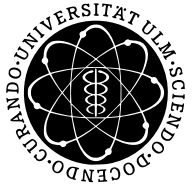
Realzeitfähiges Hybrides Planen

Pascal Bercher

Institute of Artificial Intelligence

22.10.2012



ulm university universität ulm

Motivation

Unterstützung von Menschen in Ihrem Alltag durch Planungstechnologie; Beispiele:

- Bedienung von technischen Geräten wie Smartphones[1]
- Unterstützung bei der Einrichtung einer komplexen Heimkinoanlage (vgl. Demonstrationsszenario 1)

Das Planen *für Menschen* erfordert:

- schnelle Berechnung der Handlungsempfehlungen
  → Realzeitfähigkeit
- nachvollziehbares Welt-/Planmodell
  → Hierarchische Domänenbeschreibung

[1] Susanne Biundo, Pascal Bercher, Thomas Geier, Felix Müller, and Bernd Schattenberg (2011). "Advanced user assistance based on AI planning". In: *Cognitive Systems Research* 12.3-4, pp. 219–236.

Inhaltsverzeichnis

Problemformalisierung

Hierarchische und hybride Planungsprobleme werden beschrieben durch ein Tupel von Domäne und Probleminstanz $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$.

- Die Domäne $\mathcal{D}$ beschreibt die Weltgesetze
- Die Probleminstanz $\mathcal{I}$ beschreibt den vorliegenden Weltzustand

Problemformalisierung (Domäne)

Die Domäne $\mathcal{D} = \langle V, A, C, M \rangle$ besteht aus:

- V, einer endlichen Menge von Zustandsvariablen
  2^V ist die Menge aller Zustände: der Zustandsraum

Beispiel:

$v_1 := \mathit{IsCompatible}(\mathit{AMPLIFIER}, \mathit{cable\_HDMI}) \in V$
$v_2 := \mathit{HasFreePort}(\mathit{AMPLIFIER}) \in V$
$v_3 := \mathit{IsConnected}(\mathit{AMPLIFIER}, \mathit{cable\_HDMI}) \in V$

$\{v_{i_1}, \ldots, v_{i_{42}}\} \in 2^V$ ist ein Zustand

Problemformalisierung (Domäne)

Die Domäne $\mathcal{D} = \langle V, A, C, M \rangle$ besteht aus:

- A, einer endlichen Menge von Aktionen (primitiven Tasks)
  Eine Aktion $\langle pre, add, del \rangle \in A$ besteht aus:
  - $pre \subseteq V$ ist die Vorbedingung,
  - $add, del \subseteq V$ sind die Effekte (Add-List, Delete-List)
- C, einer endlichen Menge von komplexen Tasks

Beispiel:

$v_1 := IsCompatible(AMPLIFIER, cable\_HDMI) \in V$
$v_2 := HasFreePort(AMPLIFIER) \in V$
$v_3 := IsConnected(AMPLIFIER, cable\_HDMI) \in V$

$a_1 := connect(AMPLIFIER, cable\_HDMI) \in A$
$a_1 = \langle \underbrace{\{v_1, v_2\}}_{pre}, \underbrace{\{v_3\}}_{add}, \underbrace{\{v_2\}}_{del} \rangle$

Problemformalisierung (Domäne)

Die Domäne $\mathcal{D} = \langle V, A, C, M \rangle$ besteht aus:

- A, einer endlichen Menge von Aktionen (primitiven Tasks)
  Eine Aktion $\langle pre, add, del \rangle \in A$ besteht aus:
  - $pre \subseteq V$ ist die Vorbedingung,
  - $add, del \subseteq V$ sind die Effekte (Add-List, Delete-List)
- C, einer endlichen Menge von komplexen Tasks

Beispiel:

$c_1 := setUpHomeTheater \in C$

$c_2 := connect(BLU\text{-}RAY, AMPLIFIER) \in C$

$c_3 := connect(TV, AMPLIFIER) \in C$

$c_4 := connect(RECEIVER, AMPLIFIER) \in C$

Problemformalisierung (Domäne)

Die Domäne $\mathcal{D} = \langle V, A, C, M \rangle$ besteht aus:

- M, einer endlichen Menge von Dekompositionsmethoden
  Eine Methode $\langle c, P \rangle \in M$ bildet einen komplexen Task $c \in C$ auf einen partiellen Plan $P = \langle PS, \prec, \alpha \rangle$ ab mit:
  - PS, einer endlichen Menge an Planschritten
  - $\prec \subseteq PS \times PS$, einer partiellen Ordnung auf PS
  - $\alpha : PS \rightarrow C \cup A$, der Taskzuweisung

Beispiel:

$\langle setUpHomeTheater, \langle PS_{23}, \prec_{23}, \alpha_{23} \rangle \rangle \in M$ mit:

- $PS_{23} := \{ps_1, ps_2, ps_3\}$
- $\prec_{23} := \emptyset$
- $\alpha_{23} := \{(ps_1, connect(BLU\text{-}RAY, AMPLIFIER)),$
  $(ps_2, connect(TV, AMPLIFIER)),$
  $(ps_3, connect(RECEIVER, AMPLIFIER))\}$

Problemformalisierung (Probleminstanz)

Die Probleminstanz $\mathcal{I} = \langle s_{init}, c_{init}, goal \rangle$ besteht aus:

- $s_{init} \in 2^V$, dem Initialzustand
- $c_{init} \in C$, dem initialen, komplexen Task
- $goal \subseteq V$, der Zielzustandsbeschreibung

Beispiel:

$v_n, \ldots, v_m := IsCompatible(AMPLIFIER, cable\_HDMI), \ldots$
$v_i, \ldots, v_j := HasFreePort(AMPLIFIER), \ldots$
$v_k, \ldots, v_l := IsConnected(AMPLIFIER, BLU\text{-}RAY), \ldots$

$\mathcal{I} := \langle \{v_n, \ldots, v_m, v_i, \ldots, v_j\}, setUpHomeTheater, \{v_k, \ldots, v_l\} \rangle$

Lösungskriterien

Die Lösung eines Planungsproblems $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$ mit $\mathcal{I} = \langle s_{init}, c_{init}, goal \rangle$ ist ein Plan $P = \langle PS, \prec, \alpha \rangle$, für den gilt:

