

Mathematische Modellierung und Behandlung von Stapelproblemen

Dissertation

zur Erlangung des akademischen Grades

Doctor rerum naturalium

an der Fakultät Bauingenieurwesen

der

Bauhaus-Universität Weimar

vorgelegt von

Dipl. Math. Lutz Kämmerer

aus Bad Klosterlausnitz

am 10. September 1997 in Weimar

Gutachter: Prof. Dr. L. Hempel
Prof. Dr. I. Althöfer
Prof. Dr. K. Neumann

Tag der Disputation: 30. April 1998

Inhaltsverzeichnis

Inhaltsverzeichnis	iii
1 Einführung	1
1.1 Begriffe der Graphentheorie	1
1.2 Die Türme von Hanoi	5
1.2.1 Historisches	6
1.2.2 Ein minimales Regelsystem	8
2 Einfache Stapel	11
2.1 Formale Beschreibung von Stapelproblemen	11
2.2 Streng zweistufig lösbare Stapelprobleme	14
2.3 Zweistufig lösbare Stapelprobleme	17
2.3.1 Einfache Eigenschaften	17
2.3.2 Struktur zweistufig lösbarer Stapelprobleme mit $n = \pi(n)$.	29
2.3.3 Fallunterscheidungen für $n \neq \pi(n)$	33
2.3.4 Übersicht der zu untersuchenden Fälle	34
2.3.5 Einfache Stapelprobleme sind leicht zu lösen	63
3 Verzweigte Stapel	65
3.1 Lösbarkeit von verzweigten Stapelproblemen	68
3.2 NP-Vollständigkeit	71
3.3 Lösungsstrategien	73
3.3.1 Obere und untere Schranken	73
3.3.2 Branching & Bounding	77
3.3.3 Heuristische Lösungsverfahren	78
3.3.4 Zeitkompliziertheit der Heuristik	79
3.3.5 Erfahrungen mit den Algorithmen	82
4 Nicht eindeutige Elementzuordnungen	87
4.1 Perfekte Matchings in paaren Graphen	87
4.2 Nicht eindeutige Stapelprobleme	89
4.3 Auswahl einer eindeutigen Zuordnung	91
5 Ausblick	108
Literaturverzeichnis	111

Kapitel 1

Einführung

Stapelprobleme treten in der Praxis in vielfältiger Form auf. So finden wir sie beispielsweise bei der Montage großformatiger Bauelemente im Bauwesen aber auch in einer großen Anzahl von Variationen im Logistikbereich. Optimierungsaufgaben, die bei der Be- und Entladung von Schiffen an Containerterminals oder an der Schnittstelle Straßentransport/Bahntransport auftreten, gehören offensichtlich in diese Problemklasse. Auch Sortier- und Reihenfolgeprobleme wie sie beispielsweise beim Rangieren von Schienenfahrzeugen auftreten, lassen sich als Stapelprobleme modellieren.

Wir betrachten eine gegebene Ausgangssituation als Ausgangsstapel – eine gewünschte Endsituation als Zielstapel. Zur mathematischen Beschreibung bedienen wir uns der Notation der Graphentheorie. Dabei werden die Stapel als gerichtete Graphen betrachtet: Die Stapелеlemente bilden die Knoten des Graphen, liegt ein Stapелеlement auf einem anderen, so existiert eine gerichtete Kante zwischen den entsprechenden Knoten im Graphen. Desweiteren ist zur vollständigen Repräsentation eines Stapelproblems immer eine Menge von Regeln notwendig, die beschreiben, welche Bewegungen von Elementen zwischen den Stapeln zugelassen sind. Dieses System von Regeln kann – wenigstens teilweise – ebenfalls als gerichteter Graph modelliert werden. Dementsprechend bezeichnen wir diesen Graph auch als Regelgraph.

Von der Ausgangssituation beginnend, werden nun nach jeder Elementbewegung die Zustände von Ausgangs- und Zielstapel und eventuell vorhandenen Hilfsstapelplätzen mit sogenannten Stapelkonfigurationen beschrieben.

In diesem Kapitel sollen die wichtigsten Begriffe bereitgestellt werden, die in der vorliegenden Arbeit benötigt werden. Weiterhin wird das wohl populärste Stapelproblem – Die Türme von Hanoi – erörtert.

1.1 Begriffe der Graphentheorie

Definition 1.1 Ein *gerichteter Graph* $G = (V, A)$ besteht aus einer Menge

$V = \{v_1, \dots, v_n\}$ von *Knoten* und einer Menge A von Elementen aus $V \times V$, die wir *gerichtete Kanten* oder *Bögen* nennen. Für eine solche Kante $(u, v) \in A$ heißt u *Anfangs-* und v *Endknoten*. Kanten (v, v) nennen wir *Schlingen*.

Definition 1.2 Ein *ungerichteter Graph* $G = (V, E)$ besteht aus einer Menge $V = \{v_1, \dots, v_n\}$ von Knoten und einer Kantenmenge E . Die Kanten sind Elemente der Menge der zweielementigen Teilmengen von V , wobei Kanten der Form $\{v, v\}$ (Schlingen) zugelassen sein sollen.

Ein Graph (gerichtet oder ungerichtet) heißt genau dann *endlich*, wenn seine Knotenmenge endlich ist, und genau dann *schlingenfrei*, wenn er keine Schlingen enthält.

In der vorliegenden Arbeit werden ausschließlich endliche und schlingenfreie Graphen betrachtet, die in der Regel gerichtet sind. Abweichungen hiervon werden im Text ausdrücklich erwähnt.

Die Knotenmenge V eines Graphen $G = (V, A)$ werden wir auch mit $V(G)$, die Kantenmenge A mit $A(G)$ bezeichnen. Sofern nichts anderes festgelegt wird, gelte $n := |V|$ und $m := |A|$.

Ist $(u, v) \in A$ eine Kante von G , so heißen die Knoten u und v *benachbart* oder *adjazent*, außerdem nennen wir u bzw. v mit (u, v) *inzident*. Zwei verschiedene Kanten heißen genau dann *adjazent*, wenn sie mit einem gemeinsamen Knoten inzident sind.

Ein Knoten v eines Graphen heißt genau dann *isolierter Knoten*, wenn v mit keinem anderen Knoten adjazent ist. Eine Kante e eines Graphen heißt genau dann *isolierte Kante*, wenn e mit keiner weiteren Kante des Graphen adjazent ist.

Definition 1.3 Für einen gerichteten Graphen $G = (V, A)$ und einen Knoten $v \in V$ seien folgende Mengen definiert:

$$N_G^+(v) := \{u \in V : (v, u) \in A\}, \quad N_G^-(v) := \{u \in V : (u, v) \in A\}.$$

Die Elemente aus $N_G^+(v)$ heißen auch *Nachfolger* von v in G , die Knoten aus $N_G^-(v)$ entsprechend *Vorgänger* von v in G .

Weiterhin heißt $N_G(v) := N_G^+(v) \cup N_G^-(v)$ *offene Nachbarschaft* von v in G . Für einen ungerichteten Graphen gilt

$$N_G(v) := \{u \in V : \{v, u\} \in A\},$$

Für gerichtete oder ungerichtete Graphen heißt

$$N_G[v] := N_G(v) \cup \{v\}$$

geschlossene Nachbarschaft von v in G . Für eine Knotenmenge $V' := \{v_1, \dots, v_i\}$ definieren wir zusätzlich

$$\begin{aligned} N_G(V') &:= N_G(v_1) \cup N_G(v_2) \cup \dots \cup N_G(v_i) \\ \text{und } N_G[V'] &:= N_G[v_1] \cup N_G[v_2] \cup \dots \cup N_G[v_i]. \end{aligned}$$

Sofern keine Verwechslungsgefahr besteht, werden wir auch $N(v)$, $N[v]$, $N^+(v)$, $N^-(v)$ und $N(V)$ bzw. $N[V]$ schreiben.

Definition 1.4 Es sei G ein Graph und $v \in V(G)$ ein Knoten dieses Graphen. Dann bezeichnet die *Valenz* $d_G(v)$ in G die Anzahl der mit v inzidenten Kanten und ist wie folgt definiert:

$$d_G(v) := |N_G(v)|.$$

Für einen gerichteten Graphen unterscheiden wir zusätzlich die Anzahl eingehender Kanten (*Eingangsvalenz* $d_G^-(v)$) und ausgehender Kanten (*Ausgangsvalenz* $d_G^+(v)$):

$$d_G^+(v) := |N_G^+(v)|, \quad d_G^-(v) := |N_G^-(v)|.$$

Analog zur Nachbarschaft werden wir auch die Bezeichnungen $d(v)$, $d^+(v)$ und $d^-(v)$ verwenden, wenn die Gefahr einer Verwechslung ausgeschlossen ist.

Definition 1.5 Zwei Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ heißen genau dann *isomorph*, wenn es eine eineindeutige Abbildung $f : V_1 \rightarrow V_2$ gibt, so daß

$$\forall u, v \in V_1 : \{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2,$$

bzw. im Falle von gerichteten Graphen

$$\forall u, v \in V_1 : (u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2,$$

gilt. Wir schreiben dann $G_1 \cong G_2$.

Definition 1.6 Der Graph $G_1 = (V_1, E_1)$ heißt genau dann *Untergraph* des Graphen $G = (V, E)$ (wir schreiben: $G_1 \subseteq G$), wenn $V_1 \subseteq V$ und $E_1 \subseteq E$ gilt.

Ein Untergraph $G_2 = (V_2, E_2)$ von $G = (V, E)$ heißt genau dann *induzierter Untergraph*, wenn

$$\forall u, v \in V_2 \text{ ist } \{u, v\} \in E_2 \Leftrightarrow \{u, v\} \in E$$

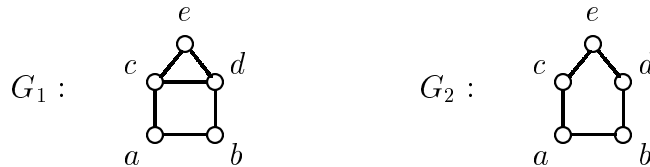
gilt. Für den Fall, daß $G = (V, A)$ und $G_2 = (V_2, A_2)$ gerichtet sind, heißt G_2 genau dann *induzierter Untergraph*, wenn analog

$$\forall u, v \in V_2 \text{ ist } (u, v) \in A_2 \Leftrightarrow (u, v) \in A$$

gilt. Wir nennen dann G_2 von V_2 induziert und schreiben $G_2 = G_{V_2}$ oder auch $G_2 = G(V_2)$.

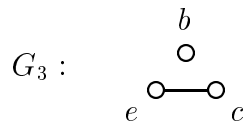
Beispiel:

$$\begin{aligned}
 G_1 &= (V_1, E_1), & V_1 &= \{a, b, c, d, e\}, & E_1 &= \{\{a, b\}, \{a, c\}, \{b, d\}, \\
 & & & & & \{c, d\}, \{c, e\}, \{d, e\}\} \\
 G_2 &= (V_2, E_2), & V_2 &= V_1, & E_2 &= \{\{a, b\}, \{a, c\}, \{b, d\}, \\
 & & & & & \{c, e\}, \{d, e\}\}
 \end{aligned}$$



Dann ist G_2 zwar Untergraph von G_1 , aber kein *induzierter* Untergraph von G_1 .

$$G_3 = (V_3, E_3), \quad V_3 = \{b, c, e\}, \quad E_3 = \{\{c, e\}\}$$



G_3 ist (induzierter) Untergraph sowohl von G_1 , als auch von G_2 .

$$G_3 = G_{V_3} = G_{\{b,c,e\}}.$$

Definition 1.7 Ein gerichteter Graph $G = (V, A)$ heißt genau dann

- i) *Pfad* der Länge $n - 1$, $n \geq 1$, wenn $V = \{v_1, \dots, v_n\}$ und $A = \{(v_i, v_{i+1}) : i \in \{1, \dots, n - 1\}\}$ ist; v_1 nennen wir *Anfangs-* und v_n *Endknoten* des Pfades.
- ii) *Kreis* der Länge n , $n \geq 3$, wenn $V = \{v_1, \dots, v_n\}$ und $A = \{(v_i, v_{i+1}) : i \in \{1, \dots, n - 1\}\} \cup \{(v_1, v_n)\}$ ist.

Einen Pfad der Länge $n - 1$ bezeichnen wir auch mit P_n und einen Kreis der Länge n mit C_n .

Definition 1.8 Es seien $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ gerichtete oder ungerichtete Graphen. Dann soll $G_1 \cup G_2$ die *Vereinigung von G_1 und G_2* bezeichnen:

$$G_1 \cup G_2 := (V_1 \cup V_2, E_1 \cup E_2).$$

Für gerichtete oder ungerichtete Graphen $G = (V, E)$ mit $e \in E$ und $u \in V$ bezeichne $G \setminus e$ denjenigen Graphen, für den in G die Kante e gelöscht wurde: $G \setminus e := (V, E \setminus \{e\})$.

$G \setminus \{u\}$ bezeichne jenen Graphen, für den in G der Knoten u mit allen inzidenten Kanten gelöscht wurde: $G \setminus \{u\} := (V \setminus \{u\}, E \setminus \{e : u \text{ inzident mit } e\})$.

1.2 Die Türme von Hanoi

Eines der bekanntesten Probleme, das sich in die Kategorie Stapelprobleme einordnen läßt, ist sicher „Die Türme von Hanoi“.

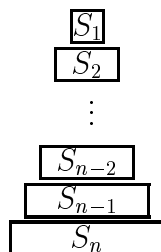
Die Problemstellung ist allseits bekannt: Gegeben ist ein Turm, bestehend aus n verschieden großen Scheiben, der so aufgebaut ist, daß die kleinste Scheibe, die wir mit S_1 bezeichnen, ganz oben, die größte Scheibe, die wir mit S_n bezeichnen, ganz unten liegen und eine Scheibe S_i immer auf der nächstgrößeren Scheibe S_{i+1} liegt.

Dieser Turm soll nun auf einen anderen Stapelplatz umgesetzt werden, wobei folgende Einschränkungen zu berücksichtigen sind:

- Es darf genau ein Hilfsstapelplatz dazu verwendet werden.
- Es darf immer nur eine einzelne Scheibe bewegt werden.
- Eine Scheibe darf nur auf einen leeren Stapelplatz gelegt werden, oder auf eine größere Scheibe.

Gefragt ist nun nach der minimalen Anzahl von Umstapelungen, um den vollständigen Turm auf dem Zielstapelplatz entstehen zu lassen.

Ausgangssituation:

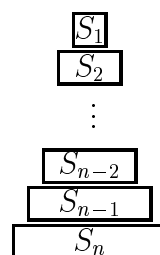


Ausgangsstapel

Zielstapel

Hilfsstapel

Endsituation:



Ausgangsstapel

Zielstapel

Hilfsstapel

1.2.1 Historisches

Das Turm von Hanoi Puzzle tauchte erstmals 1883 in Paris auf, als Erfinder gilt der französische Zahlentheoretiker Édouard Lucas (1842-91). Geschichtliche Details sowie grundlegende mathematische Eigenschaften des Problems finden sich bei HINZ [3]. Besonders in den letzten drei Jahrzehnten wurde das Problem auf vielfältige Weise variiert und unter algorithmischen Gesichtspunkten betrachtet: A. J. VAN ZANTEN [18] betrachtet eine Abwandlung des Problems, bei der die Scheiben in der Ausgangssituation über die drei Stapelplätze verteilt liegen und in eine Endsituation überführt werden sollen bei der wiederum alle drei Plätze mit Scheiben belegt sein können. Eine ähnliche Fragestellung untersuchen KLEIN und MINSKER [10] – hierbei dürfen jedoch zusätzlich kleine auf größeren Scheiben liegen.

Weiterhin betrachtet MINSKER [12] den Fall von drei gleichen Scheibenstapeln, von denen jeder auf einem der drei Stapelplätze liegt und deren Scheiben einmal rot, einmal blau und einmal grün gefärbt sind. Hierbei ist es das Ziel, die Stapel zyklisch auf den Stapelplätzen zu vertauschen.

WU und CHEN [17] führen Parallelbewegungen zur Lösung des Turm von Hanoi Problems ein. Dabei darf eine Scheibe von einem, zwei oder allen drei Stapel gleichzeitig entnommen werden und im Uhrzeigersinn auf dem jeweils nächsten Stapelplatz gelegt werden.

Erstaunlicherweise wurde die dem Original nächstliegende Fragestellung – nach der Anzahl notwendiger Bewegungen zur Lösung bei k Stapelplätzen – erst 1995 von MAJUMDAR [9] gelöst.

Um die ursprüngliche Aufgabenstellung der Türme von Hanoi zu lösen, wollen wir kurz betrachten, wie oft wir eine Scheibe nehmen und woanders ablegen müssen:

Wollen wir einen Turm der Höhe i umstapeln, so müssen wir den „Teilturm“ S_1, \dots, S_{i-1} , der auf der Scheibe S_i liegt, auf einen dritten Stapelplatz stellen, die Scheibe S_i umstapeln und den „Teilturm“ S_1, \dots, S_{i-1} wieder auf S_i stellen.

Jedesmal, wenn wir die Scheibe S_i einmal umsetzen, müssen wir die Scheibe S_{i-1} zweimal bewegen (von der Scheibe S_i wegnehmen und wieder auf die Scheibe S_i ablegen).

Für eine Scheibe S_i mit $i \in \{1, \dots, n-1\}$ müssen wir also doppelt so viele Bewegungen ausführen, wie für die Scheibe S_{i+1} .

