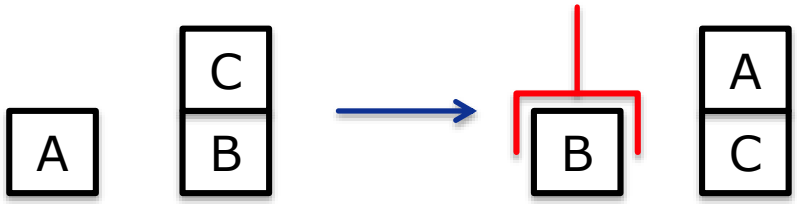
Planungssysteme



Prof. Karsten Berns
Robotics Research Lab
Department of Computer Science
University of Kaiserslautern, Germany

Inhalt

- Grundlagen für Roboterplanung
- Planungsarten
- Planen als Suche
- Cranfield-Montage-Benchmark
- Zusammenfassung der Grundlagen

Planungsarten in Roboteranwendungen

- Aktionsplanung
  - Welche Aktionen werden aus Aufgabe abgeleitet?
- Reihenfolgeplanung
  - Zeitliche Beziehungen zwischen Aktionen
- Ressourcenplanung
  - Welche Roboter/Geräte führen Aktionen aus?
- Zeitplanung
  - Wann werden Aktionen gestartet?
- Durchführungsplanung
  - Welche Parameter für Aktionsausführung (Bahn- und Greifplanung)?
- Scheduling: Ressourcen- und Zeitplanung

Planung in der Robotik: Planungssystem

- Bekannter Anfangszustand
- Erwünschter Zielzustand
- Generiert Aktionsplan (Folge von Aktionen), welcher ein System schrittweise von Anfangs- in Zielzustand überführt
- Planüberwachung mit Sensoren
- Umplanung bei Abweichung

Planung in der Robotik: Plan

- Von Planer erzeugte Aktionssequenzen oder Graphen
- Beschreibt im Zustandsraum eine Menge von Zustandsänderungen, die zum Zielzustand führen
  - Zustandsänderungen können sequentiell, nebenläufig oder koordiniert nebenläufig sein
- Kann in Teilpläne zerlegt werden
  - Plansegmente: vollständige Teilpläne
  - Planskelette: unvollständige Teilpläne
- Aktions-/Planprimitiv: Nicht zerlegbares Plansegment bzw. Zustandsübergang
- Linearer Plan: Vergleichbar mit linearer Zerlegung der Stellgröße zum Minimieren der Regelabweichung eines Regelsystems

Planungsverfahren

- Planen: Finden einer Schrittfolge, um von einer Ausgangssituation zu einer Zielsituation zu gelangen
- Verkettung von Zustandsänderungen kann berechnet, abgeleitet, gesucht oder ausgewählt werden
- Planungsverfahren verarbeiten ausschließlich Symbolmengen
- Sinnlose Pläne möglich, wenn Wissen fehlerhafte Zustände oder nicht ausführbare Zustandsänderungen enthält

Planungsverfahren

- Beschreibung eines Planungsproblems
  - Ausgangssituation
  - Mögliche Aktionen (Operationen), die eine Situation verändern können
  - Zielsituation (oder Menge von Zielsituationen)
- Einfachster Fall: Plan ist einfache Folge (oder ein azyklischer Graph) von Aktionen
- Allgemeiner Fall: Alle (aus einer Programmiersprache bekannte) Kontrollstrukturen möglich

Rand- und Vorbedingungen

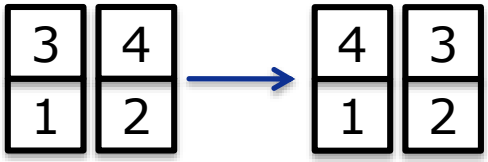
- Nicht alle Aktionen in bestimmte Situationen möglich
- Beispiel Greifen: Roboter kann Objekt greifen, wenn …
  - Objekt zugänglich
  - Masse des Objekts vom Roboter bewegbar
  - Greifer geöffnet
  - Greifer am Greifpunkt des Objekts
  - Greifer zum Greifen des Objekts geeignet

Nebenbedingungen in der Montageplanung

- Min. Bearbeitungszeit oder in fester Zeitspanne
- Möglichst wenig Werkzeugwechsel
- Mechanische Stabilität der teilmontierten Komponenten
- Verbindungsform (z.B. Abbindzeiten beim Kleben einhalten)
- Materialeigenschaften (z.B. verformbar)
- Beobachtbarkeit (z.B. Kamera)
- Genauigkeit bei der gegenseitigen Positionierung von Teilen

Arten des Planens

- Linear: strenge zeitliche Sequenz der Ausführung
- Nichtlinear: komplexe zeitliche Zusammenhänge von Aktionen (Ausführung nacheinander, überlappend, parallel)

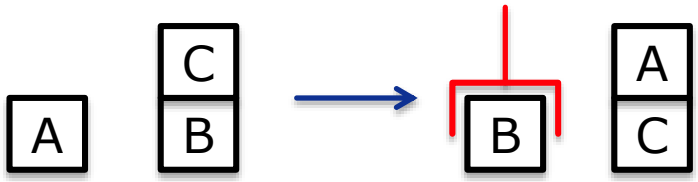


Keine Ablage auf dem Tisch möglich

- Monoton: jedes Teil wird sofort in Endposition gebracht (keine Zwischenablagen)
- Nicht monoton: mit Zwischenpositionen
- Zentral: vollständiges Wissen über alle Teilprozesse
- Verteilt: Multiagentensysteme
- Einstufig: Plan generiert direkt Aktionen
- Mehrstufig: Plan generiert Teilziele

Verträglichkeit der Teilziele

- Anfangszustand: A und B auf Tisch, C auf B
- Zielzustand: A auf C und B in Robotergreifer
- Teilziele:
  - A auf C, B in Greifer
  - Zuerst A auf C → Erstes Teilziel verhindert zweites
  - Zuerst B in Greifer → Zweites Teilziel zerstört erstes



Grundkonzepte des Planens

- Z: Menge der möglichen Zustände
- Startzustand $M_a \in Z$, Zielzustand $M_e \in Z$
- Menge O von Aktionen α mit Vorbedingung V_α
- Zustandstransformation $g_\alpha: Z \to Z$ beschreibt Zustandsübergang falls V_α erfüllt
- Gesucht: Folge M_i von Zuständen mit $M_1 = M_a$, $M_n = M_e$
  $\forall i = 1..n, \exists \alpha_i \in O$ mit $M_{i-1} \succ V_{\alpha_i}$ und $g_{\alpha_i}(M_{i-1}) = M_i$
- Lösung
  - Plangenerierung durch Folge logischer Verknüpfungen
  - Aufbau eines Baums mit Knoten als Zustände und Kanten als Aktionen
  - Verwendung von Suchverfahren zur Plangenerierung (Tiefensuche, Breitensuche, heuristische Suchverfahren)