- P enthält nur Aktionen (d.h. keine komplexen Tasks)
- Für jede durch $\prec$ induzierte Aktionssequenz $\bar{a}$ gilt:
  - $\bar{a}$ ist ausführbar
  - $\bar{a}$ erzeugt einen Zustand s, s.d. $goal \subseteq s$
- P ist eine Verfeinerung von c_{init}:
  - Anwendung von Dekompositionsmethoden (Ersetzung eines komplexen Tasks durch den vordefinierten partiellen Plan)
  - Einfügung von Ordnungsbeziehungen
  - Einfügung von primitiven Tasks **(→ Hybrides Planen!)**

Lösungskriterien

Die Lösung eines Planungsproblems $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$ mit $\mathcal{I} = \langle s_{init}, c_{init}, goal \rangle$ ist ein Plan $P = \langle PS, \prec, \alpha \rangle$, für den gilt:

- P enthält nur Aktionen (d.h. keine komplexen Tasks)
- Für jede durch $\prec$ induzierte Aktionssequenz $\bar{a}$ gilt:
  - $\bar{a}$ ist ausführbar
  - $\bar{a}$ erzeugt einen Zustand s, s.d. $goal \subseteq s$
- P ist eine Verfeinerung von c_{init}:
  - Anwendung von Dekompositionsmethoden (Ersetzung eines komplexen Tasks durch den vordefinierten partiellen Plan)
  - Einfügung von Ordnungsbeziehungen
  - Einfügung von primitiven Tasks **(→ Hybrides Planen!)**

Zusammenfassung

- Hybrider Planungsformalismus $= \begin{cases} \text{Hierarchisches Planen} \\ + \text{ Taskeinfügung} \end{cases}$
- Formalisierung basiert auf Mengensemantik

Komplexitätsergebnisse

Exkurs: (Un)Entscheidbarkeit

Eine Menge M heißt entscheidbar, falls für jedes Objekt o in endlicher Zeit entschieden werden kann ob $o \in M$.

Sei PLAN-EXISTENCE $:= \{\pi' \mid$ Es existiert eine Lösung für $\pi'\}$

Ist PLAN-EXISTENCE entscheidbar? D.h.: kann man für jedes Planungsproblem in endlicher Zeit herausfinden, ob es eine Lösung besitzt?

Entscheidbarkeit im Hierarchischen Planen

Theorem:
Hierarchisches Planen ist *nicht* entscheidbar.[2]

Beweisidee:
Reduktion des Grammatikschnittproblems auf hierarchisches Planen: Gegeben zwei kontextfreie Grammatiken G und G', existiert ein Wort in $L(G) \cap L(G')$?

Dazu: Konstruiere ein Planungsproblem π, s.d. *jede* Lösung von π zu einem Wort in $L(G) \cap L(G')$ korrespondiert. $\square$

⇝ Anhang

[2] Kutluhan Erol, James A. Hendler, and Dana S. Nau (1994). "HTN Planning: Complexity and Expressivity". In: *AAAI 1994*, pp. 1123–1128.

Entscheidbarkeit im Hybriden Planen

Theorem

Falls das hybride Planungsproblem π eine Lösung hat, so existiert eine Lösung, mit beschränkter Länge. Es folgt: Hybrides Planen ist entscheidbar.[3]

Beweisidee

1. Jeder komplexe Task muss nur ein mal expandiert werden, da fehlende Tasks eingefügt werden können
   $\Rightarrow$ beschränkte maximale Tiefe der Dekomposition
2. Zur Gewährleistung der Ausführbarkeit müssen nur beschränkt viele Aktionen eingefügt werden

$\Rightarrow$ endlicher Suchraum $\Rightarrow$ Entscheidbarkeit!

$\rightsquigarrow$ Anhang

[3] Thomas Geier and Pascal Bercher (2011). "On the Decidability of HTN Planning with Task Insertion". In: *IJCAI 2011*, pp. 1955–1961.

Zusammenfassung

Hierarchisches Planen:

- Nicht entscheidbar! (genauer: semi-entscheidbar)
- Kein immer terminierender Algorithmus möglich

Hybrides Planen:

- Entscheidbar! (PSPACE-schwierig und enthalten in EXPSPACE)
- Immer terminierender Algorithmus möglich

Realzeitfähiges Hybrides Planen

Das Planungsproblem wird gelöst durch *Suche im Planraum*.
Ein Suchbaum wird inkrementell aufgebaut:

- Die Knoten sind partielle Pläne
- Der Wurzelknoten ist der initiale Plan
- Die Kanten sind Plan-Modifikationen/Verfeinerungen:
  - Dekomposition/Methodenanwendung
  - Einfügung von Ordnungsbeziehungen
  - Taskeinfügung (nur im hybriden Planen)

Dieser Suchbaum sollte so klein wie möglich werden!

Algorithmus

Wahl der Plan-Modifikationen/Verfeinerungen:

- Jede Verletzung eines Lösungskriteriums wird durch einen *Flaw* repräsentiert.

  Beispiel:
  Ein partieller Plan P enthält zwei komplexe Tasks.
  $\rightarrow$ Dann gibt es zwei *Complex-Task-Flaws*

- Für jeden Flaw-Typ gibt es Modifikationen, die ihn beheben.

  Beispiel:
  Complex-Task-Flaw $\rightarrow$ wende Dekompositionsmethoden an

Algorithmus

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

while $\texttt{Fringe} \neq \emptyset$ **do**
  $P := \textbf{PlanSel}(\texttt{Fringe})$
  $F := \mathit{FlawDet}(P)$
  if $F = \emptyset$ **then return** P
  $f := \textbf{FlawSel}(P)$
  $\texttt{Fringe} := (\texttt{Fringe} \backslash \{P\}) \cup \mathit{Successors}(P, f)$
end
return `fail`

Algorithmus (Übersicht)

Zusammenfassung:

1. Selektiere einen partiellen Plan P
2. Berechne sämtliche Flaws $F(P)$
3. Selektiere einen Flaw $f \in F(P)$
4. Behebe diesen Flaw f und erzeuge alle Nachfolgepläne $succ_1(P, f), \ldots, succ_n(P, f)$

$\Rightarrow$ Zwei Entscheidungspunkte:

- Planselektion **PlanSel**() $\rightarrow$ Domänentransformation
- Flawselektion **FlawSel**() $\rightarrow$ Landmarkentechnik

Planselektion

Aufgabe der Planselektion:
Selektiere einen partiellen Plan mit möglichst geringem "Abstand" zu einer Lösung. Dies übernehmen Heuristiken!

Sei π ein Planungsproblem und P ein partieller Plan. Dann schätzt die Heuristikfunktion h_π den "Abstand" $h_\pi(P)$ von P zu einer Lösung.