Da diese Aussage für alle $i \in \{1, \dots, n-1\}$ gilt und wir die letzte und größte Scheibe S_n nur einmal bewegen müssen, so können wir nun berechnen, wie oft jede Scheibe bewegt werden muß:

Scheibe	Anzahl der Bewegungen
S_n	1
S_{n-1}	2
S_{n-2}	4 (bei jeder Bewegung von S_{n-1} muß die Scheibe S_{n-2} zweimal bewegt werden)
S_{n-3}	8
\vdots	\vdots
S_{n-k}	2^k
\vdots	\vdots
S_1	2^{n-1}

Die Gesamtanzahl der Bewegungen ergibt sich nun als Summe der Anzahl der Bewegungen jeder einzelnen Scheibe:

$$\sum_{i=1}^n 2^{n-i} = 2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0 = \sum_{k=0}^{n-1} 2^k = 2^n - 1.$$

Um einen Turm mit n Elementen umzustapeln, benötigt man also mindestens $2^n - 1$ Schritte.

Nun können wir den Fall betrachten, daß es im Turm mehrere Scheiben der gleichen Größe gibt, sagen wir: Die Scheibe S_i der Größe i tritt genau k_i mal auf. (Im Turm sollen dann natürlich alle Scheiben einer Größe aufeinander liegen.) n repräsentiert jetzt nicht mehr die Anzahl aller Scheiben, sondern nur noch die Anzahl der Scheiben unterschiedlicher Größe. Welche Auswirkungen hat das auf die Anzahl der zur Lösung notwendigen Schritte?

Haben wir bisher *eine* Scheibe der Größe i bewegt, so müssen wir nun jedesmal k_i solcher Scheiben umstapeln. Es ist also für jede Scheibe S_i die Anzahl ihrer notwendigen Bewegungen in unserem bisherigen Modell mit der Anzahl der Scheiben, die die gleiche Größe wie S_i haben, zu multiplizieren.

Scheibengröße	Anzahl der Scheiben	Anzahl der Bewegungen
S_n	k_n	k_n
S_{n-1}	k_{n-1}	$2k_{n-1}$
S_{n-2}	k_{n-2}	$4k_{n-2}$
S_{n-3}	k_{n-3}	$8k_{n-3}$
\vdots	\vdots	\vdots
S_{n-k}	k_{n-k}	$2^k k_{n-k}$
\vdots	\vdots	\vdots
S_1	k_1	$2^{n-1} k_1$

Die Gesamtzahl an Bewegungen zur Lösung ergibt sich wieder als Summe über

die Bewegungen jeder einzelnen Scheibengröße:

$$\sum_{i=1}^n 2^{n-i} k_i.$$

Betrachten wir nun wieder unser Ausgangsmodell als Spezialfall: Für alle $i \in \{1, \dots, n\}$ gilt $k_i = 1$, so ergibt sich wieder:

$$\sum_{i=1}^n 2^{n-i} = \sum_{k=0}^{n-1} 2^k.$$

1.2.2 Ein minimales Regelsystem

Definition 1.9 Für zwei natürliche Zahlen m und n definieren wir $P(n, m)$ wie folgt: $P(n, m) := \begin{cases} 1 & \text{für } n + m \text{ gerade} \\ 0 & \text{für } n + m \text{ ungerade} \end{cases}$.

Wir sagen, daß n und m die gleiche Parität haben, wenn $P(n, m) = 1$ gilt.

Versuchen wir, das Modell etwas „mathematischer“ zu gestalten: Zunächst können wir die Einschränkung, daß eine Scheibe S_i nur auf eine größere Scheibe S_j ($j > i$) gelegt werden darf, formulieren: Wir betrachten die Scheiben S_k des Turmes als Knoten k und tragen zwischen zwei Knoten i und j eine gerichtete Kante (i, j) ein, genau dann, wenn wir die Scheibe S_i auf die Scheibe S_j legen dürfen. Mit anderen Worten: Es existiert eine Kante von i nach j genau dann, wenn $i < j$ ist.

Den so generierten gerichteten Graphen $G_R = (V_R, A_R)$ mit $V_R = \{1, \dots, n\}$ und $A_R = \{(i, j) : i, j \in \{1, \dots, n\} : i < j\}$ nennen wir *Regelgraphen*. Die Kanten (i, j) aus A_R nennen wir *Regeln*.

Macht man sich nun die Mühe, tatsächlich nachzuvollziehen, wie die Lösungsschritte zum Umsetzen des Turms aussehen, so stellt man sehr schnell fest, daß gar nicht alle zur Verfügung stehenden Regeln angewendet werden. Wir wollen nun zeigen, welche Regeln tatsächlich benötigt werden, um einen Turm der Höhe $n + 1$ umzusetzen:

Zunächst können wir den „Teilturm“ S_1, \dots, S_n der Höhe n ohne weiteres auf den Hilfsstapel ausstapeln, dazu benötigen wir genau diejenigen Regeln, die zum Umsetzen des Turmes auf den Zielstapel nötig sind. Anschließend legen wir die Scheibe S_{n+1} auf den Zielstapel. Nun verfolgen wir etwas genauer, wie wir den Turm der Höhe n auf die Scheibe S_{n+1} auf dem Zielstapel setzen:

Es ist unser Ziel, die Scheibe S_n auf die Scheibe S_{n+1} zu legen. Dazu müssen die Scheiben S_1 bis S_{n-1} auf dem Ausgangsstapel liegen. Wie gelangt aber die Scheibe S_{n-1} auf diesen Platz? Um die Scheibe S_{n-1} vom Hilfsstapel auf den Ausgangsstapel umzusetzen, müssen die Scheiben S_1 bis S_{n-2} auf dem Zielstapel liegen — dort wo schon die Scheibe S_{n+1} liegt. Insbesondere liegt also die Scheibe

S_{n-2} auf der Scheibe S_{n+1} . Somit ist das Vorhandensein der Regel $(n-2, n+1)$ für die Lösbarkeit unabdingbar.

Wie gelangt aber S_{n-2} auf S_{n+1} ? Dazu müssen die Scheiben S_1 bis S_{n-3} auf dem Ausgangsstapel liegen. Wie gelangt aber die Scheibe S_{n-3} dorthin? Um dies zu erreichen, muß wieder ein Turm der Scheiben S_1, \dots, S_{n-4} auf dem Zielstapel liegen — insbesondere also S_{n-4} auf S_{n+1} .

Wenn wir dieses Frage-Antwort-Spiel solange fortsetzen, bis wir bei der kleinsten Scheibe angekommen sind, so erfahren wir, daß wir die Scheiben $S_{n-2}, S_{n-4}, S_{n-6}, \dots, S_{n-(n-2)} = S_2$ (für n gerade) bzw: $S_{n-2}, S_{n-4}, S_{n-6}, \dots, S_{n-(n-1)} = S_1$ (für n ungerade) auf S_{n+1} legen müssen, um den Turm der Höhe $n-1$ auf den Ausgangsstapel zu stellen.

Nun können wir also die Scheibe S_n vom Hilfsstapel auf den Zielstapel (auf die Scheibe S_{n+1}) legen und nun den „Teilturm“ S_1, \dots, S_{n-1} auf den Zielstapel setzen, ohne das wir berücksichtigen müssen, daß dort schon die Scheibe S_{n+1} liegt. Mit den zusätzlichen Regeln

$$\begin{aligned} &(n, n+1), (n-2, n+1), (n-4, n+1), \dots, (2, n+1) && \text{(für } n \text{ gerade)} \\ &(n, n+1), (n-2, n+1), (n-4, n+1), \dots, (1, n+1) && \text{(für } n \text{ ungerade)} \end{aligned}$$

von denen jede zur Lösbarkeit notwendig ist, ist somit der Turm der Höhe $n+1$ vom Ausgangs- auf den Zielstapel umstapelbar.

Lemma 1.1 Ein Hanoi-Turm der Höhe n ist mit dem Regelgraphen $G_{R_n} = (V_{R_n}, A_{R_n})$ mit

$$\begin{aligned} V_{R_n} &= \{1, \dots, n\} \text{ und} \\ A_{R_n} &= \{(i, i+3), (i, i+5), (i, i+7), \dots, (i, n-P(i, n)) : i \in \{1, \dots, n-3\}\} \\ &\quad \cup \{(i, i+1) : i \in \{1, \dots, n-1\}\} \end{aligned}$$

umstapelbar.

Bemerkung: In der obigen Beschreibung für A_{R_n} bedeutet $(i, n-P(i, n))$ nach Definition 1.9: Für den Fall, daß i und n die gleiche Parität besitzen, gilt $(i, n-1) \in A_{R_n}$, sonst gilt $(i, n) \in A_{R_n}$.

Beweis: Wir führen den Beweis mittels vollständiger Induktion über n .

Für $n=4$ gilt: $A_{R_4} = \{(1, 2), (2, 3), (3, 4), (1, 4)\}$.

Angenommen, Lemma 1.1 gälte für $n=k$: Ein Turm der Höhe k ist also mit $G_{R_k} = (V_{R_k}, A_{R_k})$ umstapelbar. Weiter oben haben wir bereits gezeigt, daß wir als zusätzliche Regeln (um einen Turm der Höhe $k+1$ umzusetzen) nur die Regeln

$$\begin{aligned} &(k, k+1), (k-2, k+1), (k-4, k+1), \dots, (2, k+1) && \text{(für } k \text{ gerade)} \\ &(k, k+1), (k-2, k+1), (k-4, k+1), \dots, (1, k+1) && \text{(für } k \text{ ungerade)} \end{aligned}$$

benötigen. Die Kante $(k, k + 1)$ entspricht der Kante $(i, i + 1)$ mit $i = k$ in $A_{R_{k+1}}$. Die Kante $(k - 2, k + 1)$ entspricht den Kanten $(i, i + 3), \dots, (i, k + 1 - P(i, k + 1))$ für $i = k - 2$ aus $A_{R_{k+1}}$ ($k - 2$ und $k + 1$ haben verschiedene Paritäten).

Die Kanten $(1, k + 1), \dots, (k - 4, k + 1)$ bzw. $(2, k + 1), \dots, (k - 4, k + 1)$ entsprechen der Kante $(i, k + 1 - P(i, k + 1))$ für $i = 1, \dots, k - 3$ aus $A_{R_{k+1}}$. (Dabei kommt nur jede zweite der $k - 3$ Kanten hinzu.)

Damit ist: $A_{R_k} \cup \{(k - 2, k + 1), \dots, (1, k + 1)\} \cup \{(k, k + 1)\} = A_{R_{k+1}}$ bzw. $A_{R_k} \cup \{(k - 2, k + 1), \dots, (2, k + 1)\} \cup \{(k, k + 1)\} = A_{R_{k+1}}$

Somit ist ein Turm der Höhe $k + 1$ mit dem Regelgraphen $G_{R_{k+1}} = (V_{R_{k+1}}, A_{R_{k+1}})$ umstapelbar. \square

Wir haben also gezeigt, daß die zur Lösbarkeit des Problems notwendige Regelmengemenge auch hinreichend ist. Damit erhalten wir:

Lemma 1.2 Für jeden Regelgraphen $G'_{R_n} = (V'_{R_n}, A'_{R_n})$ mit dem ein Hanoi-Turm der Höhe n umstapelbar ist, gilt: $|A'_{R_n}| \geq |A_{R_n}|$.

Lemma 1.3 Der Regelgraph G_{R_n} ist paar.

Beweis: Da es in der Kantenmenge A_{R_n} nur Kanten zwischen Knoten verschiedener Parität gibt und zwischen je zwei Knoten verschiedener Parität auch immer eine Kante existiert können wir A_{R_n} wie folgt interpretieren:

$$\forall i, j \in \{1, \dots, n\} : (i, j) \in A_{R_n} \iff i < j : P(i, j) = 0.$$

Die Knotenmenge V_{R_n} läßt sich in zwei unabhängige, disjunkte Mengen teilen — nämlich die der geraden Knoten und die der ungeraden Knoten: $V_{R_n} = V_{R_n\text{-gerade}} \cup V_{R_n\text{-ungerade}}$. Beim Regelgraphen G_{R_n} handelt es sich also um einen paaren Graphen. \square

Lemma 1.4 Der Regelgraph $G_{R_n} = (V_{R_n}, A_{R_n})$ enthält keinen induzierten Pfad der Länge 3.

Beweis: Angenommen, G_{R_n} enthielte einen induzierten Pfad der Länge 3. Wir bezeichnen diesen Pfad mit $[x, y, z, r]$ mit $x, y, z, r \in V_{R_n}$ und $(x, y), (y, z), (z, r) \in A_{R_n}$. Nach Lemma 1.3 folgt o.B.d.A.: $x, z \in V_{R_n\text{-gerade}}$ und $y, r \in V_{R_n\text{-ungerade}}$. Aus $(x, y), (y, z), (z, r) \in A_{R_n}$ folgt: $x < y < z < r$. Somit besitzen x und r verschiedene Paritäten, mit dem Beweis von Lemma 1.3 folgt: $(x, r) \in A_{R_n}$, darum kann $[x, y, z, r]$ kein induzierter P_4 sein. Das widerspricht jedoch der Annahme, somit enthält G_{R_n} keinen induzierten P_4 . \square

Kapitel 2

Einfache Stapel

2.1 Formale Beschreibung von Stapelproblemen

In diesem Abschnitt wollen wir kurz eine formal-theoretische Beschreibung von Stapelproblemen angeben, die es uns ermöglicht, die Problemstellung in einem streng mathematischen Sinn zu erfassen und damit zu arbeiten. Es handelt sich hierbei keineswegs um Standardnotationen, da Stapelprobleme in dieser Interpretation in der Literatur nicht vorkommen und somit auch keine übliche Schreibweise existiert.

Definition 2.1 Ein Graph $G = (V, A)$ heißt genau dann (*verzweigter*) *Stapel*, wenn G endlich, schlingenfremd, gerichtet und zyklensfrei ist.

Definition 2.2 Ein Stapel $G = (V, A)$ mit $V = \{v_1, \dots, v_n\}$ heißt genau dann *einfach*, wenn $A = \{(v_i, v_{i+1}) : i \in \{1, \dots, n-1\}\}$ ist – G ein einfacher gerichteter Pfad ist.

Definition 2.3 $P = (G_A, G_Z, G_R, k)$ nennen wir genau dann ein *einfaches Stapelproblem*, wenn $G_A = (V, A_A)$ und $G_Z = (V, A_Z)$ einfache Stapel sind, $G_R = (V, A_R)$ ein gerichteter Graph ist und k eine natürliche Zahl. G_A nennen wir *Ausgangsstapel*, G_Z nennen wir *Zielstapel* und G_R nennen wir *Regelgraph*. k gibt die Anzahl der *Hilfsstapel* an. Die Knoten v aus V nennen wir Stapel-elemente.

Definition 2.4 Zwei Stapelprobleme $P = (G_A, G_Z, G_R, k)$ und $P' = (G'_A, G'_Z, G'_R, l)$ heißen genau dann *isomorph*, wenn $k = l$ gilt und wenn es eine eindeutige Abbildung $f : V(G_A) \rightarrow V(G'_A)$ gibt, so daß für alle $u, v \in V(G_A)$ gilt:

$$\begin{aligned}(u, v) \in A(G_A) &\iff (f(u), f(v)) \in A(G'_A), \\(u, v) \in A(G_Z) &\iff (f(u), f(v)) \in A(G'_Z) \text{ und} \\(u, v) \in A(G_R) &\iff (f(u), f(v)) \in A(G'_R).\end{aligned}$$

Wir schreiben dann: $P \cong P'$.

Lemma 2.1 Für jedes einfache Stapelproblem $P = (G_A, G_Z, G_R, k)$ existiert ein isomorphes Stapelproblem $P' = (G'_A, G'_Z, G'_R, k)$, für das

$$G'_A = (\{1, \dots, |V(G_A)|\}, \{(i, i+1) : i \in \{1, \dots, |V(G_A)| - 1\}\})$$

gilt.

Beweis: Das gewünschte isomorphe Problem P' läßt sich ganz einfach beschaffen, indem die Elemente in G_A von oben nach unten durchnummeriert werden: Wir betrachten eine eindeutige Funktion π , die jedem Knoten v aus $V(G_A)$ eine natürliche Zahl $\pi(v)$ aus $\{1, \dots, |V(G_A)|\}$ zuordnet:

$$\pi(v) := \begin{cases} 1 & \text{für } d_{G_A}^-(v) = 0 \\ \pi(u) + 1 & \text{für } (u, v) \in A(G_A) \end{cases} .$$

Weiterhin sei $\Pi(V) := \{\pi(v) : v \in V(G_A)\}$.

Wir definieren nun $P' := (G'_A, G'_Z, G'_R, k)$ wie folgt:

$$\begin{aligned} G'_A &= (\Pi(V), \{(\pi(u), \pi(v)) : (u, v) \in A(G_A)\}), \\ G'_Z &= (\Pi(V), \{(\pi(u), \pi(v)) : (u, v) \in A(G_Z)\}), \\ G'_R &= (\Pi(V), \{(\pi(u), \pi(v)) : (u, v) \in A(G_R)\}). \end{aligned}$$

Somit ist $\{(\pi(u), \pi(v)) : (u, v) \in A(G_A)\} = \{(i, i+1) : i \in \{1, \dots, |V(G_A)| - 1\}\}$. Offensichtlich erfüllt π die Anforderungen an die Funktion f aus Definition 2.4. Damit folgt $P' \cong P$. \square

Definition 2.5 Es sei $P = (G_A, G_Z, G_R, k)$ ein einfaches Stapelproblem. $P' = (G'_A, G'_Z, G'_R, k)$ sei ein einfaches Stapelproblem, für das

- i) $P' \cong P$,
- ii) $V(G_{A'}) = \{1, \dots, n\}$,
- iii) $A(G_{A'}) = \{(1, 2), (2, 3), (3, 4), \dots, (n, n-1)\}$ und
- iv) $A(G_{Z'}) = \{(\pi(1), \pi(2)), (\pi(2), \pi(3)), \dots, (\pi(n), \pi(n-1))\}$ gilt.

P' nennen wir dann *normiertes Stapelproblem* und $\pi(P)$ bezeichne die Permutation der Stapelelemente auf dem Zielstapel in P' :

$$\pi(P) := [\pi(1), \pi(2), \dots, \pi(n)]$$

$V(P)$ bezeichne die Menge der Stapelelemente in P : $V(P) := V(G_A)$.