Planen als logische Abbildung

- Lösung von Planungsproblemen durch Ableiten über ein logisches Kalkül, wenn es wie folgt definiert wird: „Kann der Zielzustand ausgehend von dem aktuellen Zustand erreicht werden?“
- Verwendetes Kalkül: Prädikatenlogik 1. Ordnung oder Situationskalkül ν
- Ableiten erreichbarer Zustände durch Planen durch Inferenz
- Planen durch monoton-logisches Ableiten für komplexere Problemstellungen oder reaktive Planungssysteme nur beschränkt einsetzbar
- Viele Phänomene der realen Welt und damit auch der realen Umgebungen in denen Agenten agieren sind nicht streng monoton logisch ableitbar

Planen als logische Abbildung

- Für die meisten praktischen Anwendungen müssen nicht-monotone Logiken durch Ausnahmeregeln erweitert werden
  - Es erübrigt sich damit einen Planer für ein autonomes Robotersystem allein als „logischen Beweiser" zu implementieren
- Um Ableiten zu können, ob ein Zustand von einem anderen aus erreicht werden kann, müssen Zustände formal beschreibbar sein
- Angenommen, es gibt eine Menge von Sensoren bzw. Beobachtern, die Eigenschaften von Individuen oder Relationen zwischen Individuen im Prozess nachweisen können. Dann definiert die Menge der im Prozess nachweisbaren Eigenschaften und Relationen den aktuellen Zustand des Prozesses.

Planen als logische Abbildung

- $\text{holds}(x_j, y)$: In Prozesszustand y gilt Eigenschaft x_j
- Mit Junktoren und Quantoren lassen sich Terme bilden
- Terme bildbar, die allgemeingültige Aussagen treffen
- Es lassen sich neue Eigenschaften von Individuen und Relationen zwischen Individuen ableiten, auch dann, wenn man sie nicht direkt mit Sensoren messen kann
- Beispiel: Wenn x_j und x_k gelten, dann gilt auch x_l

Planen als logische Abbildung

- Zur Vorgangsbeschreibung (Interaktion, Aktion, Verhalten) wird im Situationskalkül ein Ergebnisoperator (result) eingeführt
- Wenn in Prozesszustand y eine Operation o oder Aktion u zur Zustandsänderung nachgewiesen werden kann, dann kommt es zur neuen Situation $\mathrm{result}(u, y)$
- Damit lässt sich das Axiom aufstellen, dass für jeden Zustand y und jede Aktion u der Zustand $\mathrm{result}(u, y)$ als erreichbar gilt. Dieser Zustand ist auch erreichbar von jedem Zustand y_s, von dem aus y erreichbar ist

$$\forall y\colon \mathrm{reachable}(y, y)$$
$$\forall y, u\colon \mathrm{reachable}(\mathrm{result}(u, y), y)$$
$$\forall y, y_s, u\colon \mathrm{reachable}(\mathrm{result}(u, y), y_s) \Leftrightarrow \mathrm{reachable}(y, y_s)$$

Planen als logische Abbildung

- Mit diesem Axiom lässt sich die Erreichbarkeit eines Zustands von einem anderen Zustand aus ableiten
- Abhängig vom gewählten Inferenzmechanismus, kann die Realisierung des Planens durch Ableitung einem Planen durch Suche entsprechen (Prolog-Tiefensuche)
- Beschreibung des Beweisproblems:
  „Weise nach, dass es min. einen beliebigen Zustand y_z gibt, der ausgehend von einem beliebigen Zustand y_s aus erreicht werden kann, wobei gilt, dass y_s durch die Eigenschaft x_s definiert ist und y_z durch die Eigenschaft x_z"

$$\forall y_s : \text{holds}(x_s, y_s) \Rightarrow \exists y_z : \text{holds}(x_z, y_z) \wedge \text{reachable}(y_z, y_s)$$

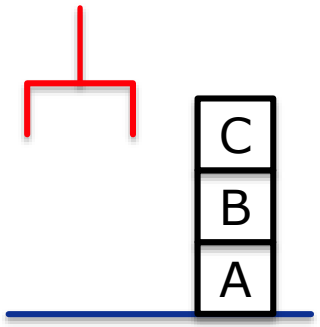
Planen als logische Abb.: Bsp. Situationskalkül

- Beweiser: Leitet ab, ob ein Plan existiert, der Startzustand (a) in Zielzustand (b) überführt
- Der Zielzustand ist durch Block A definiert

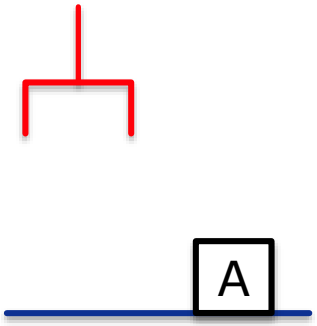
$$\text{holds}\big(\text{clear}(A, y_z)\big)$$

- Der Anfangszustand y ist definiert als

$$holds(on(C,B), y_s) \wedge holds(on(B,A), y_s) \wedge holds(clear(C), y_s)$$



(a) Startzustand



(b) Zielzustand

Planen als logische Abb.: Bsp. Situationskalkül

- Regel für die Aufsetzbewegung

$$\forall y: \text{holds}(\text{clear}(top), y) \wedge \text{holds}(\text{on}(top, bot1), y) \wedge \text{holds}(\text{clear}(bot2), y)$$
$$\Rightarrow \text{holds}\big(\text{on}(top, bot2), \text{result}(\text{puton}(top, bot2), y)\big)$$
$$\wedge \text{holds}\big(\text{clear}(bot1), \text{result}(\text{puton}(top, bot2), y)\big)$$

- Zu beweisende Bedingung

$$\forall y_s: \text{holds}(\text{on}(C, B), y_s) \wedge \text{holds}(\text{on}(B, A), y_s) \wedge \text{holds}(\text{clear}(C), y_s)$$
$$\Rightarrow \exists y_z: \text{holds}(\text{clear}(A), y_z) \wedge \text{reachable}(y_z, y_s)$$

- Ergebnis der Anwendung der Ableitungsregeln

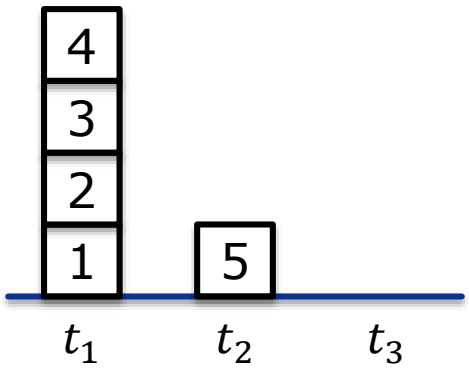
$$\forall y_s: \text{holds}(\text{on}(C, B), y_s) \wedge \text{holds}(\text{on}(B, A), y_s) \wedge \text{holds}(\text{clear}(C), y_s)$$
$$\Rightarrow \exists y_z: \text{holds}(\text{clear}(A), y_z)$$
$$\wedge \text{reachable}\left(y_z, \text{result}\big(\text{puton}(B, table), \text{result}(\text{puton}(C, table), y_s)\big)\right)$$