Bekannte Heuristiken:

- Hierarchisches Planen:
  - praktisch keine
- Nicht-Hierarchisches Planen:
  - Suche im Planraum: → nur 2 Zieldistanzschätzungen
  - Suche im Zustandsraum: → sehr gute Zieldistanzschätzungen

Heuristiken zur Planselektion

Idee:
Nutze bestehende Heuristiken des nicht-hierarchischen Planens.[4]

Problem:
Bestehende Heuristiken sind für Suche im Zustandsraum, d.h. sie schätzen $h_\pi(s)$, s Zustand, statt $h_\pi(P)$, P partieller Plan.

Lösung:
Transformiere einen Plan P in einen Zustand!

[4] Pascal Bercher and Susanne Biundo (2012). "A Heuristic for Hybrid Planning with Preferences". In: *FLAIRS 2012*, pp. 120–123.

Problemstellung

Hierarchisches/hybrides Planen:

- Problemstellung:
  $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$ mit $\mathcal{D} = \langle V, A, C, M \rangle$ und $\mathcal{I} = \langle s_{init}, c_{init}, goal \rangle$
- Algorithmus:
  Schätze $h_\pi(P)$, P partieller Plan

Nicht-Hierarchisches Planen:

- Problemstellung:
  $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$ mit $\mathcal{D} = \langle V, A \rangle$ und $\mathcal{I} = \langle s_{init}, goal \rangle$
- Algorithmus:
  Schätze $h_\pi(s)$, s Zustand

Beschränkung auf nicht-hierarchisches Planen!

Domänen-/ Problemtransformation

Gegeben:

- $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$ mit $\mathcal{D} = \langle V, A \rangle$ und $\mathcal{I} = \langle s_{init}, goal \rangle$
- ein partieller Plan P

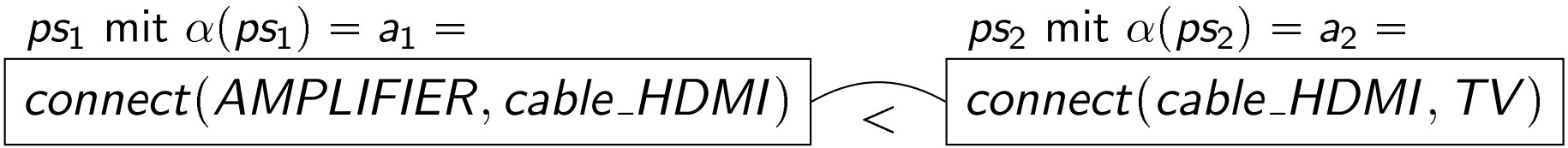
Gesucht:

Transformation, die den partiellen Plan P eliminiert.

- $\pi' = \langle \mathcal{D}', \mathcal{I}' \rangle$ mit $\mathcal{D}' = \langle V', A' \rangle$ und $\mathcal{I}' = \langle s'_{init}, goal' \rangle$
- Zusätzlich muss gelten:
  - π besitzt eine Lösung $\Rightarrow$ π' besitzt eine Lösung
  - P' ist Lösung von $\pi' \Rightarrow$ P' kodiert Verfeinerung von P
  - P' ist Lösung von $\pi' \Rightarrow$ P' kodiert Lösung für π

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P



Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P

ps_1 mit $\alpha(ps_1) = a_1 =$ $connect(AMPLIFIER, cable\_HDMI)$ $<$ ps_2 mit $\alpha(ps_2) = a_2 =$ $connect(cable\_HDMI, TV)$

Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P

ps_1 mit $\alpha(ps_1) = a_1 =$ $connect(AMPLIFIER, cable\_HDMI)$ $<$ ps_2 mit $\alpha(ps_2) = a_2 =$ $connect(cable\_HDMI, TV)$

Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P

ps_1 mit $\alpha(ps_1) = a_1 =$ $connect(AMPLIFIER, cable\_HDMI)$ $<$ ps_2 mit $\alpha(ps_2) = a_2 =$ $connect(cable\_HDMI, TV)$

Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P

ps_1 mit $\alpha(ps_1) = a_1 =$ $connect(AMPLIFIER, cable\_HDMI)$ $<$ ps_2 mit $\alpha(ps_2) = a_2 =$ $connect(cable\_HDMI, TV)$

Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P

ps_1 mit $\alpha(ps_1) = a_1 =$ $connect(AMPLIFIER, cable\_HDMI)$ $<$ ps_2 mit $\alpha(ps_2) = a_2 =$ $connect(cable\_HDMI, TV)$

Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Domänen-/ Problemtransformation (Beispiel)

Sei $\pi = \langle \underbrace{\langle V, A \rangle}_{\mathcal{D}}, \underbrace{\langle s_{init}, goal \rangle}_{\mathcal{I}} \rangle$ gegeben sowie der partielle Plan P

ps_1 mit $\alpha(ps_1) = a_1 =$ $connect(AMPLIFIER, cable\_HDMI)$ $<$ ps_2 mit $\alpha(ps_2) = a_2 =$ $connect(cable\_HDMI, TV)$

Dann ist $\pi' = \langle \overbrace{\langle V', A' \rangle}^{\mathcal{D}'}, \overbrace{\langle s_{init}, goal' \rangle}^{\mathcal{I}'} \rangle$ mit:

- $V' := V \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$
- $A' := A \cup \{enc(ps_1), enc(ps_2)\}$ mit
  - $enc(ps_1) = \langle pre(a_1) \wedge \neg occ\text{-}ps_1, effects(a_1) \wedge occ\text{-}ps_1 \rangle$
  - $enc(ps_2) = \langle pre(a_2) \wedge \neg occ\text{-}ps_2 \wedge occ\text{-}ps_1, effects(a_2) \wedge occ\text{-}ps_2 \rangle$
- $goal' := goal \cup \{occ\text{-}ps_1, occ\text{-}ps_2\}$

Zusammenfassung

Gegeben:

- $\pi = \langle \mathcal{D}, \mathcal{I} \rangle$ mit $\mathcal{D} = \langle V, A \rangle$ und $\mathcal{I} = \langle s_{init}, goal \rangle$
- ein partieller Plan P

Gesucht:
Heuristische Schätzung $h_\pi(P)$

Lösung:
Berechne $\pi' = \langle \mathcal{D}', \mathcal{I}' \rangle$ und setze $h_\pi(P) := h'_{\pi'}(s_{init})$, wobei $h'_{\pi'}$ eine beliebige Heuristik für Suche im Zustandsraum.