Definition 2.6 $T = (G_1, G_2, \dots, G_{k+2})$ heißt genau dann *Stapelkonfiguration eines einfachen Stapelproblems* $P = (G_A, G_Z, G_R, k)$, wenn für alle $i \in \{1, \dots, k+2\}$ G_i einfache Stapel sind und weiterhin

- i) $\bigcup_{i=1}^{k+2} V(G_i) = V(G_A)$ ist,
- ii) für alle $i, j \in \{1, \dots, k+2\}$ mit $i \neq j$ $V(G_i) \cap V(G_j) = \emptyset$ gilt und
- iii) $\bigcup_{i=1}^{k+2} A(G_i) \subseteq A(G_R)$ ist.

Eine Stapelkonfiguration $T = (G_1, \dots, G_{k+2})$ repräsentiert demnach den Zustand aller Stapel zu einem festen Zeitpunkt. Jeder Stapel wird als ein Graph G_i beschrieben. Hier bezeichnen G_1 und G_2 Ausgangs- und Zielstapel. Die Graphen G_3 bis G_{k+2} stellen die Hilfsstapel dar. Wird von einem beliebigen Stapel ein Element weggenommen und auf einen anderen Stapel gelegt, so erhalten wir eine neue Stapelkonfiguration. Diese Operation nennen wir *Umstapelung*:

Definition 2.7 Gegeben seien 2 Stapelkonfigurationen $T = (G_1, G_2, \dots, G_{k+2})$ und $T' = (G'_1, G'_2, \dots, G'_{k+2})$ eines einfachen Stapelproblems $P = (G_A, G_Z, G_R, k)$. T' heißt genau dann *Umstapelung von T* , wenn zwei verschiedene Stapel G_i und G_j mit $i, j \in \{1, \dots, k+2\}$ existieren, für die gilt:

- i) es existiert ein a mit $d_{G_i}^-(a) = 0$,
- ii) $G_j = (\emptyset, \emptyset)$ oder es existiert ein $b \in G_j$ mit $(a, b) \in A(G_R)$ und $d_{G_j}^-(b) = 0$,
- iii) für alle $l \in \{1, \dots, k+2\} \setminus \{i, j\}$ gilt $G_l = G'_l$,
- iv) $G'_i = G_i \setminus \{a\}$ und
- v) $G'_j = (V(G_j) \cup \{a\}, \emptyset)$ für $G_j = (\emptyset, \emptyset)$ bzw.
 $G'_j = (V(G_j) \cup \{a\}, A(G_j) \cup \{(a, b)\})$ für $V(G_j) \neq \emptyset$.

Wir schreiben auch $T \xrightarrow{[i,j]} T'$. Für $j = 1$ nennen wir T' *Ausgangs-Rückstapelung von T* , für $i = 2$ nennen wir T' *Ziel-Rückstapelung von T* . Für $i = 1$ und $j = 2$ nennen wir T' *direkte Zielumstapelung von T* , für $i, j > 2$ heißt T' *Hilfsumstapelung von T* .

Definition 2.8 Sei $P = (G_A, G_Z, G_R, k)$ ein einfaches Stapelproblem. Eine Folge $\mathcal{T} = [T_1, \dots, T_l]$ von Stapelkonfigurationen $T_i = (G_1^i, G_2^i, \dots, G_{k+2}^i)$ heißt genau dann *zulässige Stapelfolge von P* , wenn gilt:

- i) $G_1^1 = G_A$,
- ii) $G_2^1 = G_3^1 = \dots = G_{k+2}^1 = (\emptyset, \emptyset)$ und
- iii) für $i \in \{1, \dots, l-1\}$ ist T_{i+1} Umstapelung von T_i .

Die Stapelkonfigurationen T_i nennen wir dann *zulässige Stapelkonfigurationen* von P .

Definition 2.9 Ein einfaches Stapelproblem $P = (G_A, G_Z, G_R, k)$ heißt genau dann *lösbar*, wenn eine zulässige Stapelfolge $\mathcal{T} = [T_1, \dots, T_l]$ existiert, so daß für $T_l = (G_1^l, G_2^l, \dots, G_{k+2}^l)$ gilt:

$$G_2^l = G_Z \text{ und } G_1^l = G_3^l = G_4^l = \dots = G_{k+2}^l = (\emptyset, \emptyset).$$

\mathcal{T} nennen wir dann *l-Umstapelfolge* für P .

Definition 2.10 Sei $P = (G_A, G_Z, G_R, k)$ ein einfaches Stapelproblem. Weiterhin existiere eine *l-Umstapelfolge* $\mathcal{T} = [T_1, \dots, T_l]$ für P . Für $i \in \{1, \dots, l\}$ sei $T_i = (G_1^i, \dots, G_{k+2}^i)$. Wir nennen \mathcal{T} genau dann

- *zweistufig*, wenn
 - für alle $i \in \{1, \dots, l-1\}$ T_{i+1} keine Ausgangs-Rückstapelung von T_i ist,
 - für alle $i \in \{1, \dots, l-1\}$ T_{i+1} keine Ziel-Rückstapelung von T_i ist,
 - für alle $i \in \{1, \dots, l-1\}$ T_{i+1} keine direkte Zielumstapelung von T_i ist und
 - für $j := \min\{i : G_2^i \neq (\emptyset, \emptyset)\}$ folgt: $G_1^{j-1} = (\emptyset, \emptyset)$.
- *streng zweistufig*, wenn \mathcal{T} zweistufig ist und für alle $i \in \{1, \dots, l-1\}$ T_{i+1} keine Hilfsumstapelung von T_i ist.

Besitzt P eine (streng) zweistufige *l-Umstapelfolge* \mathcal{T} , so nennen wir P (*streng*) *zweistufig lösbar*.

An dieser Stelle sei bemerkt, daß eine ganze Reihe weiterer Probleme denkbar sind, die sich aus weniger rigiden Definitionen ergeben. Wir wollen hier jedoch nur (streng) zweistufige Stapelprobleme betrachten.

2.2 Streng zweistufig lösbare Stapelprobleme

Das Modell der streng zweistufig lösbaren Stapelprobleme hat seinen Ursprung in der Sortierung von Permutationen mit Hilfe von Sortierschlangen und -stacks. GOLUMBIC geht in seinem Buch [2] auf diese Probleme näher ein. Wir wollen hier nur ein Kriterium zur Bestimmung der minimalen Anzahl an Hilfsstapeln, die zur Lösung des Problems notwendig sind, angeben sowie auf einen effizienten Lösungsalgorithmus verweisen.

Nach Lemma 2.1 reicht es aus, normierte Probleme P zu betrachten, für die nach Definition 2.5

$$G_A = (\{1, \dots, n\}, \{(i, i+1) : i \in \{1, \dots, n-1\}\}) \quad (2.1)$$

gilt. Für den Zielgraphen können wir so jede Permutation π der Zahlen 1 bis n zulassen, so daß

$$G_Z = (\{1, \dots, n\}, \{(\pi(i), \pi(i+1)) : i \in \{1, \dots, n-1\}\}) \quad (2.2)$$

gilt.

Bei der Wahl des Regelgraphen treffen wir keine Einschränkungen – wir nehmen einen vollständigen, gerichteten Graph als Regelgraphen:

$$G_R = (\{1, \dots, n\}, \{(i, j) : i \neq j, i, j \in \{1, \dots, n\}\}). \quad (2.3)$$

Definition 2.11 Für eine Permutation $\pi = [a_1, \dots, a_n]$ der Zahlen 1 bis n bezeichnet π^{-1} die zu π inverse Permutation:

$$\text{Für } \pi(i) = a_i \text{ ist } \pi^{-1}(a_i) := i.$$

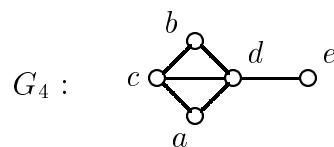
Definition 2.12 Für eine Permutation $\pi = [\pi(1), \dots, \pi(n)]$ der Zahlen 1 bis n ist der ungerichtete Graph $G[\pi] = (V, E)$ wie folgt definiert:

- i) $V := \{1, \dots, n\}$,
- ii) $\forall x, y \in V$ ist $\{x, y\} \in E \Leftrightarrow (x - y)(\pi^{-1}(x) - \pi^{-1}(y)) < 0$,

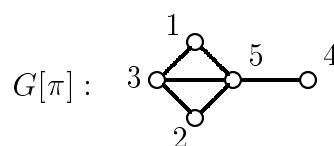
wobei $\pi^{-1}(x)$ die Position des Knotens x in der Permutation π angibt.

Definition 2.13 Ein ungerichteter Graph G heißt genau dann *Permutationsgraph*, wenn es eine Permutation π der Zahlen 1 bis n gibt, so daß $G \cong G[\pi]$ ist.

Beispiel:



G_4 ist Permutationsgraph, weil für die Permutation $\pi := [5, 3, 1, 2, 4]$ G_4 isomorph $G[\pi]$ ist:



Definition 2.14 Es sei $G = (V, E)$ ein ungerichteter Graph. Eine Knotenmenge $V' \subseteq V$ heißt *unabhängig* wenn die Knoten aus V' paarweise nicht adjazent sind.

Definition 2.15 Für einen ungerichteten Graphen $G = (V, E)$ bezeichnet $\chi(G)$ die *Färbungszahl* von G :

$$\chi(G) := \min\{k : \text{Für } V \text{ gibt es eine Zerlegung } V_1, \dots, V_k, \\ \text{so daß alle } V_i \text{ unabhängige Mengen sind.}\}$$

Satz 2.2 Es sei $P = (G_A, G_Z, G_R, k)$ ein einfaches normiertes Stapelproblem, mit G_A , G_Z und G_R wie in (2.1), (2.2) und (2.3) definiert. P ist genau dann streng zweistufig lösbar, wenn $\chi(G[\pi]) \leq k$ ist.

Beweis: („ \Rightarrow “) Es sei P streng zweistufig lösbar. Das bedeutet, der Lösungsvorgang benötigt im Höchstfall k Hilfsstapel. Wir können nun jedem Hilfsstapel eine Farbe zuordnen. Jedes Stapелеlement liegt während des Umstapelprozesses auf genau einem Hilfsstapel. Der korrespondierende Knoten im Permutationsgraphen $G[\pi]$ wird mit der Farbe dieses Hilfsstapels gefärbt. Es ergibt sich eine zulässige Färbung von $G[\pi]$.

Angenommen, das sei nicht so: Dann existieren Knoten u und v aus V mit $(u, v) \in A(G[\pi])$, die mit der gleichen Farbe gefärbt wurden. Das bedeutet, sie haben während des Umstapelprozesses auf einem gemeinsamen Hilfsstapel H_i gelegen.

O.B.d.A. wurde u zuerst auf H_i gelegt, das heißt $u < v$. Somit kann u aber erst nach v auf den Zielstapel gelegt werden, damit gilt $\pi^{-1}(u) < \pi^{-1}(v)$. Es gilt demnach $(u - v)(\pi^{-1}(u) - \pi^{-1}(v)) > 0$. Nach Definition von $G[\pi]$ gilt damit $(u, v) \notin A(G[\pi])$. Das widerspricht der Annahme, $G[\pi]$ ist deshalb mit k Farben zulässig gefärbt worden. Es gilt $\chi(G[\pi]) \leq k$.

(„ \Leftarrow “) Es gelte $\chi(G[\pi]) \leq k$. Wir wählen eine zulässige minimale Färbung von $G[\pi]$. Jeder dieser Farben wird nun ein Hilfsstapelplatz zugewiesen. Nun wird jedes Stapелеlement auf denjenigen Hilfsstapelplatz gelegt, der derjenigen Farbe zugeordnet ist, wie der entsprechende Knoten aus $G[\pi]$ in der gewählten Färbung.

Angenommen, der Zielstapel könnte nun nicht aufgebaut werden, dann gibt es Stapелеlemente u, v , die so auf einem Hilfsstapel H_i liegen, daß sie nicht in der richtigen Reihenfolge auf den Zielstapel gebracht werden können: O.B.d.A. wurde u vor v auf H_i gelegt und es gilt $\pi^{-1}(v) < \pi^{-1}(u)$. Wenn u aber vor v auf H_i gelegt wurde, gilt $u < v$. Somit ist $(u - v)(\pi^{-1}(u) - \pi^{-1}(v)) < 0$, nach Definition von $G[\pi]$ ist damit $(u, v) \in A(G[\pi])$. In einer zulässigen Färbung von $G[\pi]$ müssen u und v somit verschiedenen Farben haben – das jedoch widerspricht der Voraussetzung, da u und v beide auf H_i gelegt wurden. Damit ist die Annahme falsch und P ist streng zweistufig lösbar. \square

Die mindestens notwendige Anzahl an Hilfsstapelplätzen um ein einfaches Stapelproblem P mit dem vorgegebenen Regelgraphen streng zweistufig zu lösen ist

nach Satz 2.2 gleich der kleinsten Anzahl an Farben mit denen der Permutationsgraph $G[\pi]$ der Permutation des Zielstapels gefärbt werden kann. In GOLUMBIC [2] ist ein Algorithmus angegeben, der eine solche minimale Färbung in $O(n \log(n))$ berechnet.

2.3 Zweistufig lösbare Stapelprobleme mit 2 Hilfsstapeln

2.3.1 Einfache Eigenschaften

Wir kommen zurück auf Definition 2.10 aus Abschnitt 2.1. Für ein einfaches, zweistufig lösbares Stapelproblem P muß es eine Folge von Umstapelungen geben, bei denen

- niemals ein Element auf den Ausgangsstapel zurückgelegt wird,
- niemals ein Element vom Zielstapel weggenommen wird,
- niemals ein Element vom Ausgangsstapel direkt auf den Zielstapel gelegt wird,
- ein Element erst auf den Zielstapel gelegt werden darf, wenn der Ausgangsstapel leer ist.

Es werden also zuerst alle Elemente des Ausgangsstapels auf die Hilfsstapel gelegt und erst wenn kein Element mehr auf dem Ausgangsstapel liegt, darf der Zielstapel aufgebaut werden.

Es ist klar, daß die Lösbarkeit von einfachen Stapelproblemen nur von der Struktur des Regelgraphen und der Anzahl der Hilfsstapelplätze abhängt. Will man ein zweistufig lösbares, einfaches Stapelproblem P untersuchen, so ist es als minimale Forderung sicher sinnvoll, zu verlangen, daß der Regelgraph sowohl die Kanten des Ausgangs- als auch die des Zielgraphen enthält.

Wir wollen nun untersuchen, welche einfachen Stapelprobleme P mit einer fest vorgegebene Elementanzahl n unter dieser Mindestvoraussetzung zweistufig lösbar sind. Obwohl streng zweistufig lösbare Probleme auch zweistufig lösbar sind, ist die hier betrachtete Fragestellung eine ganz andere als im Abschnitt 2.2, weil hier mit einem ganz anders gearteten Regelgraphen gearbeitet wird.

Nach Lemma 2.1 reicht es aus, normierte Probleme P zu betrachten, für die nach Definition 2.5

$$G_A = (\{1, \dots, n\}, \{(i, i + 1) : i \in \{1, \dots, n - 1\}\})$$

ist. Für den Zielgraph können wir jede Permutation π der Zahlen 1 bis n zulassen, so daß

$$G_Z = (\{1, \dots, n\}, \{(\pi(i), \pi(i + 1)) : i \in \{1, \dots, n - 1\}\})$$

gilt. Für G_R gelte

$$G_R = (\{1, \dots, n\}, A(G_A) \cup A(G_Z)). \quad (2.4)$$

Regelgraphen dieser Form bezeichnen wir als *induzierte Regelgraphen*, entsprechend nennen wir Stapelprobleme mit induziertem Regelgraphen ebenfalls *induziert*. Ein solches Stapelproblem wird dann nur noch von der Anzahl n der Stapelelemente, der Anzahl k der Hilfsstapel und der Permutation π der Stapel­elemente im Zielgraphen bestimmt.

Definition 2.16 Für $\pi_1 = [a_1, \dots, a_k]$ und $\pi_2 = [a_{k+1}, \dots, a_n]$ definiert $\pi_1 \sqcup \pi_2$ die Permutation

$$\pi_1 \sqcup \pi_2 := [a_1, \dots, a_k, a_{k+1}, \dots, a_n].$$

Definition 2.17 Für $j \geq 0$ ist

$$\pi_{0,j} := [2, 4, 1, 6, 3, 8, 5, 10, 7, 12, \dots]$$

und

$$\pi_{1,j} := [3, 1, 5, 2, 7, 4, 9, 6, 11, 8, \dots],$$

wobei $\pi_{0,j}$ bzw. $\pi_{1,j}$ genau j Elemente besitzen.

Beispiele: $\pi_{0,0} = \pi_{1,0} = []$, $\pi_{0,1} = [2]$, $\pi_{1,1} = [3]$, $\pi_{1,7} = [3, 1, 5, 2, 7, 4, 9]$.