Beispiel: Planen als Suche

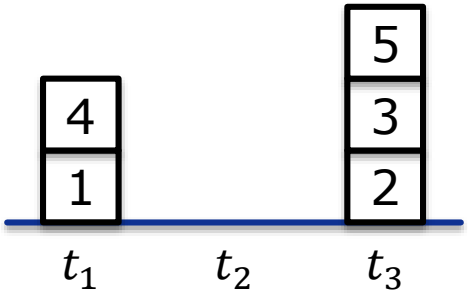
Möglicher Plan

(1) Klotz 4 auf Klotz 5
(2) Klotz 3 auf Klotz 4
(3) Klotz 2 auf t_3
(4) Klotz 3 auf Klotz 2
(5) Klotz 4 auf Klotz 1
(6) Klotz 5 auf Klotz 3

Startzustand



Endzustand



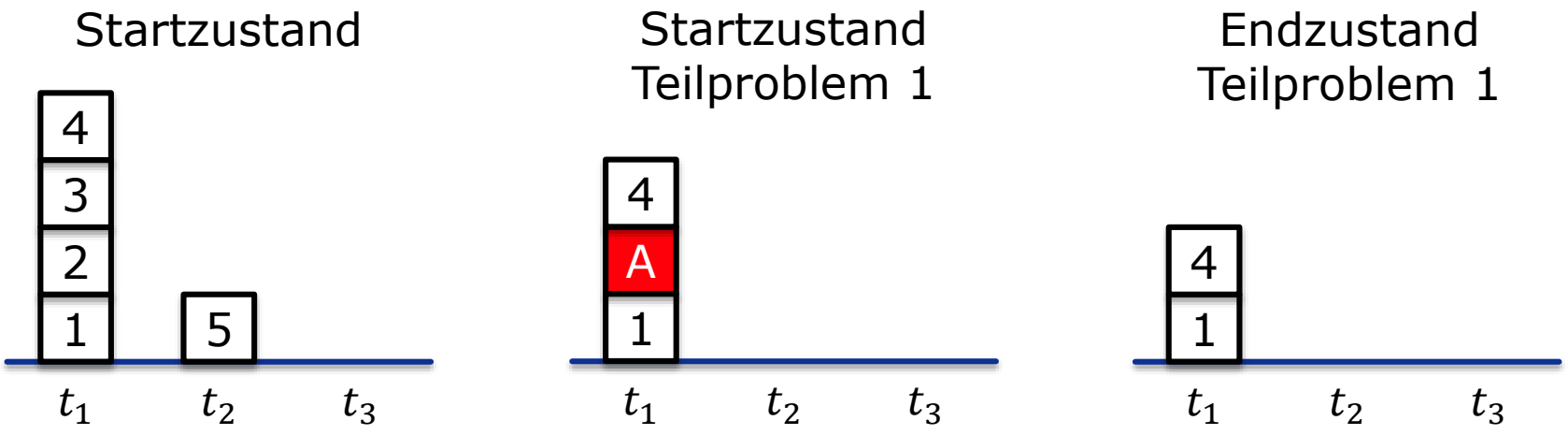
Beispiel: Planen mit Teilzielen

- Teilziel 1: $\mathrm{on}(1, t_x)$ und $\mathrm{on}(4,1)$
- Plan für Teilziel 1

(1) Klotz 4 auf Tisch

(2) Klotz A auf Tisch

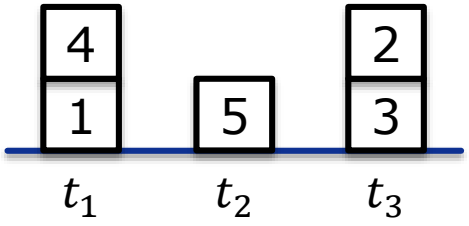
(3) Klotz 4 auf Klotz 1



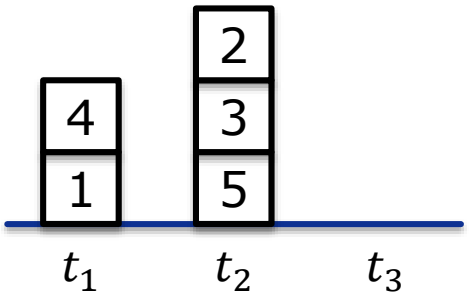
Beispiel: Planen mit Teilzielen

- Teilziel 2: $\mathrm{on}(2, t_y)$, $\mathrm{on}(3,2)$ und $\mathrm{on}(5,3)$
- Zwei mögliche Startzustände