Flawselektion (Übersicht)

Wiederholung Algorithmus:

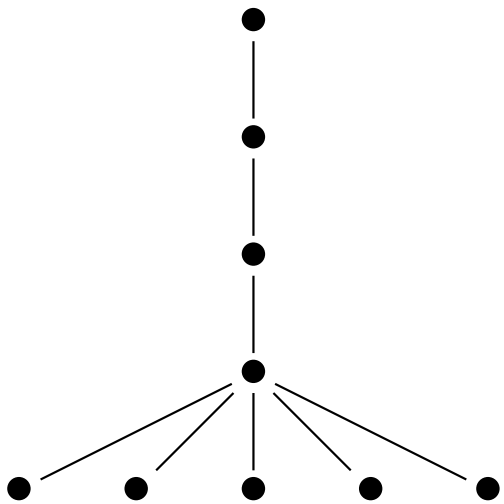
1. Selektiere einen partiellen Plan P
2. Berechne sämtliche Flaws $F(P)$
3. Selektiere einen Flaw $f \in F(P)$
4. Behebe diesen Flaw f und erzeuge alle Nachfolgepläne $succ_1(P, f), \ldots, succ_n(P, f)$

$\rightarrow$ Die Wahl des Flaws kann nicht “falsch” sein!

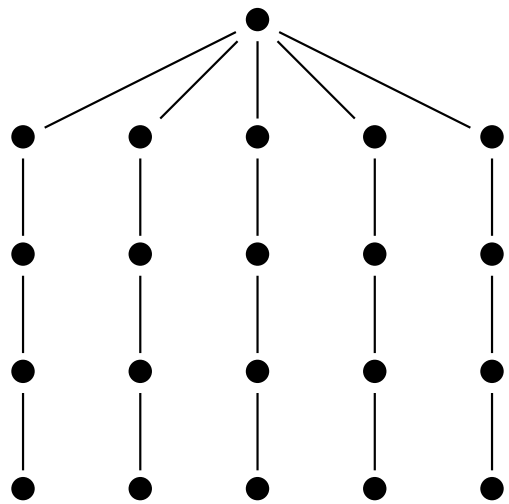
Flawselektion (Motivation)

Die Wahl des Flaws beeinflusst die Form des Suchbaums!

Beispiel:
Der initiale Plan hat vier Flaws: $f_1, \ldots, f_4$.



$f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow f_4$



$f_4 \rightarrow f_1 \rightarrow f_2 \rightarrow f_3$

Flawselektion für Hierarchisches Planen

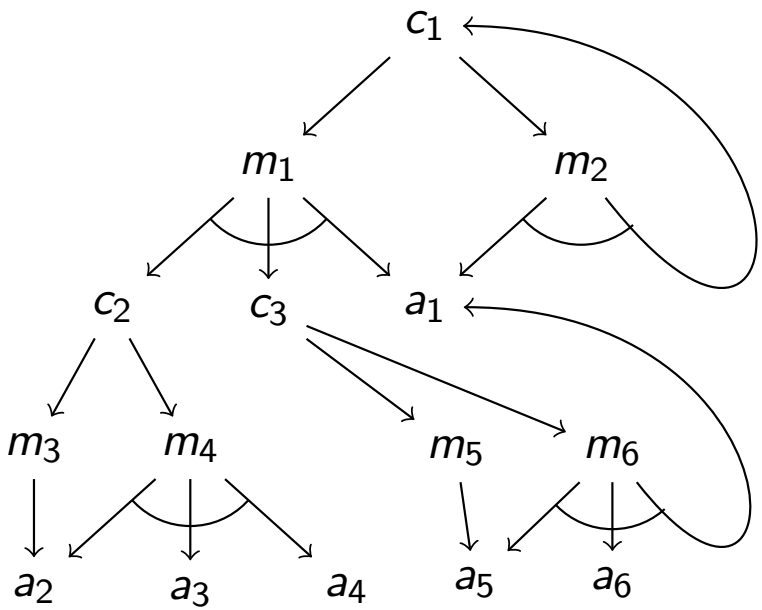
Flawselektion für komplexe Tasks:
Auswahl des komplexen Tasks hängt ab von geschätztem Verzweigungsfaktor. Dieser beruht auf Landmarken.[5]

Definition:
Eine *hierarchische Landmarke* ist ein Task, der auf jedem Pfad vom initialen Plan zu einer Lösung vorkommt.

[5] Mohamed Elkawkagy, Pascal Bercher, Bernd Schattenberg, and Susanne Biundo (2012). "Improving Hierarchical Planning Performance by the Use of Landmarks". In: *AAAI 2012*, pp. 1763–1769.

Dekompositionsgraph & Landmarkentabelle

Dekompositionsgraph:



$a_1, \ldots, a_6 \in A$,
$c_1, \ldots, c_3 \in C$,
$m_1, \ldots, m_6 \in M$

Landmarkentabelle:

t	$N(t)$	$R(t)$
c_1	$\{a_1\}$	$\{\{c_2, c_3\}, \{c_1\}\}$
c_2	$\{a_2\}$	$\{\emptyset, \{a_3, a_4\}\}$
c_3	$\{a_5\}$	$\{\emptyset, \{a_6, a_1\}\}$

- Repräsentiert den Dekompositionsgraphen
- Enthält die notwendigen Tasks $N(t)$ von t
- Enthält die restlichen Tasks $R(t)$ von t

Landmarkenstrategien $\mathrm{lm}_1, \mathrm{lm}_2$

Der partielle Plan P enthalte die komplexen Tasks c_1 und c_3.
Die Landmarkentabelle LT enthalte die folgenden Einträge:

- $\langle\ c_1\ ,\ \{a_1\}\ ,\ \{\ \{c_2, c_3\}\ ,\ \{c_1\}\ \}\ \rangle \in LT$
- $\langle\ c_3\ ,\ \{a_5\}\ ,\ \{\ \emptyset\ ,\ \{a_6, a_1\}\ \}\ \rangle \in LT$

Definition:
Die Flawselektion lm_1 selektiert c_i vor c_j gdw.

$$\sum_{r \in R(c_i)} |r| \ < \sum_{r \in R(c_j)} |r|$$

$$\sum_{r \in R(c_3)} |r| \ < \sum_{r \in R(c_1)} |r| \ \Leftrightarrow 0 + 2 < 2 + 1 \Rightarrow \text{wähle } c_3!$$

Landmarkenstrategien $\mathrm{lm}_1, \mathrm{lm}_2$

Der partielle Plan P enthalte die komplexen Tasks c_1 und c_3.
Die Landmarkentabelle LT enthalte die folgenden Einträge:

- $\langle\ c_1\ ,\ \{a_1\}\ ,\ \{\ \{c_2, c_3\}\ ,\ \{c_1\}\ \}\ \rangle \in LT$
- $\langle\ c_3\ ,\ \{a_5\}\ ,\ \{\ \emptyset\ ,\ \{a_6, a_1\}\ \}\ \rangle \in LT$

Definition:
Die Flawselektion lm_2 selektiert c_i vor c_j gdw.
$\sum_{r \in R(c_i)} |r|_C < \sum_{r \in R(c_j)} |r|_C$ mit $|r|_C := |\{c \in r \mid c \in C\}|$.