Definition 2.18 Es sei Π_n die Menge aller Permutationen der Zahlen 1 bis n , $\mathcal{P}_{n,k}$ die Menge aller einfachen, induzierten Stapelprobleme mit n Stapelelementen und k Hilfsstapeln, $L(\mathcal{P}_{n,k})$ die Menge aller zweistufig lös­baren Stapelprobleme aus $\mathcal{P}_{n,k}$ und $L_k(\Pi_n)$ die Menge aller Permutationen der Zahlen 1 bis n , für die die entsprechenden Stapelprobleme P zweistufig lösbar sind:

$$\begin{aligned} \Pi_n &:= \{\pi : \pi \text{ Permutation der Zahlen 1 bis } n\}, \\ \mathcal{P}_{n,k} &:= \{P : P = (G_A, G_Z, G_R, k) \text{ einfaches induziertes Stapelproblem} \\ &\quad \text{mit } |V(P)| = n, k \in N, \pi(P) \in \Pi_n\}, \\ L(\mathcal{P}_{n,k}) &:= \{P : P \in \mathcal{P}_{n,k}; P \text{ zweistufig lösbar}\}, \\ L_k(\Pi_n) &:= \{\pi : P = (G_A, G_Z, G_R, k) \in L(\mathcal{P}_{n,k}); \pi(P) = \pi\}. \end{aligned}$$

Satz 2.3 Es sei $n > 0$. Dann gilt für alle $\pi \in \Pi_n$:

$$\pi \in L_k(\Pi_n) \iff \pi^{-1} \in L_k(\Pi_n).$$

Beweis: Es gelte $\pi \in L_k(\Pi_n)$. Dann gibt es ein einfaches normiertes induziertes zweistufig lösbares Stapelproblem $P = (G_A, G_Z, G_R, k)$ mit

$$\begin{aligned} G_A &= (\{1, \dots, n\}, \{(1, 2), \dots, (n-1, n)\}), \\ G_Z &= (\{1, \dots, n\}, \{(\pi(1), \pi(2)), \dots, (\pi(n-1), \pi(n))\}) \\ \text{und } G_R &= G_A \cup G_Z. \end{aligned}$$

Weil die Regeln zum Lösen eines einfachen induzierten zweistufig lösbaren Stapelproblems offensichtlich symmetrisch sind bezüglich Ausgangs- und Zielstapel, können wir diese Stapel vertauschen und erhalten ein neues, zweistufig lösbares Problem $P' := (G'_A, G'_Z, G_R, k)$ für das $G'_A = G_Z$ und $G'_Z = G_A$ gilt. Um dieses Problem zu normieren, benötigen wir die Funktion π^{-1} . Damit ergibt sich

$$G''_A := (\{1, \dots, n\}, \{(1, 2), \dots, (n-1, n)\}),$$

und $G''_Z := (\{1, \dots, n\}, \{(\pi^{-1}(1), \pi^{-1}(2)), \dots, (\pi^{-1}(n-1), \pi^{-1}(n))\}).$

Da P' zweistufig lösbar ist, ist es auch das normierte Problem P'' . Damit gilt $\pi^{-1} \in L_k(\Pi_n)$. Die Gültigkeit von

$$\pi^{-1} \in L_k(\Pi_n) \Rightarrow \pi \in L_k(\Pi_n)$$

läßt sich analog zeigen. \square

Zunächst noch einige einfache Fakten:

Da unser Regelgraph G_R nun nur aus den Kanten besteht, die in G_A und G_Z existieren, gilt für alle $u \in V(G_R)$:

$$d_{G_R}^+(u) \leq 2 \text{ und } d_{G_R}^-(u) \leq 2. \quad (2.5)$$

(Jeder Knoten hat sowohl im Ausgangs-, als auch im Zielgraphen höchstens einen nachfolgenden und höchstens einen vorhergehenden Knoten.) Insbesondere gilt:

$$d_{G_R}^+(n) \leq 1 \text{ und } d_{G_R}^+(\pi(n)) \leq 1. \quad (2.6)$$

weil n im Ausgangsgraphen und $\pi(n)$ im Zielgraphen keine nachfolgenden Knoten haben. Für $n = \pi(n)$ gilt gar

$$d_{G_R}^+(n) = d_{G_R}^+(\pi(n)) = 0. \quad (2.7)$$

Da 1 im Ausgangsstapel und $\pi(1)$ im Zielstapel ganz oben liegen, ist außerdem:

$$d_{G_R}^-(1) \leq 1 \text{ und } d_{G_R}^-(\pi(1)) \leq 1. \quad (2.8)$$

Lemma 2.4 Es sei $P = (G_A, G_Z, G_R, k)$ ein einfaches, normiertes und induziertes Stapelproblem mit $n := |V(P)|$. Dann gilt:

- (i) Existiert ein $x \in V(P)$ mit $(n, x) \in A(G_R)$, dann folgt $(n, x) \in A(G_Z)$ und $(n, x) \notin A(G_A)$.
- (ii) Existiert ein $x \in V(P)$ mit $(\pi(n), x) \in A(G_R)$, dann folgt $(\pi(n), x) \in A(G_A)$ und $(\pi(n), x) \notin A(G_Z)$.
- (iii) Existiert ein $x \in V(P)$ mit $(x, 1) \in A(G_R)$, dann folgt $(x, 1) \in A(G_Z)$ und $(x, 1) \notin A(G_A)$.

- (iv) Existiert ein $x \in V(P)$ mit $(x, \pi(1)) \in A(G_R)$, dann folgt $(x, \pi(1)) \in A(G_A)$ und $(x, \pi(1)) \notin A(G_Z)$.

Beweis: Die Aussagen (i) bis (iv) folgen aus der Struktur von G_R (2.4) und nach (2.5) bis (2.8). \square

Lemma 2.5 Es sei $P = (G_A, G_Z, G_R, k)$ ein einfaches zweistufig lösbares induziertes Stapelproblem. Es seien $x, y, z \in V(P)$ Stapелеlemente und es gelte $(x, y) \in A(G_A)$ (bzw. $(x, y) \in A(G_Z)$) und $(y, z) \in A(G_Z)$ (bzw. $(y, z) \in A(G_A)$). Dann gibt es keine Kante von z nach y in G_R .

Beweis: Angenommen, die Kante (z, y) existierte. Wegen $(y, z) \in A(G_Z)$ (bzw. $(y, z) \in A(G_A)$) folgte $(z, y) \in A(G_A)$ (bzw. $(z, y) \in A(G_Z)$). Das steht aber im Widerspruch zu $(x, y) \in A(G_A)$ (bzw. $(x, y) \in A(G_Z)$). Es können nicht sowohl x als auch z im Ausgangsstapel auf y liegen (P ist induziert!). Somit kann die Kante (z, y) nicht existieren. \square

Lemma 2.6 Es seien $P = (G_A, G_Z, G_R, k)$ ein einfaches, normiertes und induziertes zweistufig lösbares Stapelproblem und i, j, r natürliche Zahlen mit $0 < i \leq r < i + j < |V(P)|$. Weiterhin gelte:

$$\begin{aligned} & (i, i+1), (i+1, i+2), \dots, (i+j-1, i+j) \in A(G_R) \\ \text{und} \quad & (i+1, i), (i+2, i+1), \dots, (i+j, i+j-1) \in A(G_R). \end{aligned}$$

Für $(r, r+1) \in A(G_A)$ (bzw. $(r, r+1) \in A(G_Z)$) folgt dann:

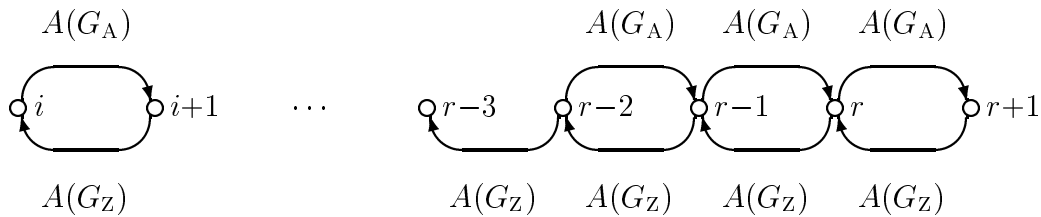
$$\begin{aligned} & (i, i+1), (i+1, i+2), \dots, (i+j-1, i+j) \in A(G_A) \text{ (bzw. } \in A(G_Z)) \\ \text{und} \quad & (i+1, i), (i+2, i+1), \dots, (i+j, i+j-1) \in A(G_Z) \text{ (bzw. } \in A(G_A)). \end{aligned}$$

Beweis: Wir betrachten $(r, r+1) \in A(G_A)$:

Da G_R nur Kanten der Ausgangs- und Zielgraphen enthält, folgt $(r+1, r) \in A(G_Z)$ und $(r, r-1) \in A(G_Z)$. (r kann im Ausgangsstapel nicht sowohl auf $r-1$ als auch auf $r+1$ liegen.)

Aus $(r, r-1) \in A(G_Z)$ folgt $(r-1, r) \in A(G_A)$ und $(r-2, r-1) \in A(G_A)$.

Aus $(r-2, r-1) \in A(G_A)$ folgt $(r-1, r-2) \in A(G_Z)$ und $(r-2, r-3) \in A(G_Z)$.



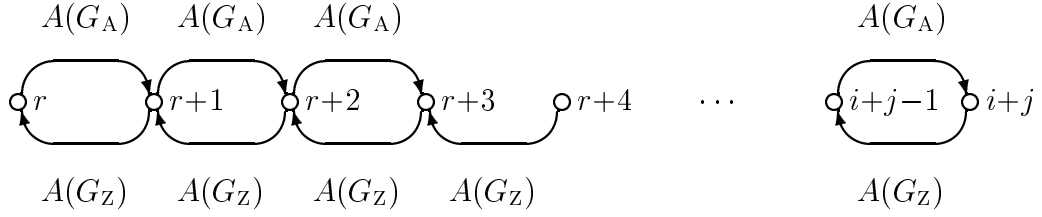
Setzen wir diese Schlußkette bis $i + 1$ und i fort, so erhalten wir:

$$(i, i + 1), (i + 1, i + 2), \dots, (r, r + 1) \in A(G_A)$$

und

$$(i + 1, i), (i + 2, i + 1), \dots, (r + 1, r) \in A(G_Z).$$

Mit $(r, r + 1) \in A(G_A)$ folgt weiterhin $(r + 2, r + 1) \in A(G_Z)$.
 Aus $(r + 2, r + 1) \in A(G_Z)$ folgt $(r + 1, r + 2) \in A(G_A)$ und $(r + 2, r + 3) \in A(G_A)$.
 Aus $(r + 2, r + 3) \in A(G_A)$ folgt $(r + 3, r + 2) \in A(G_Z)$ und $(r + 4, r + 3) \in A(G_Z)$.



Setzen wir diese Schlußkette bis $i + j - 1$ und $i + j$ fort, so erhalten wir:

$$(r + 1, r + 2), (r + 2, r + 3), \dots, (i + j - 1, i + j) \in A(G_A)$$

und

$$(r + 2, r + 1), (r + 3, r + 2), \dots, (i + j, i + j - 1) \in A(G_Z).$$

Für $(r, r + 1) \in A(G_Z)$ verläuft die Beweisführung analog. Damit ist das Lemma bewiesen. \square

Lemma 2.7 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, induziertes Stapelproblem. Weiter seien $e_1, e_2, e_3 \in V(P)$ Stapelelemente und $\mathcal{T} = [T_1, \dots, T_l]$ eine minimale zweistufige Umstapelfolge für P . Existiert ein $i \in \{2, \dots, l\}$ mit:

1. $T_{i-1} = (G_1^{i-1}, (\emptyset, \emptyset), G_3^{i-1}, G_4^{i-1})$ mit $G_3^{i-1} = (\{e_2\}, \emptyset)$,
2. $(e_1, e_2) \in A(G_Z)$, $(e_2, e_3) \in A(G_A)$ und $T_{i-1} \xrightarrow{[3,4]} T_i$ und
3. bildet e_1 die Spitze von G_1^{i-1} und das Element e_3 die Spitze von G_4^{i-1} ,

dann folgt $T_{i-2} \xrightarrow{[1,4]} T_{i-1}$.

Beweis: Weil \mathcal{T} zweistufig ist, der Ausgangsstapel nicht leer ist und wegen $k = 2$, kommen nur folgende 4 Möglichkeiten in Betracht:

- $T_{i-2} \xrightarrow{[4,3]} T_{i-1}$
- $T_{i-2} \xrightarrow{[1,3]} T_{i-1}$
- $T_{i-2} \xrightarrow{[3,4]} T_{i-1}$

$$\bullet T_{i-2} \xrightarrow{[1,4]} T_{i-1}$$

Gälte $T_{i-2} \xrightarrow{[4,3]} T_{i-1}$, so wäre $T_{i-2} = T_i$ und somit wäre $\mathcal{T} = [T_1, \dots, T_{i-2}, T_{i+1}, \dots, T_l]$ eine $l - 2$ -Umstapelfolge, was der Minimalität von \mathcal{T} widerspricht.

Angenommen, es wäre $T_{i-2} \xrightarrow{[1,3]} T_{i-1}$, so folgte $(e_2, e_1) \in A(G_A)$, was $(e_2, e_3) \in A(G_A)$ widerspricht.

Gälte $T_{i-2} \xrightarrow{[3,4]} T_{i-1}$, so wäre $(e_3, e_2) \in A(G_Z)$, was $(e_1, e_2) \in A(G_Z)$ widerspricht. Somit gilt $T_{i-2} \xrightarrow{[1,4]} T_{i-1} \cdot \square$

Definition 2.19 Wir führen abkürzende Bezeichnungen für häufig auftretende Stapel ein:

(i) Für $j < k$ sei

$$\begin{aligned} K_{v_j, v_k} &:= (\{v_j, \dots, v_k\}, \{(v_i, v_{i+1}) : i \in \{j, \dots, k-1\}\}), \\ K_{v_k, v_j} &:= (\{v_j, \dots, v_k\}, \{(v_i, v_{i-1}) : i \in \{j+1, \dots, k\}\}) \text{ und} \\ K_{v_j, v_j} &:= (\{v_j\}, \emptyset). \end{aligned}$$

(ii) Für $j < k - 2$ sei

$$\begin{aligned} W_{v_j, v_j} &:= (\{v_j\}, \emptyset), \\ W_{v_j, v_{j-1}} &:= (\{v_j, v_{j-1}\}, \{(v_j, v_{j-1})\}), \\ W_{v_j, v_{j-2}} &:= (\{v_j, v_{j-1}, v_{j-2}\}, \{(v_j, v_{j-1}), (v_{j-2}, v_j)\}) \text{ und} \\ W_{v_k, v_j} &:= W_{v_{k-2}, v_j} \cup (\{v_k, v_{k-1}, v_{k-3}\}, \{(v_k, v_{k-1}), (v_{k-3}, v_k)\}). \end{aligned}$$

(iii) Für $k + j$ ungerade und $j \leq k - 1$ sei

$$\begin{aligned} W_{v_j, v_k} &:= (\{v_j, \dots, v_k\}, \{(v_i, v_{i+1}) : i \in \{j, j+2, j+4, \dots, k-1\}\} \\ &\quad \cup \{(v_i, v_{i-3}) : i \in \{j+3, j+5, j+7, \dots, k\}\}). \end{aligned}$$

(iv) Für $k + j$ gerade und $j \leq k - 2$ sei

$$W_{v_j, v_k} := W_{v_j, v_{k-1}} \cup (\{v_k, v_{k-2}\}, \{(v_k, v_{k-2})\}).$$

Beispiele:

$$\begin{aligned} K_{a_{13}, a_{10}} &: \quad a_{10} \leftarrow a_{11} \leftarrow a_{12} \leftarrow a_{13} \quad , \\ W_{a_1, a_5} &: \quad a_2 \leftarrow a_1 \leftarrow a_4 \leftarrow a_3 \leftarrow a_5 \quad , \\ W_{a_5, a_1} &: \quad a_4 \leftarrow a_5 \leftarrow a_2 \leftarrow a_3 \leftarrow a_1 \quad . \end{aligned}$$

Lemma 2.8 Gilt für ein einfaches normiertes induziertes zweistufig lösbares Stapelproblem $P = (G_A, G_Z, G_R, 2)$

$$G_A = W_{a_i, a_k} \cup G_{\text{Rest}_A} \text{ und } G_Z = W_{a_{i+1}, a_k} \cup G_{\text{Rest}_Z}$$

mit $d_{G_A}^-(a_{k-1}) = d_{G_Z}^-(a_k) = 0$ für $i + k$ ungerade und $d_{G_A}^-(a_k) = d_{G_Z}^-(a_{k-1}) = 0$ für $i + k$ geradzahlig, so folgt $\pi(P) = \pi_{P(i,k), k-i} \sqcup \pi_{\text{Rest}}$.

Beweis: Nach Voraussetzung folgt:

$$\begin{array}{l}
 T_1 : \quad A : G_{\text{Rest}_A} a_{i+1} a_i \cdots a_k a_{k-1} \\
 \quad \quad Z : \\
 \quad \quad H_1 : \\
 \quad \quad H_2 :
 \end{array}
 \quad \text{und } T_l : \quad
 \begin{array}{l}
 A : \\
 Z : G_{\text{Rest}_Z} a_{i+2} a_{i+1} \cdots a_{k-2} a_k \\
 H_1 : \\
 H_2 :
 \end{array}$$

für $i + k$ ungerade und

$$\begin{array}{l}
 T_1 : \quad A : G_{\text{Rest}_A} a_{i+1} a_i \cdots a_{k-2} a_k \\
 \quad \quad Z : \\
 \quad \quad H_1 : \\
 \quad \quad H_2 :
 \end{array}
 \quad \text{und } T_l : \quad
 \begin{array}{l}
 A : \\
 Z : G_{\text{Rest}_Z} a_{i+2} a_{i+1} \cdots a_k a_{k-1} \\
 H_1 : \\
 H_2 :
 \end{array}$$

für $i + k$ geradzahlig. Da P normiert ist, gilt für $i + k$ ungerade

$$\pi(P) = [2, 4, 1, 6, 3, \dots, k-4, k-7, k-2, k-5, k, k-3] \sqcup \pi_{\text{Rest}}$$

und für $i + k$ geradzahlig

$$\pi(P) = [3, 1, 5, 2, 7, \dots, k-4, k-7, k-2, k-5, k, k-3] \sqcup \pi_{\text{Rest}},$$

wobei π_{Rest} $n - k + i$ Elemente enthält. \square

Lemma 2.9 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, induziertes Stapelproblem. $T_r = (G_1^r, G_2^r, G_3^r, G_4^r)$ und $T_s = (G_1^s, G_2^s, G_3^s, G_4^s)$ seien Stapelkonfigurationen einer minimalen zweistufigen Stapelfolge $\mathcal{T} = [T_1, \dots, T_l]$. Es gelte für $1 < j < k$:

$$\begin{aligned}
 G_1^r &= G_{\text{Rest}_A} \text{ wobei } a \text{ auf der Spitze von } G_{\text{Rest}_A} \text{ liegt,} \\
 G_2^r &= (\emptyset, \emptyset), \\
 G_3^r &= (\emptyset, \emptyset), \\
 G_4^r &= K_{a_j, a_k}, \\
 G_1^s &= (\emptyset, \emptyset), \\
 G_2^s &= G_{\text{Rest}_Z} \text{ wobei } b \text{ auf der Spitze von } G_{\text{Rest}_Z} \text{ liegt} \\
 G_3^s &= (\emptyset, \emptyset) \text{ und} \\
 G_4^s &= K_{a_{j+1}, a_k},
 \end{aligned}$$

sowie existiere ein $x \in V(P)$ mit $x \neq a_{j+1}, a$ und $(a_j, x) \in A(G_Z)$. Weiterhin existiere ein $y \in V(P)$ mit $y \neq a_{j+1}$ und $(y, a_j) \in A(G_Z)$. Dann folgt:

1. $T_1 = [G_A, (\emptyset, \emptyset), (\emptyset, \emptyset), (\emptyset, \emptyset)]$ und $T_l = [(\emptyset, \emptyset), G_Z, (\emptyset, \emptyset), (\emptyset, \emptyset)]$ mit

$$G_A = G_{\text{Rest}_A} \cup W_{a_j, a_k} \cup (\{a_{j+1}, a\}, \{(a_{j+1}, a)\}),$$

$$G_Z = G_{\text{Rest}_Z} \cup W_{a_{j+1}, a_k} \cup (\{a_{j+2}, b\}, \{(a_{j+2}, b)\}),$$

wobei a auf der Spitze von G_{Rest_A} und b auf der Spitze von G_{Rest_Z} liegen.