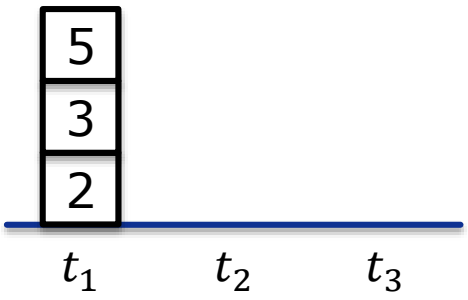
Startzustand 1
Teilproblem 2

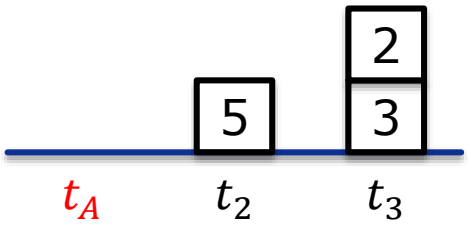


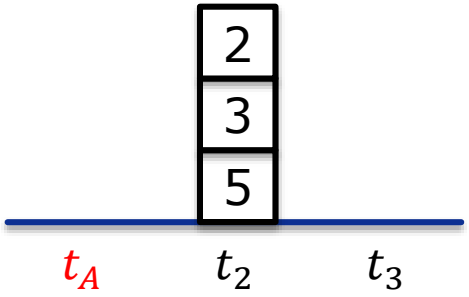
Startzustand
Teilproblem 1



Zielzustand
Teilproblem 2





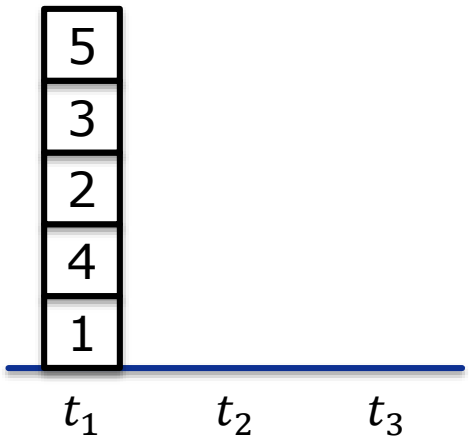


Abstrakte
Startzustände

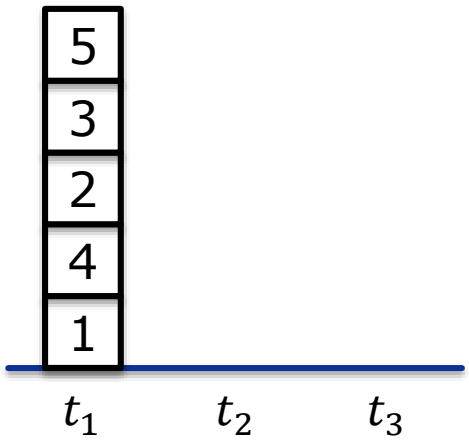
Beispiel: Planen mit Teilzielen

- Plan für Teilziel 2
  (1) Klotz 2 auf Tisch
  (2) Klotz 3 auf Klotz 2
  (3) Klotz 5 auf Klotz 3
- Drei mögliche Lösungen

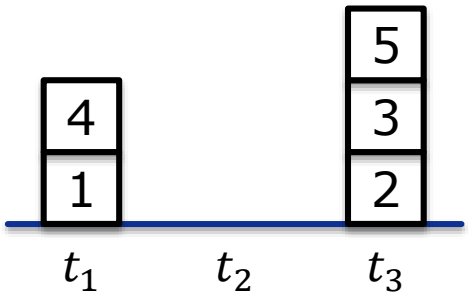
Lösung 1
Teilproblem 2



Lösung 2
Teilproblem 2



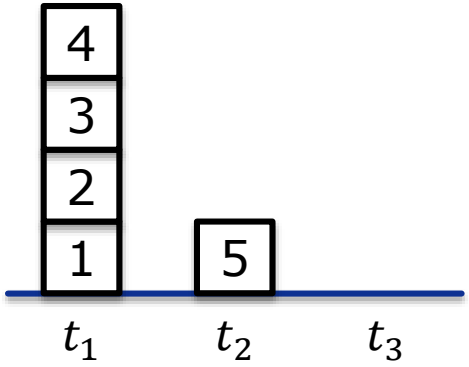
Lösung 3
Teilproblem 2



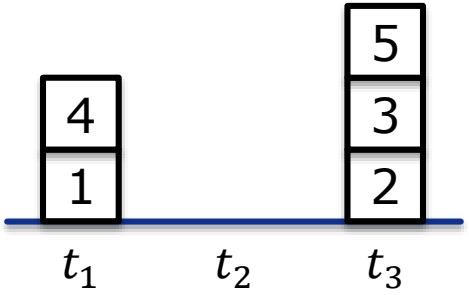
Beispiel: Optimale Gesamtlösung

(1) Klotz 4 auf t_3
(2) Klotz 3 auf Klotz 5
(3) Klotz 2 auf Klotz 3
(4) Klotz 4 auf Klotz 1
(5) Klotz 2 auf t_3
(6) Klotz 3 auf Klotz 2
(7) Klotz 5 auf Klotz 3

Startzustand

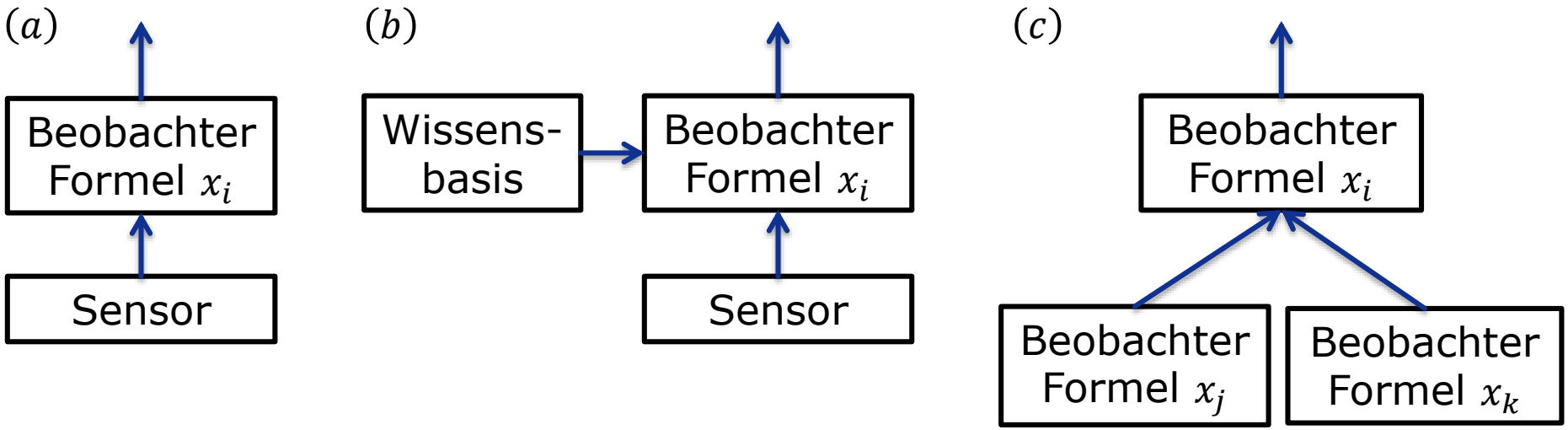


Endzustand



Planen als Suche: STRIPS

- Logische Formel x_i trifft Aussagen über Eigenschaften/Relationen von Prozess y
- Prozesszustand $x(y)$ als Konjunktion von Formeln x_i
- $x(y) = \bigcup x_i$ mit x_i gilt in Zustand y
- Beruht auf Informationen aus Sensordaten (a), abgelegtem Vorwissen (b) oder Kalkülberechnungen (c)



Planen als Suche: STRIPS

- Zustände werden durch Operatoren o oder über aus Operatoren bestehende Pläne p verändert
- Ein Operator ist ein Quadrupel $o = (u, cond, add, sub)$
  - Bezeichnung u: Semantische Bedeutung der Änderung
  - Vorbedingung $cond$: Logische Formel, prüft ob für Zustand der Operator anwendbar ist
  - Eigenschaftsadditionsmenge add: Zusätzliche Formeln des neuen Zustands gegenüber ursprünglichem Zustand
  - Eigenschaftssubtraktionsmenge sub: Fehlende Formeln im neuen Zustand gegenüber ursprünglichem Zustand

