgrad empfanden die *Erklärungen* besser.

Nutzerstudie des Assistenz-Systems: Ergebnisse III

Die positivsten Zitate:

- „unterstützt sinnvoll“
- „Ich würde diese Art der Bedienungsanleitung allen bisher vorziehen“
- „Ich finde das Assistenz-System sehr sinnvoll, da auch Leute ohne Kenntnisse die Bedienungsanleitung erfolgreich ausführen können.“
- „Dieses Gerät/Assistent wäre prima für meine Eltern :-)“

Nutzerstudie des Assistenz-Systems: Ergebnisse IV

Angaben zu den Probanden:

- $\leq$ 30 Jahre: 19 weiblich, 27 männlich.
- $\geq$30 Jahre: 3 weiblich, 7 männlich.
- (Alter von drei Probanden wurde nicht angegeben.)
- 26 Probanden hatten einen Uni-Abschluss, 9 machten gerade ihren Doktortitel.
- 7 Probanden hatten kein Abitur.
- 30 Probanden hatten einen technischen oder mathematischen Hintergrund: Informatik/Ingenieurwissenschaften, Naturwissenschaften, Mathematik.