2. Für r und s gilt:

$$r = \begin{cases} \frac{3}{2}(k-j+1) + 1 & \text{für } k+j \text{ ungerade} \\ \frac{3}{2}(k-j) + 2 & \text{für } k+j \text{ gerade} \end{cases} \quad \text{und}$$

$$s = \begin{cases} l - \frac{3}{2}(k-j-1) - 1 & \text{für } k+j \text{ ungerade} \\ l - \frac{3}{2}(k-j) & \text{für } k+j \text{ gerade} \end{cases}.$$

Beweis: Zunächst wollen wir versuchen, durch eine grafische Darstellung der Stapel die Situation zu erläutern:

$$T_r : \begin{array}{l} A : \cdots a \\ Z : \\ H_1 : \\ H_2 : a_k \cdots a_j \end{array}, \quad T_s : \begin{array}{l} A : \\ Z : \cdots b \\ H_1 : \\ H_2 : a_k \cdots a_{j+1} \end{array}.$$

Mit A und Z bezeichnen wir Ausgangs- und Zielstapel, mit H_1 und H_2 die beiden Hilfsstapel. Die Spitze der Stapel befindet sich in unseren Darstellungen auf der rechten Seite.

Wir zeigen nun zunächst Punkt (1) des Lemma. Dieser sagt aus, daß sich aus der Kenntnis über die Gestalt von T_r und T_s beweisen läßt, daß T_1 und T_l wie folgt aussehen:

Für $k+j$ ungerade:

$$T_1 : \begin{array}{l} A : G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} a_{j+2} \cdots a_k a_{k-1} \\ Z : \\ H_1 : \\ H_2 : \end{array},$$

$$T_l : \begin{array}{l} A : \\ Z : G_{\text{Rest}_Z} a_{j+2} a_{j+1} a_{j+4} a_{j+3} \cdots a_{k-1} a_{k-2} a_k \\ H_1 : \\ H_2 : \end{array}.$$

Für $k + j$ geradzahlig:

$$T_1 : \begin{array}{l} A : G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} a_{j+2} \cdots a_{k-1} a_{k-2} a_k \\ Z : \\ H_1 : \\ H_2 : \end{array} ,$$

$$T_l : \begin{array}{l} A : \\ Z : G_{\text{Rest}_Z} a_{j+2} a_{j+1} a_{j+4} a_{j+3} \cdots a_k a_{k-1} \\ H_1 : \\ H_2 : \end{array} .$$

Ausgehend von den Stapelkonfigurationen T_r und T_s werden wir nun untersuchen, zwischen welchen Stapелеlementen welche Kantenbeziehungen existieren: In G_4^r liegt a_j auf a_{j+1} , für $(a_j, x) \in A(G_Z)$ folgt somit $(a_j, a_{j+1}) \in A(G_A)$.

Für T_{r-1} bedeutet das, daß a_j nicht auf a liegen kann. Somit folgt:

$$T_{r-1} = (G_1^r, (\emptyset, \emptyset), (\{a_j\}, \emptyset), K_{a_{j+1}, a_k}),$$

$$T_{r-1} : \begin{array}{l} A : \cdots a \\ Z : \\ H_1 : a_j \\ H_2 : a_k \cdots a_{j+1} \end{array} .$$

Nach Lemma 2.5 (für y, a_j, a_{j+1}) existiert die Kante (a_{j+1}, a_j) nicht. Wegen $(a_j, a_{j+1}) \in A(G_A)$ erhalten wir damit sofort:

$$\begin{aligned} T_{r-3} &= (G_1^{r-3}, (\emptyset, \emptyset), (\emptyset, \emptyset), K_{a_{j+2}, a_k}) \text{ mit} \\ G_1^{r-3} &= G_1^r \cup (\{a_j, a_{j+1}, a\}, \{(a_j, a_{j+1}), (a_{j+1}, a)\}), \end{aligned}$$

$$T_{r-3} : \begin{array}{l} A : G_{\text{Rest}_A} a_{j+1} a_j \\ Z : \\ H_1 : \\ H_2 : a_k \cdots a_{j+2} \end{array} .$$

Es ist also $(a_{j+1}, a) \in A(G_A)$. Weil aber auch die Kante (a_{j+1}, a_{j+2}) existiert, folgt $(a_{j+1}, a_{j+2}) \in A(G_Z)$.

Wir wenden uns nun der Konfiguration T_s zu und haben sofort:

$$T_{s+1} = ((\emptyset, \emptyset), G_2^s, (\{a_{j+1}\}, \emptyset), K_{a_{j+2}, a_k}),$$

$$\begin{array}{l}
A : \\
T_{s+1} : \quad Z : G_{\text{Rest}_Z} \\
\quad \quad H_1 : a_{j+1} \\
\quad \quad H_2 : a_k \cdots a_{j+2}
\end{array}
.$$

Nach Lemma 2.5 (für a_j, a_{j+1}, a_{j+2}) existiert die Kante (a_{j+2}, a_{j+1}) nicht. Wegen $(a_{j+1}, a_{j+2}) \in A(G_Z)$ erhalten wir damit sofort:

$$\begin{aligned}
T_{s+3} &= ((\emptyset, \emptyset), G_2^{s+3}, (\emptyset, \emptyset), K_{a_{j+3}, a_k}) \text{ mit} \\
G_2^{s+3} &= G_2^s \cup (\{a_{j+1}, a_{j+2}, b\}, \{(a_{j+1}, a_{j+2}), (a_{j+2}, b)\}),
\end{aligned}$$

$$\begin{array}{l}
A : \\
T_{s+3} : \quad Z : G_{\text{Rest}_Z} a_{j+2} a_{j+1} \\
\quad \quad H_1 : \\
\quad \quad H_2 : a_k \cdots a_{j+3}
\end{array}
.$$

Es ist also $(a_{j+2}, b) \in A(G_Z)$. Weil aber auch die Kante (a_{j+2}, a_{j+3}) existiert, folgt $(a_{j+2}, a_{j+3}) \in A(G_A)$.

Betrachten wir jetzt wieder T_{r-3} , so folgt sofort:

$$T_{r-4} = (G_1^{r-3}, (\emptyset, \emptyset), (\{a_{j+2}\}, \emptyset), K_{a_{j+3}, a_k}),$$

$$\begin{array}{l}
A : G_{\text{Rest}_A} a_{j+1} a_j \\
T_{r-4} : \quad Z : \\
\quad \quad H_1 : a_{j+2} \\
\quad \quad H_2 : a_k \cdots a_{j+3}
\end{array}
.$$

Nach Lemma 2.5 (für $a_{j+1}, a_{j+2}, a_{j+3}$) existiert die Kante (a_{j+3}, a_{j+2}) nicht. Wegen $(a_{j+2}, a_{j+3}) \in A(G_A)$ folgt sofort:

$$\begin{aligned}
T_{r-6} &= (G_1^{r-6}, (\emptyset, \emptyset), (\emptyset, \emptyset), K_{a_{j+4}, a_k}) \text{ mit} \\
G_1^{r-6} &= G_1^{r-3} \cup (\{a_{j+2}, a_{j+3}\}, \{(a_{j+2}, a_{j+3}), (a_{j+3}, a_j)\}),
\end{aligned}$$

$$\begin{array}{l}
A : G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} a_{j+2} \\
T_{r-6} : \quad Z : \\
\quad \quad H_1 : \\
\quad \quad H_2 : a_k \cdots a_{j+4}
\end{array}
.$$

Somit ist $(a_{j+3}, a_j) \in A(G_A)$. Da die Kante (a_{j+3}, a_{j+4}) existiert, ist damit $(a_{j+3}, a_{j+4}) \in A(G_Z)$.

Setzen wir diese Argumentation fort, so erhalten wir für $k + j$ geradzahlig, T_u, T_v mit:

$$\begin{array}{l}
T_u : \quad \begin{array}{l}
\text{A : } G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} a_{j+2} \cdots a_{k-3} a_{k-4} a_{k-1} a_{k-2} \\
\text{Z :} \\
\text{H}_1 : \\
\text{H}_2 : a_k
\end{array} \\
\text{und} \\
T_v : \quad \begin{array}{l}
\text{A :} \\
\text{Z : } G_{\text{Rest}_Z} a_{j+2} a_{j+1} a_{j+4} a_{j+3} \cdots a_{k-2} a_{k-3} \\
\text{H}_1 : \\
\text{H}_2 : a_k a_{k-1}
\end{array}
\end{array}$$

Es ist also $(a_{k-2}, a_{k-1}) \in A(G_A)$ und wegen $(a_{k-1}, a_{k-4}) \in A(G_A)$ folgt $(a_{k-1}, a_k) \in A(G_Z)$. Nach Lemma 2.5 existiert damit die Kante (a_k, a_{k-1}) nicht und es folgt:

$$\begin{array}{l}
T_1 = T_{u-1} : \quad \begin{array}{l}
\text{A : } G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} \cdots a_{k-3} a_{k-4} a_{k-1} a_{k-2} a_k \\
\text{Z :} \\
\text{H}_1 : \\
\text{H}_2 :
\end{array} \\
\text{und} \\
T_l = T_{v+2} : \quad \begin{array}{l}
\text{A :} \\
\text{Z : } G_{\text{Rest}_Z} a_{j+2} a_{j+1} a_{j+4} a_{j+3} \cdots a_{k-2} a_{k-3} a_k a_{k-1} \\
\text{H}_1 : \\
\text{H}_2 :
\end{array}
\end{array}$$

Für $k + j$ ungerade erhalten wir T_u, T_v mit:

$$\begin{array}{l}
T_u : \quad \begin{array}{l}
\text{A : } G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} a_{j+2} \cdots a_{k-4} a_{k-5} a_{k-2} a_{k-3} \\
\text{Z :} \\
\text{H}_1 : \\
\text{H}_2 : a_k a_{k-1}
\end{array} \\
\text{und} \\
T_v : \quad \begin{array}{l}
\text{A :} \\
\text{Z : } G_{\text{Rest}_Z} a_{j+2} a_{j+1} a_{j+4} a_{j+3} \cdots a_{k-4} a_{k-1} a_{k-2} \\
\text{H}_1 : \\
\text{H}_2 : a_k
\end{array}
\end{array}$$

Es ist also $(a_{k-2}, a_{k-1}) \in A(G_Z)$ und wegen $(a_{k-1}, a_{k-4}) \in A(G_Z)$ folgt $(a_{k-1}, a_k) \in A(G_A)$. Nach Lemma 2.5 existiert damit die Kante (a_k, a_{k-1}) nicht

und es folgt:

$$\begin{array}{l}
T_1 = T_{u-2} : \quad \begin{array}{l}
A : G_{\text{Rest}_A} a_{j+1} a_j a_{j+3} \cdots a_{k-5} a_{k-2} a_{k-3} a_k a_{k-1} \\
Z : \\
H_1 : \\
H_2 :
\end{array} \\
\text{und} \\
T_l = T_{v+2} : \quad \begin{array}{l}
A : \\
Z : G_{\text{Rest}_Z} a_{j+2} a_{j+1} a_{j+4} a_{j+3} \cdots a_{k-4} a_{k-1} a_{k-2} a_k \\
H_1 : \\
H_2 :
\end{array}
\end{array}$$

Damit ist $T_1 = [G_1^1, (\emptyset, \emptyset), (\emptyset, \emptyset), (\emptyset, \emptyset)]$ und $T_l = [(\emptyset, \emptyset), G_2^l, (\emptyset, \emptyset), (\emptyset, \emptyset)]$ mit

$$\begin{array}{l}
G_1^1 = G_{\text{Rest}_A} \cup W_{a_j, a_k} \cup (\{a_{j+1}, a\}, \{(a_{j+1}, a)\}) \\
\text{und} \\
G_2^l = G_{\text{Rest}_Z} \cup W_{a_{j+1}, a_k} \cup (\{a_{j+2}, b\}, \{(a_{j+2}, b)\}).
\end{array}$$

Damit ist Punkt (1) des Lemmas bewiesen.

Nun zu Punkt (2): Es sei $k+j$ ungerade: Um von T_1 zu T_r zu gelangen, muß ein Stapel der Form W_{a_j, a_k} ausgestapelt werden und auf einem Hilfsstapel als Stapel der Form K_{a_j, a_k} auftreten. Die Stapelelemente $a_{j+1}, a_{j+3}, \dots, a_k$ können jeweils sofort vom Ausgangsstapel auf diesen Hilfsstapel gelegt werden – für jedes dieser Elemente ist also nur eine Umstapelung nötig. Die Elemente $a_j, a_{j+2}, \dots, a_{k-1}$ müssen vorher jedoch auf den anderen Hilfsstapelplatz gelegt werden, damit muß jedes dieser Elemente 2 mal bewegt werden. Da es sich um $k-j+1$ Elemente handelt, müssen also $\frac{3}{2}(k-j+1)$ Umstapelungen vorgenommen werden. somit ist $r = \frac{3}{2}(k-j+1) + 1$.

Nun sei $k+j$ gerade: Da es sich nun um eine ungerade Anzahl von Elementen handelt und sowohl a_k als auch a_{j+1} mit nur einer Umstapelung an die richtige Stelle zu setzen sind, müssen $\frac{3}{2}(k-j) + 1$ Umstapelungen vorgenommen werden. In diesem Fall ist also $r = \frac{3}{2}(k-j) + 2$.

Für T_s erfolgt die Beweisführung analog. Somit ist Punkt (2) und damit das Lemma bewiesen. \square

Es geht nun im folgenden darum, die Menge $L_2(\Pi_n)$ zu bestimmen, das sind diejenigen Permutationen π , für die $P = (G_A, G_Z, G_R, 2)$ mit $\pi(P) = \pi$ und $|V(P)| = n$ zweistufig lösbar ist.

Wenn ein Stapelproblem P zweistufig lösbar sein soll, muß eine minimale Umstapelfolge $\mathcal{T} = [T_1, \dots, T_l]$ und ein kleinstes $j \in \{1, \dots, l\}$ existieren, für das

$T_j = ((\emptyset, \emptyset), (\emptyset, \emptyset), G_3^j, G_4^j)$ gilt. Die Stapelkonfiguration T_j repräsentiert folgende Situation: Alle Elemente wurden vom Ausgangsstapel auf die Hilfsstapel gebracht und noch kein Element wurde auf den Zielstapel gelegt. Diese Konfiguration T_j machen wir nun zum Ausgangspunkt unserer weiteren Betrachtungen. Weil Probleme mit weniger als 5 Stapelelementen sich sehr leicht lösen lassen (für $n = 0, 1, 2, 3$ gilt gar $L_2(\Pi_n) = \Pi_n$), betrachten wir von nun an nur noch Probleme P mit mehr als 4 Stapelelementen: $|V(P)| > 4$. Für $n = 4$ lassen sich die 13 Permutationen, für die das Problem lösbar ist, sehr leicht angeben.

Wir unterscheiden nun zwei Fälle: Erstens $n = \pi(n)$ und zweitens $n \neq \pi(n)$. Im Fall $n = \pi(n)$ sieht T_j o.B.d.A. wie folgt aus:

$$T_j : \begin{array}{l} A : \\ Z : \\ H_1 : \pi(n) \\ H_2 : b_{n-1} \cdots b_1 \end{array}, \quad (2.9)$$

da $n = \pi(n)$ sowohl im Ausgangsstapel als auch im Zielstapel keinen Nachfolger hat und somit im Regelgraphen ebenfalls kein Nachfolger existiert.