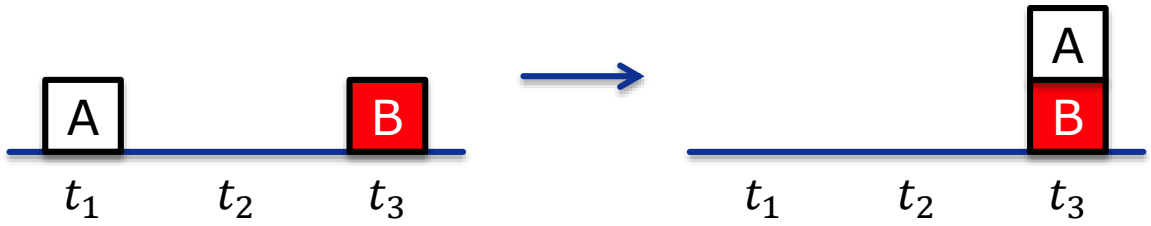
STRIPS: Repräsentation der Blockwelt

- Aktueller Zustand

$$x(y) = \{\text{clear}(A), \text{clear}(B), \text{red}(B)\}$$

- Operator-Quadrupel

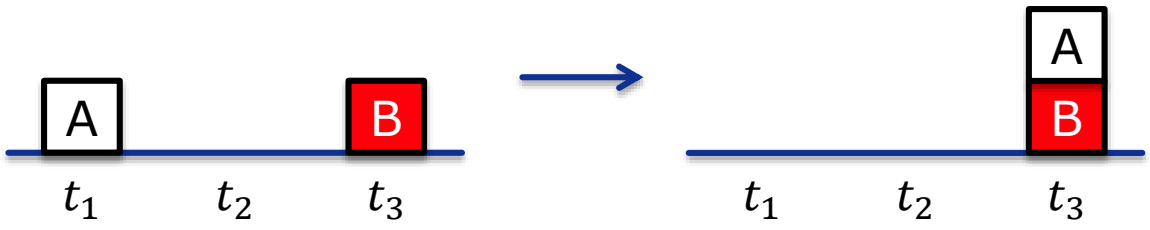
$$(\text{puton}(A, B), \{\text{clear}(A), \text{clear}(B)\}, \{\text{clear}(A), \text{on}(A, B)\}\{\text{clear}(B)\})$$



STRIPS: Repräsentation der Blockwelt

- Anwendung des Operators
  - $\text{puton}(A, B)$ darf angewendet werden, da $\{\text{clear}(A), \text{clear}(B)\}$ zutrifft
  - Es gilt weiterhin $\text{clear}(A)$
  - Zusätzlich gilt $\text{on}(A, B)$
  - $\text{clear}(B)$ gilt nicht mehr
  - Über $\text{red}(B)$ wird keine Aussage getroffen
- Ergebnis der Operatoranwendung

$$x(y) = \{\text{clear}(A), \text{red}(B), \text{on}(A, B)\}$$



STRIPS: Eigenschaften

- In der Additionsliste müssen alle Aussageformeln stehen, die im neuen Zustand gültig sind
  - Beispiel: Würde $\mathrm{clear}(A)$ fehlen, könnte $\mathrm{puton}(A, B)$ in einen Zustand führen, in dem $\mathrm{clear}(A)$ nicht gilt
- In der Subtraktionsliste müssen alle Aussageformeln stehen, die im neuen Zustand nicht mehr gültig sind

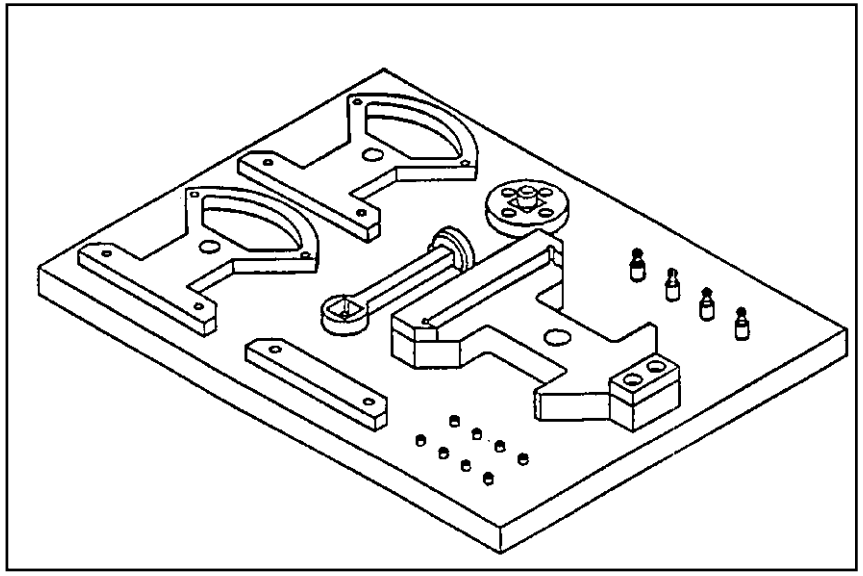
Planungssysteme

- ASTRIPS (hierarchische)
- HACKER (einstufig)
- NOAH (hierarchische)
- ATLAS
- SHARP
- TWAIN (mehrstufig)
- …

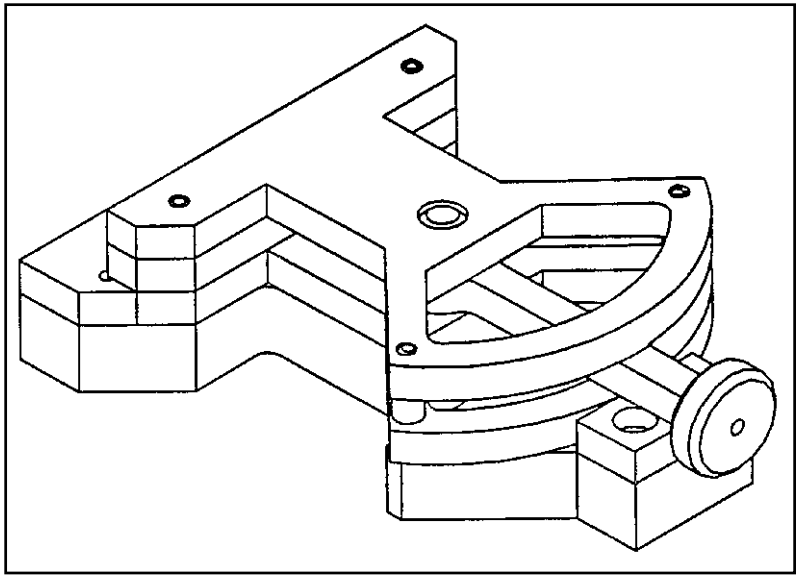
Beispiel: Planungsphasen für einfache Montage

- Weltmodellierung
- Aufgabenspezifikation
- Aufgabenanalyse
- Erstellung von Vorranggraphen
- Planung von Ausführungsdetails
- Ausführung