$\sum_{r \in R(c_3)} |r|_C < \sum_{r \in R(c_1)} |r|_C \Leftrightarrow 0 + 0 < 2 + 1 \Rightarrow$ wähle c_3!

Landmarkenstrategien $\mathrm{lm}_1^*, \mathrm{lm}_2^*$

Die transitive Hülle der Strategien traversiert den Graphen bis zur primitiven Ebene!

$$R^*(t) = R(t) \cup \bigcup_{r \in R(t)} \left(\bigcup_{t' \in r} R^*(t') \right)$$

$$\text{mit } \langle t, N(t), R(t) \rangle \in LT$$

t	$N(t)$	$R(t)$	$R^*(t)$
c_1	$\{a_1\}$	$\{\{c_2, c_3\}, \{c_1\}\}$	$R(c_1) \cup R(c_2) \cup R(c_3)$
c_2	$\{a_2\}$	$\{\emptyset, \{a_3, a_4\}\}$	$R(c_2)$
c_3	$\{a_5\}$	$\{\emptyset, \{a_6, a_1\}\}$	$R(c_3)$

Evaluation

Besser als andere evaluierte Flaw-Selektionsstrategien!

Evaluation wurde durchgeführt mit:

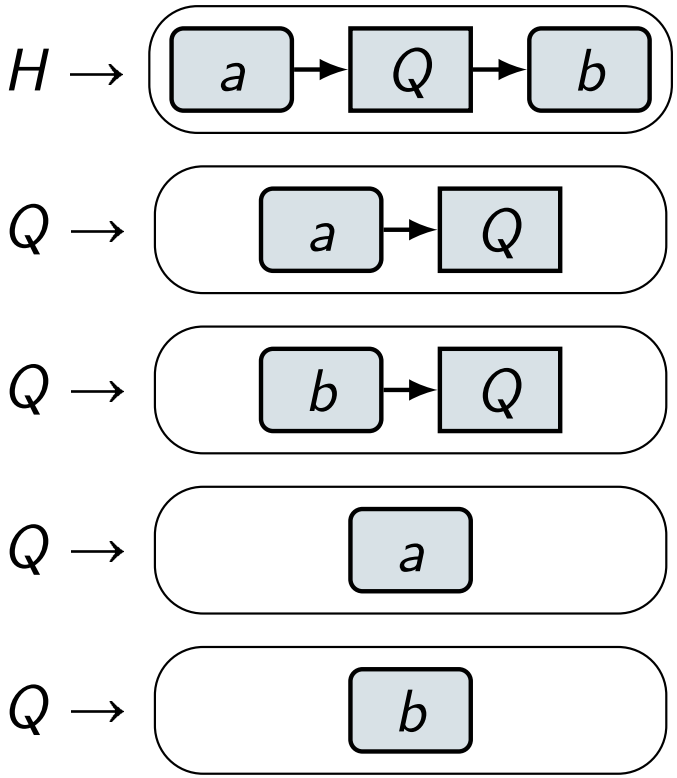
- nicht aktueller Version von PANDA (anderer Algorithmus!)
- vorgestellter Algorithmus beruht auf neuer Version PANDA_2

⇝ Anhang

Zusammenfassung

- Neue Formalisierung von hierarchischem und hybriden Planen
- Nachweis der Entscheidbarkeitseigenschaften:
  - Hierarchisches Planen ist *nicht* entscheidbar
  - Hybrides Planen *ist* entscheidbar
- Techniken zum effizienten Lösen hybrider Planungsprobleme
  - Hierarchisches Planen: Landmarkentechnik
  - Nicht-Hierarchisches Planen: Domänentransformation

Beispiel (Domäne)



$C := \{H, Q\}$

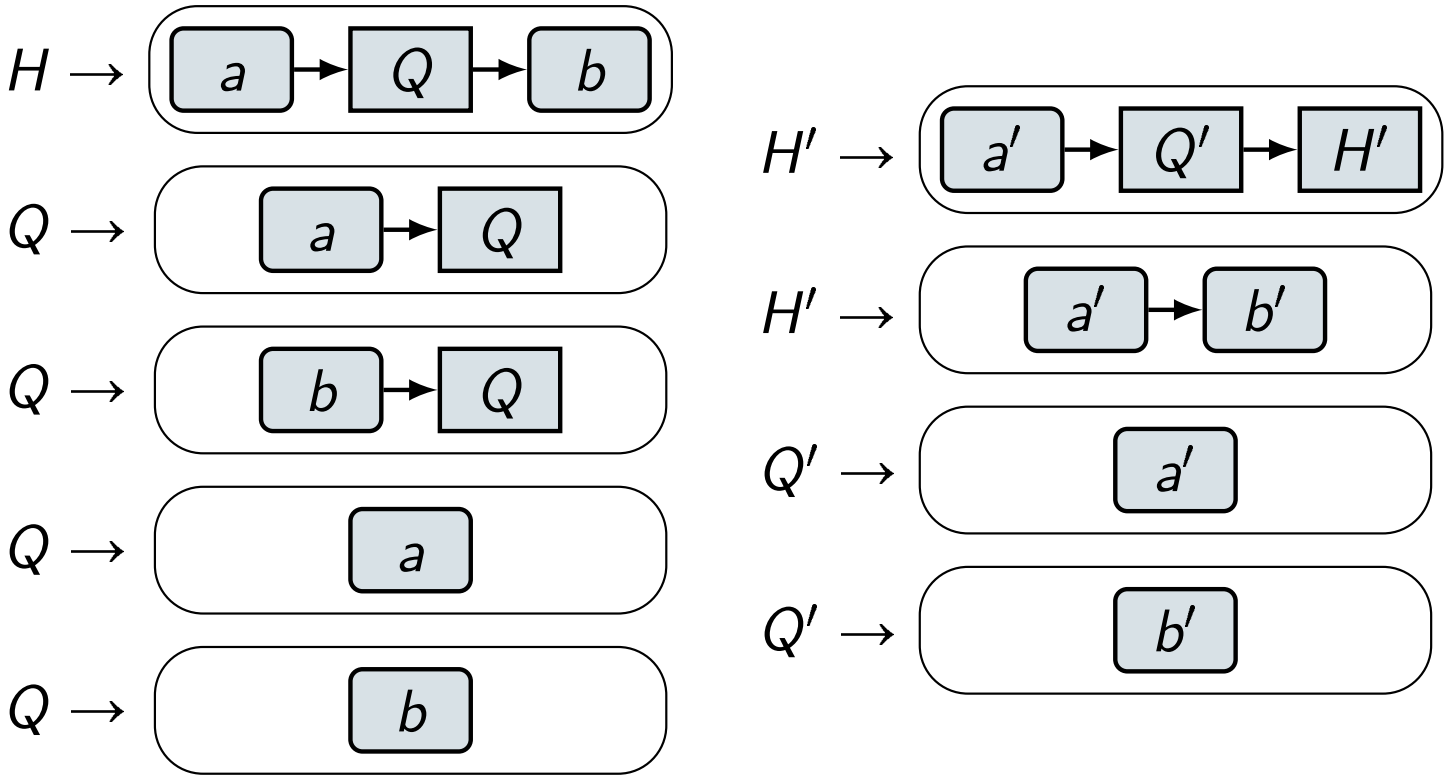
$A := \{a, b\}$ (Tupel nicht angegeben)