2.3.2 Struktur zweistufig lösbarer Stapelprobleme mit $n = \pi(n)$

Lemma 2.10 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$ und $V(P) = \{b_1, \dots, b_{n-1}, n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Weiterhin seien b_r und b_s diejenigen Stapelelemente aus $\{b_1, \dots, b_{n-1}\}$ für die $(b_r, n) \in A(G_A)$ und $(b_s, \pi(n)) \in A(G_Z)$ gilt. Für $n = \pi(n)$ gilt dann mit der Struktur von T_j (2.9):

1. Aus $r = s$ folgt $r = s = n - 1$ und
2. aus $r \neq s$ folgt $r = 1$ oder $s = 1$.

Beweis:

1. Angenommen, (1) gälte nicht, dann wäre $r = s$ und $r, s < n - 1$. Auf dem Hilfsstapel in T_j läge dann $b_r = b_s$ auf $b_{r+1} = b_{s+1}$ und somit wäre $(b_r = b_s, b_{r+1} = b_{s+1}) \in A(G_R)$. Mit $(b_r = b_s, n) \in A(G_A)$ liegt $b_r = b_s$ im Ausgangsstapel bereits auf n . Im Zielstapel liegt $b_r = b_s$ aber wegen $(b_s = b_r, \pi(n)) \in A(G_Z)$ auf $\pi(n)$. Somit kann die Kante $(b_r = b_s, b_{r+1} = b_{s+1})$ nicht existieren. Somit ist die Annahme falsch und es gilt (1).

2. Angenommen, (2) gälte nicht, dann wäre $r \neq s$ und $r, s > 1$.

Wir betrachten den Fall $1 < r < s \leq n - 1$. Aus den Voraussetzungen wissen wir bereits, daß

$$(b_r, b_{r+1}) \in A(G_R) \text{ und } (b_r, n = \pi(n)) \in A(G_R) \text{ gilt.} \quad (2.10)$$

Um nun, von T_j ausgehend, auf den Zielstapel das Element b_s auf $\pi(n)$ legen zu können, müssen zuvor die Elemente b_1 bis b_{s-1} auf den ersten Hilfsstapelplatz gestapelt werden. Dort liegt dann insbesondere b_r auf b_{r-1} , somit ist $(b_r, b_{r-1}) \in A(G_R)$ – mit (2.10) widerspricht das jedoch (2.5).

Die Beweisführung für den Fall $1 < s < r \leq n - 1$ erfolgt analog. Damit ist die Annahme falsch, das Lemma damit bewiesen. \square

Lemma 2.11 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches zweistufig lösbares, normiertes und induziertes Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$ und $V(P) = \{b_1, \dots, b_{n-1}, n = \pi(n)\}$. Weiterhin seien $\mathcal{T} := [T_1, \dots, T_l]$ eine minimale zweistufige Umstapelfolge und j der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt.

Für $n = \pi(n)$ folgt dann

- (i) $\pi = [n - 2, n - 3, \dots, 2, 1, n - 1, n]$ oder
- (ii) $\pi = \pi_{P(n-1,s), n-s-1} \sqcup [n - 1, n - 2, \dots, n - s + 1, n - s - 1, n]$
für $s \in \{2, \dots, n - 2\}$ oder
- (iii) $\pi = \hat{\pi}^{-1}$, mit $\hat{\pi} = \pi_{P(n-1,s), n-s-1} \sqcup [n - 1, n - 2, \dots, n - s + 1, n - s - 1, n]$
für $s \in \{2, \dots, n - 2\}$ oder
- (iv) $\pi = [n - 1, n - 2, n - 3, \dots, 2, 1, n]$.

Für den Fall (i) ist $l = 5n - 6$, für die Fälle (ii) bis (iv) ist $l = 3n - 1$.

Beweis: Es sei $V(P) = \{b_1, \dots, b_{n-1}, n\}$ und b_r und b_s seien diejenigen Stapelelemente aus $\{b_1, \dots, b_{n-1}\}$ für die $(b_r, n = \pi(n)) \in A(G_A)$ und $(b_s, \pi(n) = n) \in A(G_Z)$ gilt. Nach Lemma 2.10 sind nur die drei Fälle

- 1. $r = s = n - 1$,
- 2. $r \neq s$, $r = 1$ und
- 3. $r \neq s$, $s = 1$

zu untersuchen.

1. Um, von T_j ausgehend, $b_s = b_{n-1}$ auf $\pi(n)$ auf den Zielstapel legen zu können, müssen zunächst die Elemente b_1, \dots, b_{n-2} auf den zweiten Hilfsstapel gestapelt werden. Somit existieren sowohl die Kanten $(b_1, b_2), (b_2, b_3), \dots, (b_{n-2}, b_{n-1}) \in A(G_R)$ als auch die Kanten $(b_{n-2}, b_{n-3}), \dots, (b_3, b_2), (b_2, b_1)$. Mit Lemma 2.6 (für i und $i+j$ stehen hier die Elemente b_1 und b_{n-2} , für r steht b_1) gilt für $(b_1, b_2) \in A(G_A)$:

$$\begin{aligned} G_A &= (\{b_1, \dots, b_{n-1}, n\}, \{(b_1, b_2), \dots, (b_{n-2}, b_{n-1}), (b_{n-1}, n)\}) \\ G_Z &= (\{b_1, \dots, b_{n-1}, n\}, \{(b_{n-2}, b_{n-3}), \dots, (b_2, b_1), (b_1, b_{n-1}), (b_{n-1}, n)\}) \end{aligned}$$

und für $(b_1, b_2) \in A(G_Z)$:

$$\begin{aligned} G_A &= (\{b_1, \dots, b_{n-1}, n\}, \{(b_{n-2}, b_{n-3}), \dots, (b_2, b_1), (b_1, b_{n-1}), (b_{n-1}, n)\}) \\ G_Z &= (\{b_1, \dots, b_{n-1}, n\}, \{(b_1, b_2), \dots, (b_{n-2}, b_{n-1}), (b_{n-1}, n)\}). \end{aligned}$$

Weil P nach Voraussetzung normiert ist, erhalten wir in beiden Fällen (sowohl $(b_1, b_2) \in A(G_A)$ als auch $(b_1, b_2) \in A(G_Z)$):

$$\begin{aligned} G_A &= (\{1, \dots, n\}, \{(1, 2), \dots, (n-1, n)\}) \\ G_Z &= (\{1, \dots, n\}, \{(n-2, n-3), \dots, (2, 1), (1, n-1), (n-1, n)\}) \end{aligned}$$

Für den Fall $r = s = n - 1$ folgt damit

$$\pi = [n-2, n-3, \dots, 2, 1, n-1, n].$$

Es ist leicht nachzuvollziehen, daß $l = 5n - 6$ gilt.

2. Wenn wir, von T_j ausgehend, b_s auf $\pi(n) = n$ auf den Zielstapel legen wollen, so müssen wir zunächst die Stapелеlemente b_1 bis b_{s-1} auf den zweiten Hilfsstapel legen. Daraus folgt die Existenz der Kanten $(b_{s-1}, b_{s-2}), (b_{s-2}, b_{s-3}), \dots, (b_2, b_1) \in A(G_R)$, aus der Voraussetzung wissen wir, daß $(b_1, b_2), (b_2, b_3), \dots, (b_{s-2}, b_{s-1}) \in A(G_R)$ gilt. Wegen $r = 1$ gilt $(b_1, n) \in A(G_A)$, damit folgt $(b_1, b_2) \in A(G_Z)$. Mit Lemma 2.6 (für i und $i+j$ stehen hier b_1 und b_{s-1} , für r steht b_1) folgt:

$$\begin{aligned} &(b_1, b_2), (b_2, b_3), \dots, (b_{s-1}, b_s) \in A(G_Z) \\ \text{und} &(b_{s-1}, b_{s-2}), (b_{s-2}, b_{s-3}), \dots, (b_2, b_1) \in A(G_A). \end{aligned}$$

Damit existieren die folgenden Stapelkonfigurationen T_{j-s} und T_{j+2s} :

$$\begin{array}{ccc} & A : n b_1 \cdots b_{s-1} & A : \\ T_{j-s} : & Z : & Z : \pi(n) b_s \cdots b_1 \\ & H_1 : & H_1 : \\ & H_2 : b_{n-1} \cdots b_s & H_2 : b_{n-1} \cdots b_{s+1} \end{array}, \quad T_{j+2s} :$$

Nach Lemma 2.9 folgt nun für $s \in \{2, \dots, n-2\}$:

$$G_A = K_{b_{s-1}, b_1} \cup W_{b_s, b_{n-1}} \cup (\{b_{s+1}, b_{s-1}, b_1, n\}, \{(b_{s+1}, b_{s-1}), (b_1, n)\})$$

und

$$G_Z = K_{b_1, b_s} \cup W_{b_{s+1}, b_{n-1}} \cup (\{b_{s+2}, b_1, b_s, \pi(n)\}, \{(b_{s+2}, b_1), (b_s, \pi(n))\})$$

und für $n+s$ geradzahlig:

$$j-s = \frac{3}{2}(n-s) + 1, \quad j+2s = l - \frac{3}{2}(n-s-2) - 1$$

und für $n+s$ ungerade:

$$j-s = \frac{3}{2}(n-s-1) + 2, \quad j+2s = l - \frac{3}{2}(n-s-1).$$

Da P ein normiertes Problem ist, ergibt sich nach Lemma 2.8 für $r=1$, $r \neq s$ und $s \in \{2, \dots, n-2\}$ sofort:

$$\pi = \pi_{P(n-1, s), n-s-1} \sqcup [n-1, n-2, \dots, n-s+1, n-s-1, n].$$

Sowohl für den Fall $n+s-1$ ungerade, als auch für $n+s-1$ geradzahlig erhalten wir für $s \in \{2, \dots, n-2\}$ $l = 3n-1$.

Im Fall $s = n-1$ sehen T_{j-s} und T_{j+2s} wie folgt aus:

$$T_{j-s} : \begin{array}{l} A : n b_1 \cdots b_{s-1} \\ Z : \\ H_1 : \\ H_2 : b_{n-1} \end{array}, \quad T_{j+2s} : \begin{array}{l} A : \\ Z : \pi(n) b_{n-1} \cdots b_1 \\ H_1 : \\ H_2 : \end{array}$$

Somit ist für $s = n-1$

$$G_A = K_{b_s, b_1} \cup (\{b_1, n\}, \{(b_1, n)\})$$

und $G_Z = K_{b_1, b_s} \cup (\{b_s, \pi(n)\}, \{(b_s, \pi(n))\})$

und $j-s-1 = 1$ und $j+2s = l$ und damit $l = 3n-1$. Weil P normiert ist gilt:

$$\pi = [n-1, n-2, \dots, 2, 1, n].$$

3. Wenn wir nun den Fall $s = 1$ mit $r \in \{2, \dots, n-1\}$ betrachten, so erhalten wir diejenigen Stapel, wie im Fall (2), nur daß nun die Stapelplätze vertauscht sind: Die Stapel, die sich im Fall (2) auf dem Ausgangsstapel

ergeben haben, stehen nun auf dem Zielstapel und umgekehrt. Wir vertauschen also G_A und G_Z und erhalten somit für $r \in \{2, \dots, n-2\}$:

$$G_A = K_{b_1, b_r} \cup W_{b_{r+1}, b_{n-1}} \cup (\{b_{r+2}, b_1, b_r, \pi(n)\}, \{(b_{r+2}, b_1), (b_r, \pi(n))\})$$

und

$$G_Z = K_{b_{r-1}, b_1} \cup W_{b_r, b_{n-1}} \cup (\{b_{r+1}, b_{r-1}, b_1, n\}, \{(b_{r+1}, b_{r-1}), (b_1, n)\})$$

und für $r = n-1$:

$$G_A = K_{b_1, b_r} \cup (\{b_r, \pi(n)\}, \{(b_r, \pi(n))\})$$

und

$$G_Z = K_{b_r, b_1} \cup (\{b_1, n\}, \{(b_1, n)\}).$$

Weil P normiert ist, gilt für $r \in \{2, \dots, n-2\}$:

$$\pi = \hat{\pi}^{-1}$$

$$\text{mit } \hat{\pi} = \pi_{P(n-1, r), n-r-1} \sqcup [n-1, n-2, \dots, n-r+1, n-r-1, n]$$

und für $r = n-1$:

$$\pi = [n-1, n-2, \dots, 2, 1, n]^{-1} = [n-1, n-2, \dots, 2, 1, n].$$

Damit haben wir auch Punkt (3) gezeigt – die Gültigkeit des Lemmas ist bewiesen. \square

2.3.3 Fallunterscheidungen für $n \neq \pi(n)$

Im Abschnitt 2.3.2 behandelten wir den Fall $\pi(n) = n$. Wenden wir uns nun dem Fall $\pi(n) \neq n$ zu, so kommt eine Fülle von Fallunterscheidungen auf uns zu.

Zunächst: Wie sieht die typische Konfiguration T_j (j minimal) aus, in der alle Elemente des Ausgangsstapels auf die Hilfsstapel gelegt wurden und sich noch kein Element auf dem Zielstapel befindet?

Lemma 2.12 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches induziertes Stapelproblem mit $n := |V(P)| > 4$. Weiterhin seien $\mathcal{T} := [T_1, \dots, T_l]$ eine minimale zweistufige Umstapelfolge und $\pi := \pi(P)$ mit $n \neq \pi(n)$.

Dann gilt o.B.d.A. für T_j :

$$G_3^j = K_{a_1, a_r} \cup (\{n, a_1\}, \{(n, a_1)\})$$

$$G_4^j = K_{x_1, x_t} \cup K_{b_1, b_s} \cup (\{x_t, \pi(n), b_1\}, \{(x_t, \pi(n)), (\pi(n), b_1)\})$$

mit $r, s, t \geq 0$ und $r + s + t = n - 2 > 2$:

$$T_j : \begin{array}{l} \text{A :} \\ \text{Z :} \\ \text{H}_1 : a_r \cdots a_1 n \\ \text{H}_2 : b_s \cdots b_1 \pi(n) x_t \cdots x_1 \end{array} .$$

Beweis: Wir haben folgendes zu zeigen:

1. n liegt auf der Spitze eines Hilfsstapels,
 2. n und $\pi(n)$ liegen in verschiedenen Hilfsstapeln.
1. Weil j der kleinste Index sein soll, für denn in T_j keine Elemente mehr auf dem Ausgangsstapel liegen und noch kein Element auf dem Zielstapel liegt, folgt, daß n auf der Spitze eines Hilfsstapels (o.B.d.A. von H_1) liegen muß.
 2. Angenommen, n und $\pi(n)$ lägen auf *einem* Hilfsstapel – o.B.d.A. auf H_2 , dann sähe T_j wie folgt aus:

$$\begin{aligned} G_3^j &= K_{a_1, a_r} \text{ und} \\ G_4^j &= K_{x_1, x_t} \cup K_{b_1, b_s} \\ &\quad \cup (\{n, x_1, x_t, \pi(n), b_1\}, \{(n, x_1), (x_t, \pi(n)), (\pi(n), b_1)\}), \end{aligned}$$

$$\begin{array}{l} T_j : \\ \text{A :} \\ \text{Z :} \\ \text{H}_1 : a_r \cdots a_1 \\ \text{H}_2 : b_s \cdots b_1 \pi(n) x_t \cdots x_1 n \end{array} .$$

Um nun, von T_j ausgehend, T_l zu erreichen (um insbesondere $\pi(n)$ auf den Zielstapel legen zu können), müssen die Stapелеlemente n, x_1, \dots, x_t auf a_1 umgestapelt werden. Somit folgt $(n, a_1) \in A(G_R)$. Wegen $(n, x_1) \in A(G_R)$ folgt wegen (2.6) $r = 0$. Für $r = 0$ ergibt sich jedoch in T_{j+1} :

$$\begin{array}{l} T_{j+1} : \\ \text{A :} \\ \text{Z :} \\ \text{H}_1 : n \\ \text{H}_2 : b_s \cdots b_1 \pi(n) x_t \cdots x_1 \end{array} .$$

Somit ist $\mathcal{T}' := [T_1, \dots, T_{j-1}, T_{j+1}, \dots, T_l]$ eine kürzere Umstapelfolge als \mathcal{T} , das widerspricht jedoch der Voraussetzung, daß \mathcal{T} minimal ist, somit ist die Annahme falsch und das Lemma bewiesen. \square

2.3.4 Übersicht der zu untersuchenden Fälle

Um im folgenden Text nicht den Durchblick zu verlieren, geben wir zunächst alle Fälle an, die zu untersuchen sind, um $L_2(\Pi_n)$ zu bestimmen.

1. $\pi(n) = n$

- (a) $r = s, s = n - 1$
 - (b) $r \neq s, r = 1$
 - (c) $r \neq s, s = 1$
2. $\pi(n) \neq n$
- (a) $(n, \pi(n)), (\pi(n), n) \in A(G_R)$
 - (b) $(n, \pi(n)) \in A(G_R), (\pi(n), n) \notin A(G_R)$
 - (c) $(n, \pi(n)) \notin A(G_R), (\pi(n), n) \in A(G_R)$
 - (d) $(n, \pi(n)) \notin A(G_R), (\pi(n), n) \notin A(G_R)$
 - i. $t = 0$
 - A. $r = 0, s > 2$
 - B. $s = 0, r > 2$
 - C. $r, s \neq 0$
 - ii. $t \neq 0$
 - A. $r, s \neq 0$
 - B. $r = 0, s > 1$
 - C. $s = 0, r > 1$
 - D. $s = r = 0$

Die Fälle (1a) bis (1c) haben wir bereits in Lemma 2.11 untersucht. Wir wenden uns nun dem Fall (2a) zu:

Lemma 2.13 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$ und $V(P) = \{a_1, \dots, a_r, b_1, \dots, b_s, x_1, \dots, x_t, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Aus den Punkten

- $\pi(n) \neq n$,
- $(n, \pi(n)) \in A(G_R)$ und
- $(\pi(n), n) \in A(G_R)$

folgt $\pi = [n - 2, \dots, 1, n, n - 1]$ und $l = 5n - 5$.