Beispiel: Cranfield-Montage-Benchmark



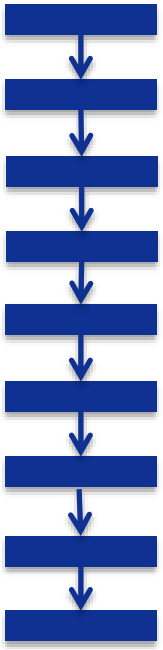
Anfangskonfiguration



Endkonfiguration

Beispiel: Planung u. Überwachung der Montage

1. Produkt- und Aufgabenbeschreibung
2. Vorranggraph
3. Erweiterter Vorranggraph
4. Strategische Planung
5. Symbolischer Aktionsplan
6. Geometrische Planung
7. Geometrisch detaillierte Planung
8. Generierung
9. Roboteranweisung

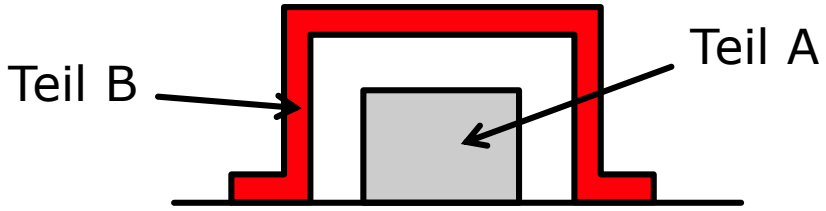


Aufgabenspezifikation

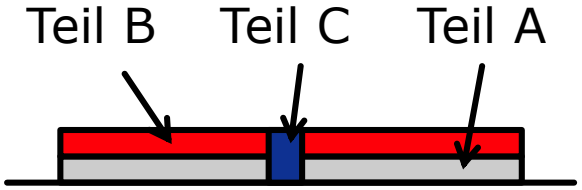
- Modellierung der Anfangs- und Endkonfiguration der Montage
- Konfiguration
  - Geometrie (3D)
  - Physikalischen Eigenschaften (Gravitationszentrum, Material)
  - Kartesischen Koordinaten des Werkstücks
- Graphischer Editor zur Spezifizierung …
  - … des Ortes des Werkstücks
  - … der Greifkonfigurationen