$M := \{m_1, \ldots, m_5\}$ wie links; Bsp.:
$m_1 := \langle H, \langle PS_1, \prec_1, \alpha_1 \rangle \rangle$ mit:

- $PS_1 := \{ps_1, ps_2, ps_3\}$,
- $\prec_1 := \{(ps_1, ps_2), (ps_2, ps_3)\}$
- $\alpha_1 := \{(ps_1, a), (ps_2, Q), (ps_3, b)\}$

Die Dekompositionsmethoden beschreiben die Grammatik G.
Das Startsymbol von G ist H; damit ist $L(G) = a(a|b)^+ b$.

Beispiel (Domäne)



Die Dekompositionsmethoden beschreiben die Grammatik 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 $L(G') = (a'(a'|b'))^*a'b'$.

Beispiel (Planungsproblem)

Gibt es ein Wort, das von beiden Sprachen erzeugt wird?

Prinzipielle Idee:

- Der "initiale Plan" besteht aus den Startsymbolen beider Grammatiken
- Wurde eine Aktion ausgeführt, die einen Buchstaben i' erzeugt, muss unmittelbar zuvor der Buchstabe i erzeugt worden sein
- Der letzte erzeugte Buchstabe muss ein Buchstabe i' der Grammatik G' sein

$\Rightarrow$ Damit werden immer identische Worte über a, b sowie über a', b' erzeugt!

Beispiel eines Planungsproblems (Problemstellung)

Domänenmodell $\mathcal{D} = \langle V, C, A, M \rangle$:

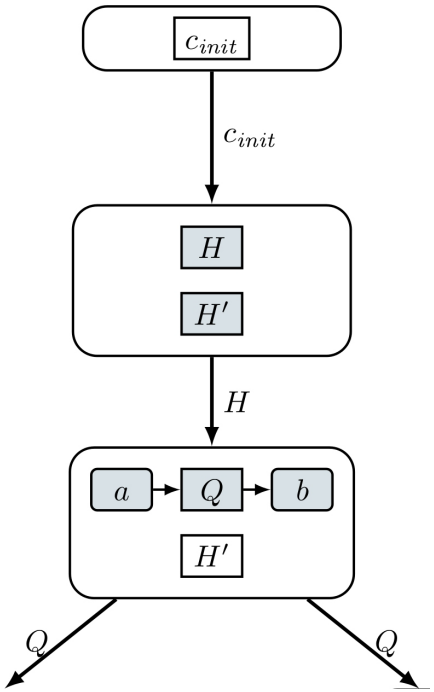
- $V := \{a, a', b, b', turn_G, turn_{G'}\}$
- $C := \{H, Q, H', Q', c_{init}\}$
- $A := \{a_1, a_2, a_3, a_4\}$ mit: $\overbrace{}^{pre}, \overbrace{}^{add}, \overbrace{}^{del}$
  - a_1 zur Erzeugung von a: $\langle \overbrace{\{turn_G\}}^{pre}, \overbrace{\{turn_{G'}, a\}}^{add}, \overbrace{\{turn_G\}}^{del} \rangle$
  - a_2 zur Erzeugung von a': $\langle \{turn_{G'}, a\}, \{turn_G\}, \{turn_{G'}, a\} \rangle$
- $M := \{m_1, \ldots, m_5, m'_1, \ldots, m'_4, m_{c_{init}}\}$ kodiert zusätzlich den initialen Plan: $m_{c_{init}} := \langle c_{init}, \langle PS_{init}, \prec_{init}, \alpha_{init} \rangle \rangle$
  - $PS_{init} := \{ps_1, ps_2\}$
  - $\prec_{init} := \emptyset$
  - $\alpha_{init} := \{(ps_1, H), (ps_2, H')\}$

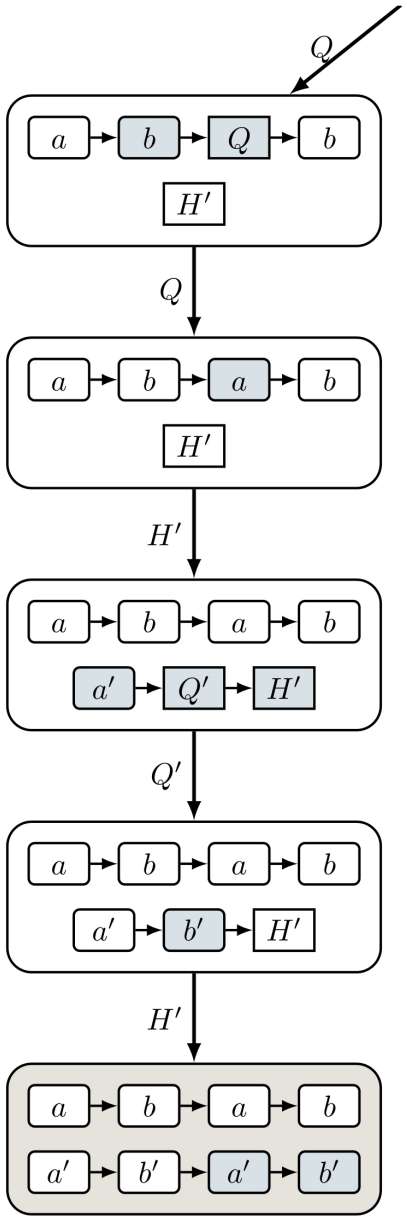
Probleminstanz:

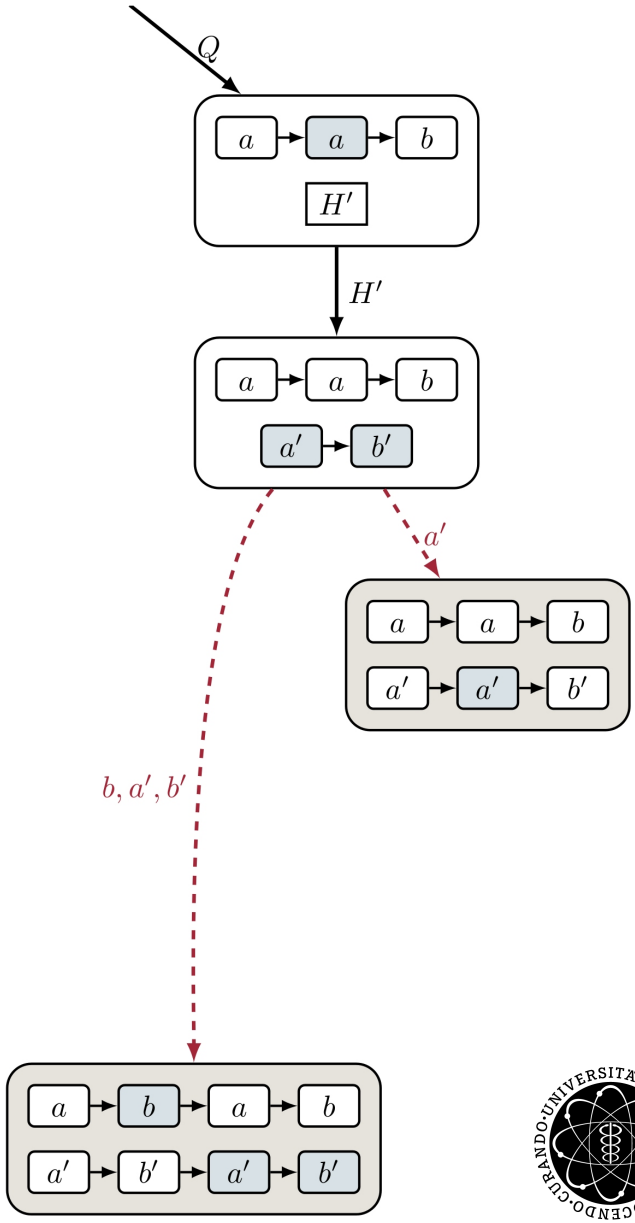
- $\mathcal{I} := \langle \underbrace{\{turn_G\}}_{s_{init}}, c_{init}, \underbrace{\{turn_G\}}_{goal} \rangle$

Beispiel (Planungsprozess)

$$L(G) = a(a|b)^+ b$$
$$L(G') = (a'(a'|b'))^* a' b'$$





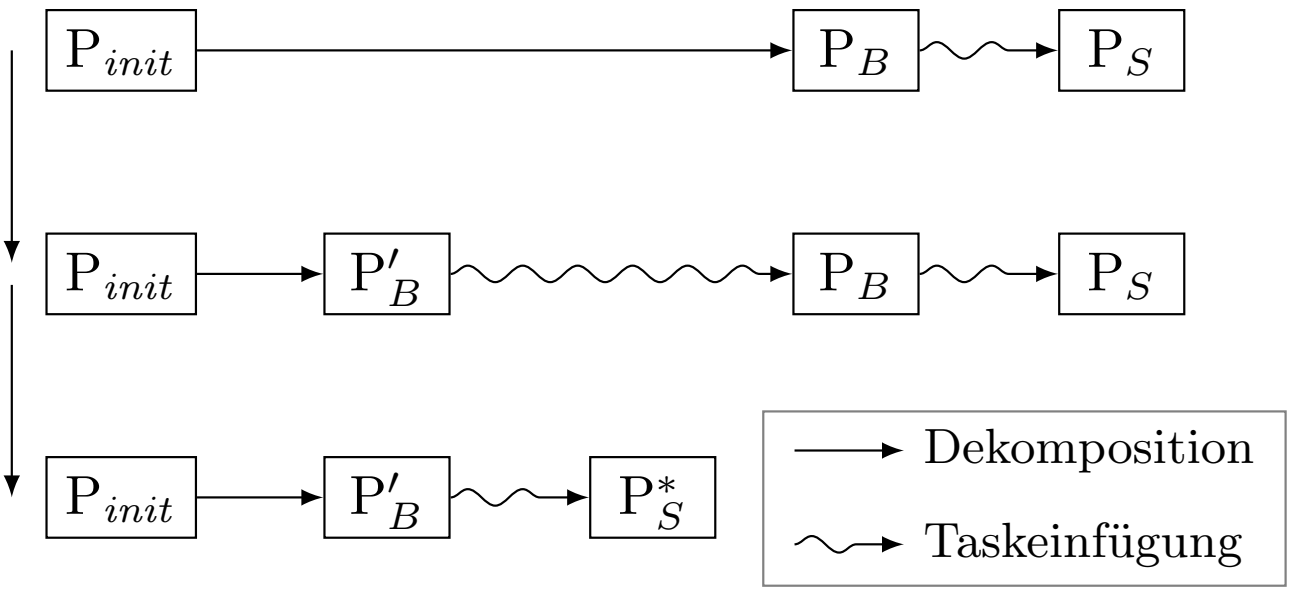


Entscheidbarkeit im Hybriden Planen (Beweisidee)

Die maximale Länge der kürzesten Lösung jedes hybriden Planungsproblems ist beschränkt. Wieso?