Beweis: Wegen $(n, \pi(n)) \in A(G_R)$ und $(\pi(n), n) \in A(G_R)$ folgt wegen (2.6) und mit Lemma 2.12:

$$\begin{aligned} G_3^j &= (\{n\}, \emptyset) \\ G_4^j &= K_{x_1, x_t} \cup (\{x_t, \pi(n)\}, \{(x_t, \pi(n))\}), \end{aligned}$$

$$\begin{array}{l}
A : \\
Z : \\
T_j : \quad H_1 : n \quad \cdot \\
\quad \quad H_2 : \pi(n) x_t \cdots x_1
\end{array}$$

Es gilt somit $r = s = 0$ und damit $t = n - 2$. Um nun, von T_j ausgehend, $\pi(n)$ auf den Zielstapel legen zu können, müssen die Stapelelemente x_1 bis x_{n-2} auf den Hilfsstapel H_1 , auf n gelegt werden. Somit gilt:

$$\begin{array}{l}
(x_1, x_2), \dots, (x_{n-3}, x_{n-2}) \in A(G_R) \\
\text{und } (x_1, n), (x_{n-2}, x_{n-3}), \dots, (x_2, x_1) \in A(G_R).
\end{array}$$

Wegen $(\pi(n), n) \in A(G_A)$ folgt $(x_1, n) \in A(G_Z)$, darum folgt wiederum $(x_1, x_2) \in A(G_A)$. Nach Lemma 2.6 gilt damit

$$\begin{array}{l}
(x_1, x_2), \dots, (x_{n-3}, x_{n-2}) \in A(G_A) \\
\text{und } (x_{n-2}, x_{n-3}), \dots, (x_2, x_1) \in A(G_Z).
\end{array}$$

G_A und G_Z sehen also wie folgt aus:

$$\begin{array}{l}
G_A = K_{x_1, x_{n-2}} \cup (\{x_{n-2}, \pi(n), n\}, \{(x_{n-2}, \pi(n)), (\pi(n), n)\}) \\
G_Z = K_{x_{n-2}, x_1} \cup (\{x_1, \pi(n), n\}, \{(x_1, n), (n, \pi(n))\}).
\end{array}$$

Weil P ein normiertes Problem ist, folgt $\pi(n) = [n - 2, \dots, 1, n, n - 1]$. Um von T_j zu T_l zu gelangen sind zunächst x_1 bis x_{n-2} umzustapeln, dann wird $\pi(n)$ auf den Zielstapel gelegt. Anschließend werden die Elemente x_{n-2} bis x_1 wiederum umgestapelt, n wird auf den Zielstapel gelegt und nun folgen x_{n-2} bis x_1 . Es sind also noch $n - 2 + 1 + n - 2 + 1 + n - 2$ Umstapelungen notwendig, damit ist $l = j + 3(n - 2) + 2$.

Um von T_1 nach T_j zu gelangen, werden zunächst x_1 bis x_{n-2} ausgestapelt, und anschließend $\pi(n)$ auf den freigebliebenen Hilfsstapel gelegt. Nun werden die Elemente x_{n-2} bis x_1 auf $\pi(n)$ gestapelt und es wird das verbleibende Element n auf den freigewordenen Hilfsstapel gelegt. Es werden also $n - 2 + 1 + n - 2 + 1$ Umstapelungen ausgeführt, damit ist $j = 1 + 2(n - 2) + 2$ und so gilt $l = 5n - 5 \square$

Wir betrachten nun den Fall (2b):

Lemma 2.14 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$ und $V(P) = \{a_1, \dots, a_r, b_1, \dots, b_s, x_1, \dots, x_t, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Aus

- $\pi(n) \neq n$,
- $(n, \pi(n)) \in A(G_R)$ und
- $(\pi(n), n) \notin A(G_R)$

folgt $\pi = \pi_{P(2,n),n-2} \sqcup [n, n-2]$ und $l = 3n - 1$.

Beweis: Wegen $(n, \pi(n)) \in A(G_R)$ und (2.6) folgt $r = 0$. Nach Lemma 2.12 gilt damit für T_j :

$$\begin{aligned} G_3^j &= (\{n\}, \emptyset) \\ G_4^j &= K_{x_1, x_t} \cup K_{b_1, b_s} \cup (\{x_t, \pi(n), b_1\}, \{(x_t, \pi(n)), (\pi(n), b_1)\}), \end{aligned}$$

$$T_j : \begin{array}{l} A : \\ Z : \\ H_1 : n \\ H_2 : b_s \cdots b_1 \pi(n) x_t \cdots x_1 \end{array} .$$

Um von T_j zu T_l zu gelangen, müssen wir zunächst x_1 bis x_t auf n stapeln und dann $\pi(n)$ auf den Zielstapel legen – so liegt dann insbesondere x_t auf x_{t-1} . Um dann jedoch n auf $\pi(n)$ auf dem Zielstapel legen zu können, müssen wir wieder x_1 bis x_t auf H_2 legen, insbesondere liegt dann x_t auf b_1 . Somit gilt:

$$(x_t, \pi(n)), (x_t, x_{t-1}), (x_t, b_1) \in A(G_R),$$

was (2.5) widerspricht. Um diesen Widerspruch zu vermeiden, kann es entweder keine Elemente x_i oder keine Elemente b_i geben, ist also $t = 0$ oder $s = 0$.

Es sei $t = 0$. Dann gilt für T_{j-1} und T_{j+2} :

$$T_{j-1} : \begin{array}{l} A : n \\ Z : \\ H_1 : \\ H_2 : b_s \cdots b_1 \pi(n) \end{array} , \quad T_{j+2} : \begin{array}{l} A : \\ Z : \pi(n) n \\ H_1 : \\ H_2 : b_s \cdots b_1 \end{array} .$$

Wegen $(\pi(n), b_1) \in A(G_A)$ und $(n, \pi(n)) \in A(G_Z)$ kann die Kante $(b_1, \pi(n))$ nicht im Regelgraphen existieren. Damit ist

$$T_{j-4} : \begin{array}{l} A : n b_1 \pi(n) \\ Z : \\ H_1 : \\ H_2 : b_s \cdots b_2 \end{array} .$$

Daher wissen wir sofort, daß $(b_1, b_2) \in A(G_Z)$ gelten muß. Weil $\pi(n)$ bereits im Ausgangsstapel auf b_1 liegt, kann die Kante (b_2, b_1) im Regelgraph nicht existieren. Damit gilt

$$T_{j+5} : \begin{array}{l} A : \\ Z : \pi(n) n b_2 b_1 \\ H_1 : \\ H_2 : b_s \cdots b_3 \end{array} .$$

Für die Stapelkonfigurationen T_{j-4} und T_{j+5} folgt damit nach Lemma 2.9

$$G_A = W_{b_2, b_s} \cup (\{b_3, \pi(n), b_1, n\}, \{(b_3, \pi(n)), (\pi(n), b_1), (b_1, n)\})$$

und $G_Z = W_{b_3, b_s} \cup (\{b_4, b_1, b_2, n, \pi(n)\}, \{(b_4, b_1), (b_1, b_2), (b_2, n), (n, \pi(n))\})$.

P ist ein normiertes Problem, damit gilt nach Lemma 2.8

$$\pi = \pi_{P(2,s),s-2} \sqcup [n-1, n-4, n, n-2] = \pi_{P(2,n),n-2} \sqcup [n, n-2].$$

Im Fall $s = 0$ haben wir die folgende Situation:

$$T_j : \begin{array}{l} A : \\ Z : \\ H_1 : n \\ H_2 : \pi(n) x_t \cdots x_1 \end{array}$$

Um von T_j aus zu T_l gelangen zu können, müssen wir x_1 bis x_t auf n legen. Damit gilt sowohl

$$(x_1, x_2), \dots, (x_{t-1}, x_t) \in A(G_R)$$

als auch $(x_t, x_{t-1}), \dots, (x_2, x_1) \in A(G_R)$.

Wegen $(n, \pi(n)) \in A(G_Z)$ ist $(x_t, \pi(n)) \in A(G_A)$ und damit gilt $(x_t, x_{t-1}) \in A(G_Z)$. Nach Lemma 2.6 folgt

$$(x_1, x_2), \dots, (x_{t-1}, x_t), (x_t, \pi(n)) \in A(G_A).$$

Weil keine weiteren Stapelelemente vorhanden sind, liegt im Ausgangsstapel $\pi(n)$ auf n : $(\pi(n), n) \in A(G_A)$. Das widerspricht jedoch der Voraussetzung, somit kann der Fall $s = 0$ nicht existieren.

Wie wir gerade gesehen haben, reicht es aus, den Fall $r = 0$ zu betrachten. Es gelten die Voraussetzungen von Lemma 2.9. Angewendet auf die Stapelkonfigurationen T_{j-4} und T_{j+5} liefert es für s ungerade

$$j - 4 = \frac{3}{2}(s - 2 + 1) + 1 \quad \text{und} \quad j + 5 = l - \frac{3}{2}(s - 2 - 1) - 1$$

und damit $l = 3s + 5$. Wegen $r = t = 0$ folgt $s = n - 2$ und somit erhalten wir $l = 3n - 1$. Für s geradzahlig erhalten wir

$$j - 4 = \frac{3}{2}(s - 2) + 2 \quad \text{und} \quad j + 5 = l - \frac{3}{2}(s - 2)$$

und damit für $l = 3s + 5 = 3n - 1$. In jedem Fall ergibt sich demnach $l = 3n - 1$.
□

Der Fall (2c):

Lemma 2.15 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$ und $V(P) = \{a_1, \dots, a_r, b_1, \dots, b_s, x_1, \dots, x_t, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Aus

- $\pi(n) \neq n$,
- $(n, \pi(n)) \notin A(G_R)$ und
- $(\pi(n), n) \in A(G_R)$

folgt $\pi = \pi_{P(1,n),n-1} \sqcup [n - 1]$ und $l = 3n - 1$.

Beweis: Wegen $(\pi(n), n) \in A(G_R)$ und (2.6) folgt $s = 0$. Nach Lemma 2.12 gilt damit für T_j :

$$\begin{aligned} G_3^j &= (\{n, a_1, \dots, a_r\}, \{(n, a_1), (a_1, a_2), \dots, (a_{r-1}, a_r)\}), \\ G_4^j &= (\{x_1, \dots, x_t, \pi(n)\}, \{(x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t), (x_t, \pi(n))\}), \end{aligned}$$

$$\begin{array}{l} T_j : \\ \text{A :} \\ \text{Z :} \\ \text{H}_1 : a_r \cdots a_1 n \quad \cdot \\ \text{H}_2 : \pi(n) x_t \cdots x_1 \end{array}$$

Um von T_1 zu T_j zu gelangen, muß es eine Stapelkonfiguration T_g mit $1 < g < j$ geben, die wie folgt aussieht:

$$\begin{array}{l} T_g : \\ \text{A : } n \pi(n) \\ \text{Z :} \\ \text{H}_1 : a_r \cdots a_1 x_1 \cdots x_t \quad \cdot \\ \text{H}_2 : \end{array}$$

Es gilt demnach

$$(x_1, x_2), (x_1, a_1) \in A(G_R). \quad (2.11)$$

Um, von T_j ausgehend, $\pi(n)$ auf den Zielstapel legen zu können, müssen wir die Elemente x_1 bis x_t auf n legen – es liegt dann insbesondere x_1 auf n : $(x_1, n) \in A(G_R)$. Mit (2.11) und (2.5) folgt jedoch, daß entweder keine Elemente x_i oder keine Elemente a_i existieren, also $t = 0$ oder $r = 0$ gilt.

Es sei $t = 0$. Dann gilt für T_{j-2} und T_{j+1} :

$$T_{j-2} : \begin{array}{l} A : n \pi(n) \\ Z : \\ H_1 : a_r \cdots a_1 \\ H_2 : \end{array}, \quad T_{j+1} : \begin{array}{l} A : \\ Z : \pi(n) \\ H_1 : a_r \cdots a_1 n \\ H_2 : \end{array}$$

Wir betrachten die Konfiguration T_{j+1} : Wegen $(n, \pi(n)) \notin A(G_R)$ folgt für T_{j+2} :

$$T_{j+2} : \begin{array}{l} A : \\ Z : \pi(n) \\ H_1 : a_r \cdots a_1 \\ H_2 : n \end{array}.$$

Aus der Voraussetzung $(\pi(n), n) \in A(G_R)$ folgt nach Lemma 2.4

$$(\pi(n), n) \in A(G_A) \quad (2.12)$$

Für $(n, a_1) \in A(G_R)$ gilt nach Lemma 2.4: $(n, a_1) \in A(G_Z)$. Wäre $(a_1, n) \in A(G_R)$, so folgte $(a_1, n) \in A(G_A)$, das widerspricht jedoch (2.12). Somit gilt $(a_1, n) \notin A(G_R)$ und damit gilt für T_{j+4} :

$$T_{j+4} : \begin{array}{l} A : \\ Z : \pi(n) a_1 n \\ H_1 : a_r \cdots a_2 \\ H_2 : \end{array}.$$

Nach Lemma 2.9 für T_{j-2} und T_{j+4} folgt damit:

$$\begin{aligned} G_A &= W_{a_1, a_r} \cup (\{a_2, \pi(n), n\}, \{(a_2, \pi(n)), (\pi(n), n)\}) \\ \text{und} \quad G_Z &= W_{a_2, a_r} \cup (\{a_3, a_1, n, \pi(n)\}, \{(a_3, n), (n, a_1), (a_1, \pi(n))\}). \end{aligned}$$

P ist ein normiertes Problem, damit gilt mit Lemma 2.8

$$\pi = \pi_{P(1,r),r-1} \sqcup [n, n-3, n-1] = \pi_{P(1,n),n-1} \sqcup [n-1].$$

Im Fall $r = 0$ haben wir die folgende Situation:

$$T_j : \begin{array}{l} A : \\ Z : \\ H_1 : n \\ H_2 : \pi(n) x_t \cdots x_1 \end{array} .$$

Um, von T_j aus, zu T_l gelangen zu können, müssen wir x_1 bis x_t auf n legen, damit gilt sowohl

$$\begin{array}{l} (x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t) \in A(G_R) \\ \text{als auch} \quad (x_t, x_{t-1}), \dots, (x_2, x_1), (x_1, n) \in A(G_R). \end{array}$$

Wegen $(\pi(n), n) \in A(G_A)$ ist $(x_1, n) \in A(G_Z)$, deswegen folgt $(x_1, x_2) \in A(G_A)$. Nach Lemma 2.6 folgt:

$$(x_t, x_{t-1}), \dots, (x_2, x_1), (x_1, n) \in A(G_Z).$$

Weil keine weiteren Stapelelemente existieren, liegt somit im Zielstapel n auf $\pi(n)$: $(n, \pi(n)) \in A(G_Z)$. Das widerspricht jedoch der Voraussetzung, somit kann der Fall $r = 0$ nicht existieren. Wie wir gerade gesehen haben, genügt es, den Fall $t = 0$ zu betrachten. Die Anwendung von Lemma 2.9 auf die Stapelkonfigurationen T_{j-2} und T_{j+4} liefert für r geradzahlig

$$j - 2 = \frac{3}{2}(r - 1 + 1) + 1 \quad \text{und} \quad j + 4 = l - \frac{3}{2}(r - 1 - 1) - 1$$

und damit $l = 3r + 5$. Wegen $s = t = 0$ folgt $r = n - 2$ und somit erhalten wir $l = 3n - 1$. Für r ungerade erhalten wir

$$j - 2 = \frac{3}{2}(r - 1) + 2 \quad \text{und} \quad j + 4 = l - \frac{3}{2}(r - 1)$$

und damit für $l = 3r + 5 = 3n - 1$. In jedem Fall ergibt sich demnach $l = 3n - 1$. \square

Betrachten wir den Fall (2(d)i):

Für die Stapelkonfiguration T_j aus Lemma 2.12 untersuchen wir nun eine Reihe von Spezialfällen. Zunächst gelte $t = r = 0$ und $s > 2$ (Fall (2(d)iA)).

Lemma 2.16 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{b_1, \dots, b_{n-2}, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin$

$A(G_R)$ und $(\pi(n), n) \notin A(G_R)$.

Für

$$T_j = ((\emptyset, \emptyset), (\emptyset, \emptyset), (n, \emptyset), (\{\pi(n), b_1\}, \{(\pi(n), b_1)\}) \cup K_{b_1, b_{n-2}}),$$

$$T_j : \begin{array}{l} A : \\ Z : \\ H_1 : n \\ H_2 : b_{n-2} \cdots b_1 \pi(n) \end{array}$$

folgt

1.

$$\begin{array}{l} \pi = \pi_{P(n,2),n-4} \sqcup [n, n-1, n-4, n-2] \\ \text{oder } \pi = [n-1, n, n-2, \dots, 1] \\ \text{oder } \pi = [n, n-1, n-2, \dots, 1] \\ \text{oder } \pi = \pi_{P(n-i,1),n-i-3} \sqcup [n-1, n-i-3, n, n-2, \dots, n-i-1], \\ \quad \text{für } i \in \{2, \dots, n-4\}, \\ \text{oder } \pi = [n-1, 1, n, n-2, \dots, 2] \\ \text{oder } \pi = [n, n-1, 1, n-2, \dots, 2] \\ \text{oder } \pi = [2, n, n-1, 1, n-2, \dots, 3] \\ \text{oder } \pi = [n, 2, n-1, 1, n-2, \dots, 3] \\ \text{oder } \pi = \pi_{P(n-i,1),n-i-3} \sqcup [n, n-1, n-i-3, n-2, \dots, n-i-1], \\ \quad \text{für } i \in \{2, \dots, n-5\} \text{ und} \end{array}$$

2. $l = 3n - 1$.

Beweis: Wir zeigen die Gültigkeit von (1) und berechnen dabei für jede Permutation das entsprechende l .

Wenn wir, von T_j ausgehend, T_l erreichen wollen, so legen wir zunächst das Element $\pi(n)$ auf den Zielstapel und erhalten die Konfiguration T_{j+1} :

$$T_{j+1} : \begin{array}{l} A : \\ Z : \pi(n) \\ H_1 : n \\ H_2 : b_{n-2} \cdots b_1 \end{array} .$$

In den Voraussetzungen haben wir die Kante $(n, \pi(n))$ im Regelgraphen ausgeschlossen. Um weiter zu T_{j+2} zu gelangen, bleiben demnach die beiden folgenden Möglichkeiten:

1. $(b_1, n) \in A(G_R)$ oder
2. $(b_1, \pi(n)) \in A(G_R)$.