Kriterien der Aufgabenanalyse

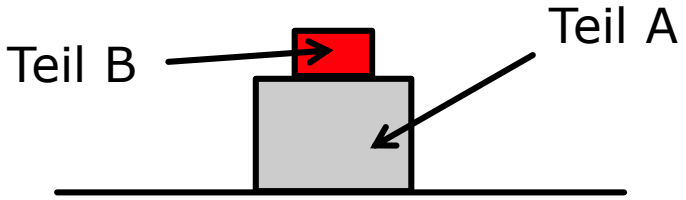
- Keine Durchdringung



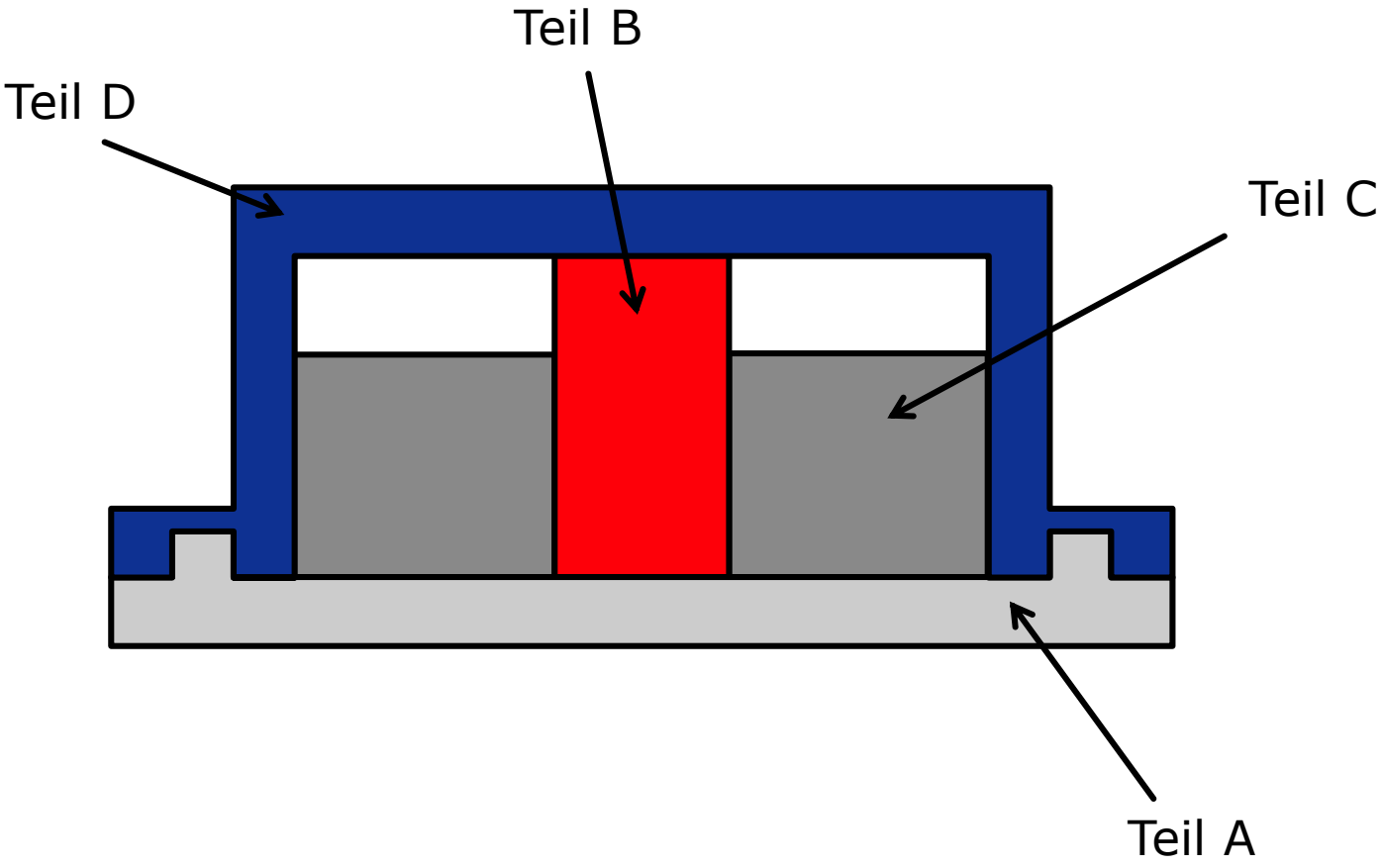
- Keine Seiteneffekte



- Stabilität



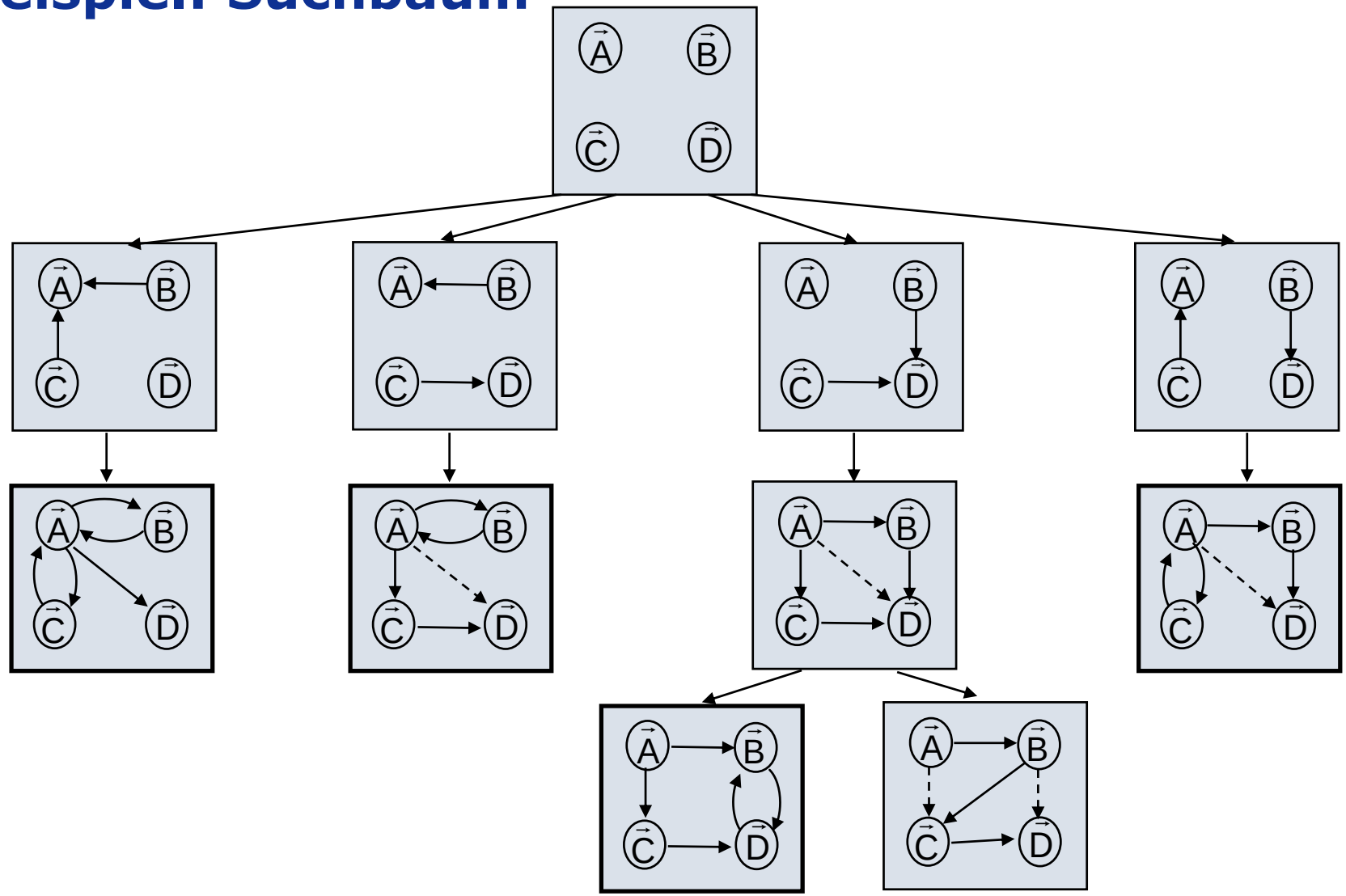
Beispiel: Anordnung



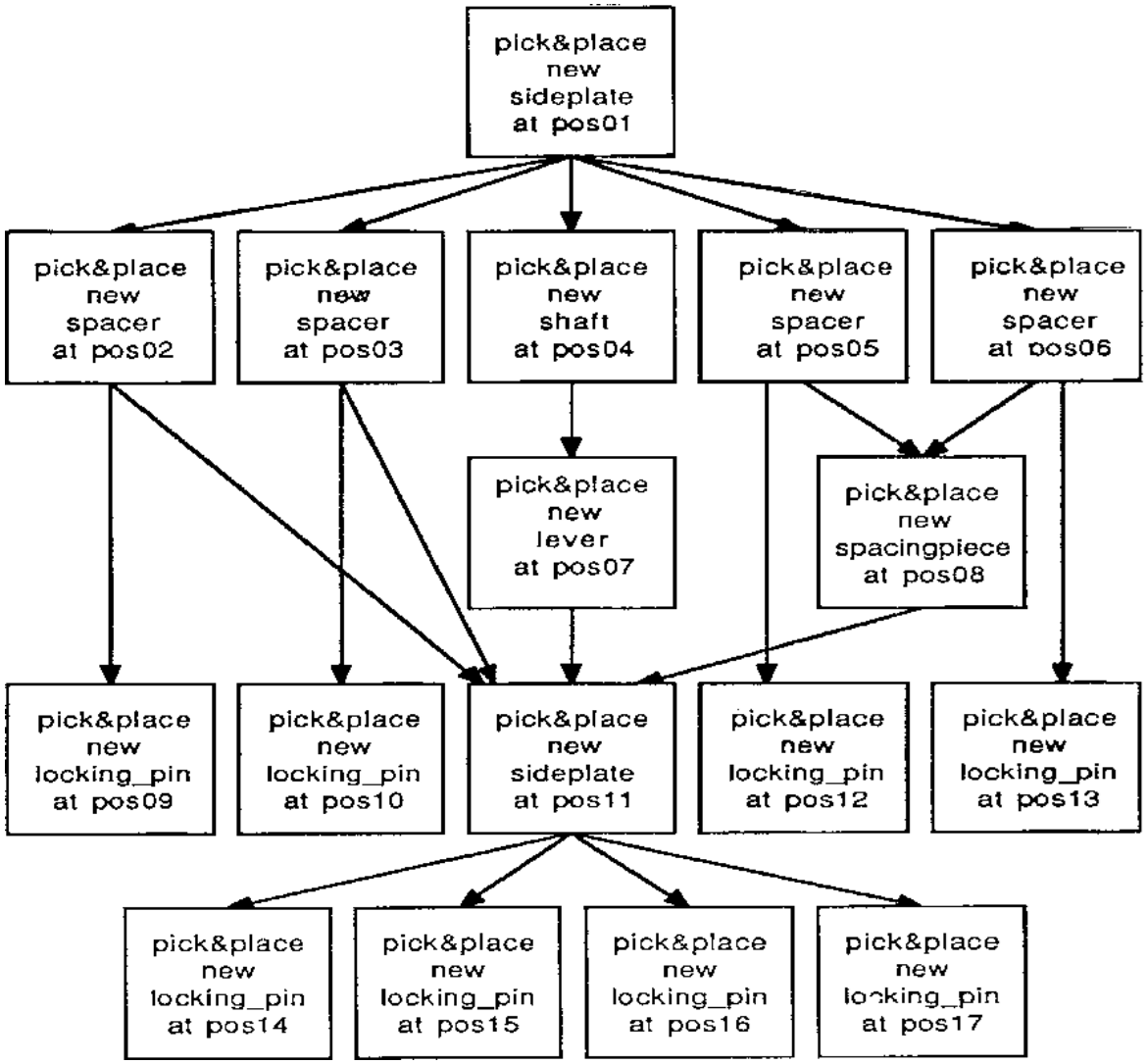
Beispiel: Vorrangbedingung

Beschränkende Quelle	Resultierende Vorrangbedingung
Durchdringungsregel	$\text{OR}\left(\text{AND}\left(\pi(\vec{B},\vec{A}),\pi(\vec{C},\vec{A})\right)\right),$ $\text{AND}\left(\pi(\vec{B},\vec{A}),\pi(\vec{C},\vec{C})\right),$ $\text{AND}\left(\pi(\vec{B},\vec{D}),\pi(\vec{C},\vec{D})\right),$ $\text{AND}\left(\pi(\vec{B},\vec{D}),\pi(\vec{C},\vec{A})\right)$
Stabilitätsregel	$\text{AND}\left(\pi(\vec{A},\vec{B}),\pi(\vec{A},\vec{C}),\pi(\vec{A},\vec{D})\right)$
Seiteneffektregel	$\text{OR}\left(\pi(\vec{B},\vec{C}),\pi(\vec{D},\vec{B})\right)$

Beispiel: Suchbaum



Vorranggraph des Cranfield-Benchmarks