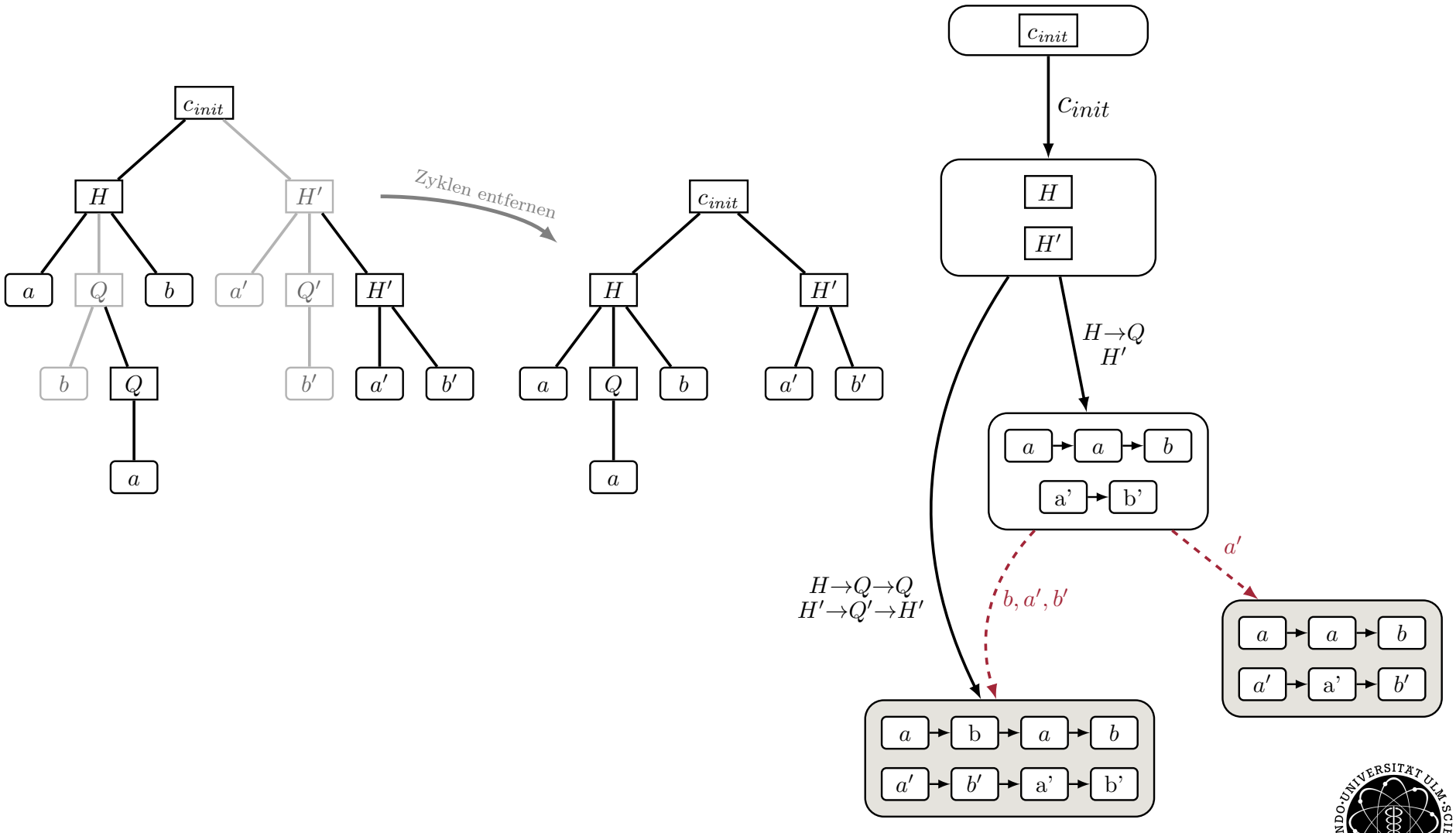
- Falls $P_{init} \xrightarrow{D} P_B$, so gilt $P_{init} \xrightarrow{D} P'_B \xrightarrow{I} P_B$, $|P'_B| \leqslant n$
- Falls $P'_B \xrightarrow{I} P_S$ so gilt $P'_B \xrightarrow{I} P^*_S$, $|P^*_S| \leqslant m$

$\Rightarrow$ Falls $P_{init} \xrightarrow{D} P_B \xrightarrow{I} P_S$, so $P_{init} \xrightarrow{D,I} P^*_S$, $|P^*_S| \leqslant f(n, m)$



Entscheidbarkeit im Hybriden Planen (Beispiel)

Rekonstruktion der Dekomposition:



Evaluation der Landmarkenstrategien (Domänen)

Experimente wurden auf zwei Domänen durchgeführt

- UM Translog
  - Logistikdomäne – ursprünglich für UMCP entworfen
  - 12 qualitativ verschiedene Probleminstanzen
  - Tiefer Dekompositionsgraph (Tiefe 10)
- Satellite
  - Observationsaufgaben
  - Problemstruktur: x Observationen – y Satelliten – z Modi.
  - Flacher Dekompositionsgraph (Tiefe 3)

Evaluation der Landmarkenstrategien (UM-Translog)

rot → bestes Ergebnis **fett** → zweitbestes Ergebnis
#n → Anzahl Suchknoten

FlawSel	1 #n	2 #n	3 #n	4 #n	5 #n	6 #n	7 #n	8 #n	9 #n	10 #n	11 #n	12 #n
UMCP	58	156	177	55	57	308	63	75	**220**	161	92	90
EMS	147	405	211	127	114	–	1571	113	–	2558	879	500
SHOP	89	164	146	106	83	926	98	95	–	247	121	173
lm_1	62	**135**	**141**	**53**	55	339	109	73	420	211	84	91
lm_1^*	124	146	**137**	57	**51**	305	110	81	367	**142**	87	84
lm_2	**52**	**133**	145	62	**53**	**291**	63	**71**	**158**	183	**75**	**72**
lm_2^*	**51**	**135**	154	**52**	65	**266**	**61**	**61**	304	**158**	**78**	89
lcf	55	155	162	78	127	327	**62**	86	–	227	79	90
da-HotSpot	144	644	239	114	148	723	99	120	–	184	641	588
HotZone	55	197	191	55	55	–	159	122	–	701	81	**76**

Evaluation der Landmarkenstrategien (Satellite)

rot → bestes Ergebnis **fett** → zweitbestes Ergebnis
#n → Anzahl Suchknoten

FlawSel	1–1–1 #n	1–2–1 #n	2–1–1 #n	2–1–2 #n	2–2–1 #n	2–2–2 #n
UMCP	83	**36**	883	1558	**278**	1062
EMS	65	47	1586	–	1219	–
SHOP	62	105	**138**	–	1406	–
lm_1	78	128	4890	**200**	–	483
lm_1^*	73	91	–	1905	–	**146**
lm_2	73	194	**560**	**352**	693	295
lm_2^*	78	**34**	847	1803	739	619
lcf	86	71	1120	3022	**407**	–
da-HotSpot	**56**	68	782	832	2186	**142**
HotZone	**61**	57	1281	–	1094	871