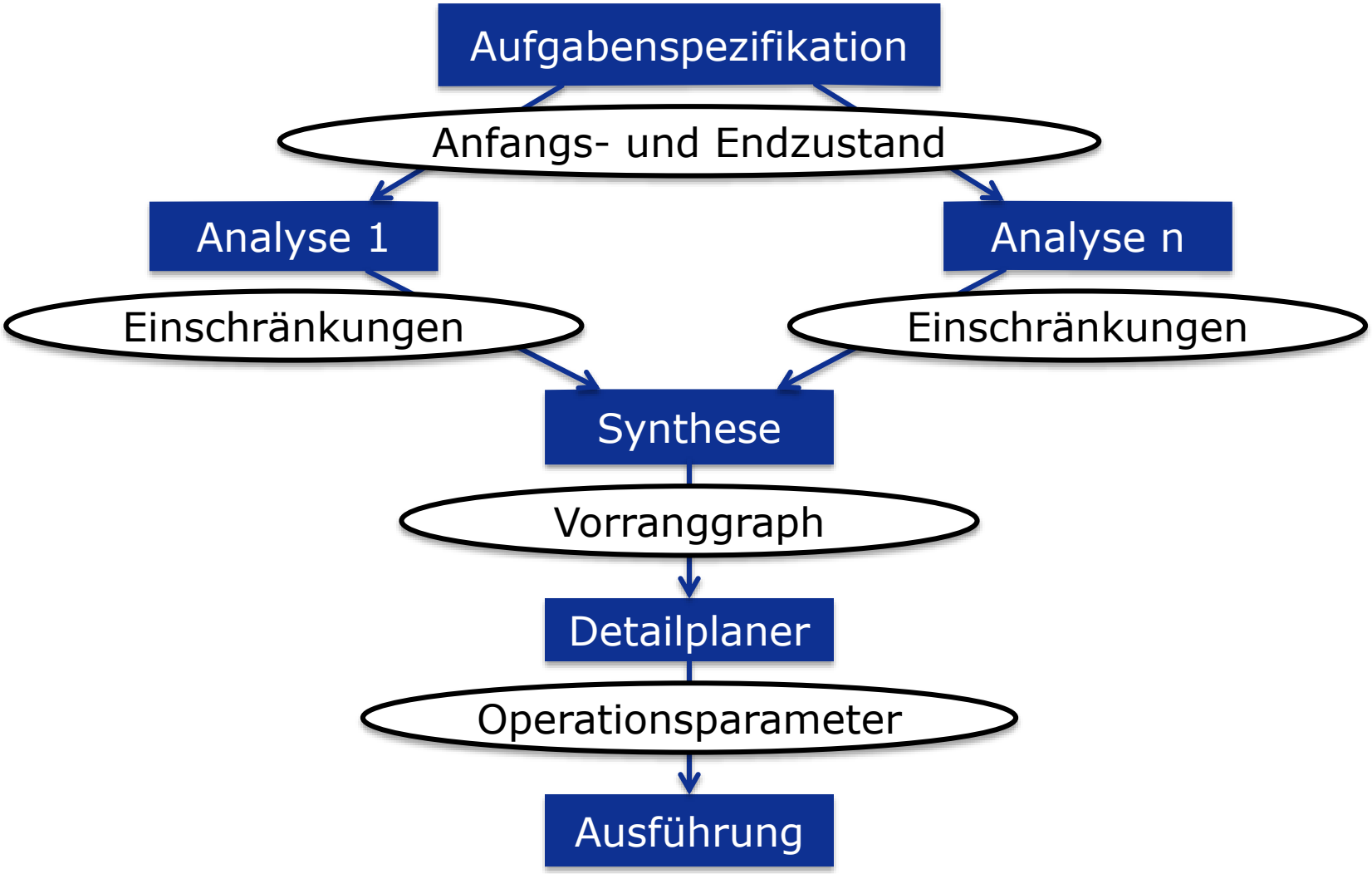
Detailplanung

- Roboterauswahl
- Layoutplanung
  - Finden einer Roboterposition
  - Kriterien: Erreichbarkeit, Genauigkeit und Effizienz
- Bewegungsplanung
  - Kollisionsfreie Bewegungen zum Greifen und Loslassen eines Werkstücks
  - Kollisionsfreie Feinbewegungen um eine gezielte Berührung zweier Werkstücke hervorzurufen
  - Kollisionsfreie grobe Manipulatorbewegungen zum Transport eines Werkstücks

Synthese der Vorranggraphen

- Verschmelzung des Regelsatzes, der aus den Analysekriterien abgeleitet wird
- Auswahl des „besten“ Vorranggraphens
- Eliminierung redundanter Kanten
- Detektion von Zyklen

Gesamtsystem



Nächste Vorlesung ...

Roboterarchitekturen

- Fähigkeiten eines Robotersystems
- Hierarchisch funktionsorientierte Architekturen
- Verteilte funktionsorientierte Architekturen
- Hierarchisch verhaltensbasierte Architekturen
- Verteilte verhaltensbasierte Architekturen