Nun untersuchen wir diese Möglichkeiten der Reihe nach.

Es gelte $(b_1, n) \in A(G_R)$ und $(b_1, \pi(n)) \notin A(G_R)$. Wegen $(\pi(n), n) \notin A(G_R)$ und $(b_1, \pi(n)) \notin A(G_R)$ folgt für T_{j-3} :

$$T_{j-3} : \begin{array}{l} A : n b_1 \\ Z : \\ H_1 : \pi(n) \\ H_2 : b_{n-2} \cdots b_2 \end{array} .$$

In T_j liegt $\pi(n)$ auf b_1 , damit gilt $(\pi(n), b_1) \in A(G_A)$ und somit ist:

$$T_{j-4} : \begin{array}{l} A : n b_1 \pi(n) \\ Z : \\ H_1 : \\ H_2 : b_{n-2} \cdots b_2 \end{array} .$$

Weil b_1 im Ausgangsstapel auf n liegt, folgt $(b_1, b_2) \in A(G_Z)$. Für T_{j+2} gilt:

$$T_{j+2} : \begin{array}{l} A : \\ Z : \pi(n) \\ H_1 : n b_1 \\ H_2 : b_{n-2} \cdots b_2 \end{array} .$$

Nach Lemma 2.5 existiert die Kante (b_2, b_1) in G_R nicht, damit erhalten wir sofort

$$T_{j+4} : \begin{array}{l} A : \\ Z : \pi(n) b_2 b_1 \\ H_1 : n \\ H_2 : b_{n-2} \cdots b_3 \end{array} .$$

Damit ist $(b_2, b_3) \in A(G_A)$. Mit $(b_1, b_2) \in A(G_Z)$ und $(b_2, b_3) \in A(G_A)$ folgt nach Lemma 2.5, daß (b_3, b_2) in G_R nicht existiert. Für T_{j-7} gilt somit:

$$T_{j-7} : \begin{array}{l} A : n b_1 \pi(n) b_3 b_2 \\ Z : \\ H_1 : \\ H_2 : b_{n-2} \cdots b_4 \end{array} .$$

Weil im Ausgangsstapel somit b_3 auf $\pi(n)$ liegt, folgt $(b_3, b_4) \in A(G_Z)$ und damit $(b_3, n) \notin A(G_R)$. So erhalten wir für T_{j+5} :

$$T_{j+5} : \begin{array}{l} A : \\ Z : \pi(n) b_2 b_1 n \\ H_1 : \\ H_2 : b_{n-2} \cdots b_3 \end{array} .$$

Wir wenden nun das Lemma 2.9 auf T_{j-4} und T_{j+5} an und erhalten

$$\begin{aligned} G_A &= W_{b_2, b_{n-2}} \cup (\{b_1, b_3, n, \pi(n)\}, \\ &\quad \{b_3, \pi(n)\}, (\pi(n), b_1), (b_1, n)\}) \text{ und} \\ G_Z &= W_{b_3, b_{n-2}} \cup (\{b_1, b_2, b_4, n, \pi(n)\}, \{(b_4, n), (n, b_1), (b_1, b_2), (b_2, \pi(n))\}). \end{aligned}$$

Da P ein normiertes Problem ist, folgt mit Lemma 2.8

$$\pi = \pi_{P(n,2), n-4} \sqcup [n, n-1, n-4, n-2].$$

Ebenfalls nach Lemma 2.9 erhalten wir für n ungerade:

$$j-4 = \frac{3}{2}(n-2-2+1) + 1, \quad j+5 = l - \frac{3}{2}(n-2-2-1) - 1$$

und für n geradzahlig:

$$j-4 = \frac{3}{2}(n-2-2) + 2, \quad j+5 = l - \frac{3}{2}(n-2-2).$$

Es folgt jeweils $l = 3n - 1$.

Wir untersuchen die zweite Möglichkeit: Es gelte $(b_1, \pi(n)) \in A(G_R)$ und $(b_1, n) \notin A(G_R)$. Aus der Konfiguration T_j folgt ferner $(\pi(n), b_1) \in A(G_A)$. Damit wissen wir, daß

$$\begin{aligned} (\pi(n), b_1) &\in A(G_A), \\ (b_1, \pi(n)) &\in A(G_Z) \text{ und} \\ (b_1, b_2) &\in A(G_A) \text{ ist.} \end{aligned}$$

Mit $(b_1, n) \notin A(G_R)$ folgt

$$T_{j-3} : \begin{array}{l} A : n \\ Z : \\ H_1 : \pi(n) b_1 \\ H_2 : b_{n-2} \cdots b_2 \end{array} \quad \text{und} \quad T_{j+2} : \begin{array}{l} A : \\ Z : \pi(n) b_1 \\ H_1 : n \\ H_2 : b_{n-2} \cdots b_2 \end{array} .$$

Es sei nun $i \in \{2, \dots, n-2\}$ der kleinste Index, für den $(b_i, b_{i-1}) \notin A(G_R)$ gilt. Mit $(b_1, b_2) \in A(G_A)$ und Lemma 2.6 folgt:

$$\begin{aligned} & (b_1, b_2), (b_2, b_3), \dots, (b_{i-2}, b_{i-1}) \in A(G_A) \\ \text{und} & \quad (b_{i-1}, b_{i-2}), \dots, (b_3, b_2), (b_2, b_1) \in A(G_Z) \end{aligned}$$

Somit gilt:

$$\begin{array}{l} T_{j-i-1} : \\ \quad A : n \\ \quad Z : \\ \quad H_1 : \pi(n) b_1 \cdots b_{i-1} \\ \quad H_2 : b_{n-2} \cdots b_i \end{array} \quad \text{und} \quad \begin{array}{l} T_{j+i} : \\ \quad A : \\ \quad Z : \pi(n) b_1 \cdots b_{i-1} \\ \quad H_1 : n \\ \quad H_2 : b_{n-2} \cdots b_i \end{array} .$$

Wegen $(b_{i-1}, b_{i-2}) \in A(G_Z)$ und $(b_{i-1}, b_i) \in A(G_R)$ folgt $(b_{i-1}, b_i) \in A(G_A)$ und $(b_{i-1}, n) \notin A(G_R)$. Damit erhalten wir

$$\begin{array}{l} T_{j-2i-1} : \\ \quad A : n b_i b_{i-1} \cdots b_1 \\ \quad Z : \\ \quad H_1 : \pi(n) \\ \quad H_2 : b_{n-2} \cdots b_{i+1} \end{array} \quad \text{und} \quad \begin{array}{l} T_{j-2i-2} : \\ \quad A : n b_i b_{i-1} \cdots b_1 \pi(n) \\ \quad Z : \\ \quad H_1 : \\ \quad H_2 : b_{n-2} \cdots b_{i+1} \end{array} .$$

Es ist also $(b_i, n) \in A(G_A)$ und damit $(b_i, b_{i+1}) \in A(G_Z)$.

An dieser Stelle fügen wir eine Betrachtung für den Spezialfall $i = n-2$ ein: In diesem Fall gilt $T_{j-2i-2} = T_1$ und damit

$$G_A = (\{\pi(n), b_1, b_i, n\}, \{(\pi(n), b_1), (b_i, n)\}) \cup K_{b_1, b_i}.$$

Für T_l gilt $T_l = T_{j+i+2}$ und

$$\begin{array}{l} T_{j+i+2} : \\ \quad A : \\ \quad Z : \pi(n) b_1 \cdots b_{i-1} b_{n-2} n \\ \quad H_1 : \\ \quad H_2 : \end{array}$$

oder

$$\begin{array}{l} T_{j+i+2} : \\ \quad A : \\ \quad Z : \pi(n) b_1 \cdots b_{i-1} n b_{n-2} \\ \quad H_1 : \\ \quad H_2 : \end{array}$$

und damit

$$G_Z = (\{\pi(n), b_1, b_i, n\}, \{(n, b_i), (b_1, \pi(n))\}) \cup K_{b_i, b_1}$$

oder

$$G_Z = (\{\pi(n), b_1, b_{i-1}, b_i, n\}, \{(b_i, n), (n, b_{i-1}), (b_1, \pi(n))\}) \cup K_{b_{i-1}, b_1}.$$

Weil P normiert ist, gilt demnach

$$\pi = [n-1, n, n-2, \dots, 2, 1] \quad \text{oder} \quad \pi = [n, n-1, n-2, \dots, 2, 1],$$

wegen $j-2i-2=1$ und $l=j+i+2$ gilt $l=3i+5$. Da wir $i=n-2$ betrachteten folgt $l=3n-1$.

Jetzt betrachten wir wieder T_{j+i} für $i \in \{2, \dots, n-3\}$. Auch hier bieten sich uns wieder zwei Möglichkeiten:

1. $(n, b_{i-1}) \in A(G_Z)$ oder
2. $(n, b_{i-1}) \notin A(G_Z)$

Wir untersuchen $(n, b_{i-1}) \in A(G_Z)$: Wegen $(b_i, b_{i+1}) \in A(G_Z)$ folgt:

$$T_{j+i+2} : \begin{array}{l} A : \\ Z : \pi(n) b_1 \cdots b_{i-1} n \\ H_1 : b_i \\ H_2 : b_{n-2} \cdots b_{i+1} \end{array}$$

Nach Lemma 2.5 existiert die Kante (b_{i+1}, b_i) nicht in G_R , damit ist

$$T_{j+i+4} : \begin{array}{l} A : \\ Z : \pi(n) b_1 \cdots b_{i-1} n b_{i+1} b_i \\ H_1 : \\ H_2 : b_{n-2} \cdots b_{i+2} \end{array} .$$

Mit T_{j-2i-2} und T_{j+i+4} folgt nach Lemma 2.9 für $i \in \{2, \dots, n-4\}$:

$$G_A = (\{n, b_i, b_1, \pi(n), b_{i+2}\}, \{(b_{i+2}, \pi(n)), (\pi(n), b_1), (b_i, n)\}) \\ \cup K_{b_1, b_i} \cup W_{b_{i+1}, b_{n-2}}$$

und

$$G_Z = (\{\pi(n), b_1, b_{i-1}, n, b_{i+1}, b_i, b_{i+3}\}, \\ \{(b_{i+3}, b_i), (b_i, b_{i+1}), (b_{i+1}, n), (n, b_{i-1}), (b_1, \pi(n))\}) \\ \cup K_{b_{i-1}, b_1} \cup W_{b_{i+2}, b_{n-2}},$$

für $i=n-3$ erhalten wir

$$G_A = (\{n, b_{n-3}, b_1, \pi(n), b_{n-2}\}, \{(b_{n-2}, \pi(n)), (\pi(n), b_1), (b_{n-3}, n)\}) \\ \cup K_{b_1, b_{n-3}}$$

und

$$G_Z = (\{\pi(n), b_1, b_{n-4}, n, b_{n-2}, b_{n-3}\}, \\ \{(b_{n-3}, b_{n-2}), (b_{n-2}, n), (n, b_{n-4}), (b_1, \pi(n))\}) \\ \cup K_{b_{n-4}, b_1} .$$

Weil P ein normiertes Problem ist, erhalten wir nach Lemma 2.8 für $i \in \{2, \dots, n-4\}$:

$$\pi = \pi_{P(n-i,1),n-i-3} \sqcup [n-1, n-i-3, n, n-2, \dots, n-i-1].$$

Für $i+n$ geradzahlig erhalten wir außerdem

$$j-2i-2 = \frac{3}{2}(n-2-i-1+1) + 1, \quad j+i+4 = l - \frac{3}{2}(n-2-i-2) - 1$$

und für $i+n$ ungerade:

$$j-2i-2 = \frac{3}{2}(n-2-i-1) + 2, \quad j+i+4 = l - \frac{3}{2}(n-2-i-1).$$

In beiden Fällen folgt mit $l = 3n - 1$.

Für $i = n - 3$ erhalten wir

$$\pi = [n-1, 1, n, n-2, \dots, 2],$$

wobei $j-2i-2 = 2$ und $l = j+i+4$ und damit $l = 3i+8 = 3n-1$ gilt.

Wenden wir uns der zweiten Möglichkeit, dem Fall $(n, b_{i-1}) \notin A(G_Z)$ zu, so ist

$$\begin{array}{l} A : \\ T_{j+i+1} : \quad \begin{array}{l} Z : \pi(n) b_1 \cdots b_{i-1} \\ H_1 : n b_i \\ H_2 : b_{n-2} \cdots b_{i+1} \end{array} \end{array} .$$

Wegen $(b_{i-1}, b_i) \in A(G_A)$ und $(b_i, b_{i+1}) \in A(G_Z)$ folgt mit Lemma 2.5: Die Kante (b_{i+1}, b_i) existiert nicht, damit gilt für T_{j+i+3} :

$$\begin{array}{l} A : \\ T_{j+i+3} : \quad \begin{array}{l} Z : \pi(n) b_1 \cdots b_{i-1} b_{i+1} b_i \\ H_1 : n \\ H_2 : b_{n-2} \cdots b_{i+2} \end{array} \end{array} .$$

Somit muß $(b_{i+1}, b_{i+2}) \in A(G_A)$ sein. Wegen $(b_i, b_{i+1}) \in A(G_Z)$ und $(b_{i+1}, b_{i+2}) \in A(G_A)$ kann nach Lemma 2.5 die Kante (b_{i+2}, b_{i+1}) nicht existieren, somit gilt

$$\begin{array}{l} A : n b_i b_{i-1} \cdots b_1 \pi(n) b_{i+2} b_{i+1} \\ T_{j-2i-5} : \quad \begin{array}{l} Z : \\ H_1 : \\ H_2 : b_{n-2} \cdots b_{i+3} \end{array} \end{array} .$$

Damit ist $(b_{i+2}, b_{i+3}) \in A(G_Z)$ und (b_{i+2}, n) kann nicht existieren. Daraus folgt für T_{j+i+4} :

$$T_{j+i+4} : \begin{array}{l} A : \\ Z : \pi(n) b_1 \cdots b_{i-1} b_{i+1} b_i n \\ H_1 : \\ H_2 : b_{n-2} \cdots b_{i+2} \end{array} .$$

Mit den Stapelkonfigurationen T_{j-2i-2} und T_{j+i+4} folgt nach Lemma 2.9 für $i \in \{2, \dots, n-5\}$:

$$\begin{aligned} G_A &= (\{b_{i+2}, \pi(n), b_1, b_i, n\}, \{(b_{i+2}, \pi(n)), (\pi(n), b_1), (b_i, n)\}) \\ &\quad \cup K_{b_1, b_i} \cup W_{b_{i+1}, b_{n-2}} \\ \text{und} \\ G_Z &= (\{b_{i+3}, n, b_i, b_{i+1}, b_{i-1}, b_1, \pi(n)\}, \\ &\quad \{(b_{i+3}, n), (n, b_i), (b_i, b_{i+1}), (b_{i+1}, b_{i-1}), (b_1, \pi(n))\}) \\ &\quad \cup K_{b_{i-1}, b_1} \cup W_{b_{i+2}, b_{n-2}} . \end{aligned}$$

Für $i = n-4$ ergibt sich

$$\begin{aligned} G_A &= (\{b_{i+1}, b_{i+2}, \pi(n), b_1, b_{i-1}, b_i, n\}, \\ &\quad \{(b_{i+1}, b_{i+2}), (b_{i+2}, \pi(n)), (\pi(n), b_1), (b_{i-1}, b_i), (b_i, n)\}) \\ &\quad \cup K_{b_1, b_{i-1}} \end{aligned}$$

und für

$$\begin{aligned} G_Z &= (\{n, b_{n-2}, b_{n-4}, b_{n-3}, b_{n-5}, b_1, \pi(n)\}, \\ &\quad \{(n, b_{n-2}), (b_{n-2}, b_{n-4}), (b_{n-3}, b_{n-5}), (b_1, \pi(n))\}) \\ &\quad \cup K_{b_{n-5}, b_1} \end{aligned}$$

oder

$$\begin{aligned} G_Z &= (\{n, b_{n-2}, b_{n-4}, b_{n-3}, b_{n-5}, b_1, \pi(n)\}, \\ &\quad \{(b_{n-2}, n), (n, b_{n-4}), (b_{n-4}, b_{n-3}), (b_{n-3}, b_{n-5}), (b_1, \pi(n))\}) \\ &\quad \cup K_{b_{n-5}, b_1} . \end{aligned}$$

Für $i = n-3$ ist

$$\begin{aligned} G_A &= (\{b_{n-2}, \pi(n), b_1, b_{n-3}, n\}, \{(b_{n-2}, \pi(n)), (\pi(n), b_1), (b_{n-3}, n)\}) \\ &\quad \cup K_{b_1, b_{n-3}} \end{aligned}$$

und

$$\begin{aligned} G_Z &= (\{n, b_{n-3}, b_{n-2}, b_{n-4}, b_1, \pi(n)\}, \\ &\quad \{(n, b_{n-3}), (b_{n-3}, b_{n-2}), (b_{n-2}, b_{n-4}), (b_1, \pi(n))\}) \\ &\quad \cup K_{b_{n-4}, b_1} . \end{aligned}$$

Weil P ein normiertes Problem ist, erhalten wir: für $i = n - 3$

$$\pi = [n, n - 1, 1, n - 2, \dots, 2]$$

mit $j - 2i - 2 = 2$ und $j + i + 4 = l$. Es folgt $l = 3n - 1$.

Für $i = n - 4$ ist

$$\pi = [2, n, n - 1, 1, n - 2, \dots, 3]$$

oder

$$\pi = [n, 2, n - 1, 1, n - 2, \dots, 3]$$

mit $j - 2i - 5 = 1$ und $j + i + 5 = l$. Damit ist auch hier $l = 3n - 1$.

Für $i \in \{2, \dots, n - 5\}$ gilt mit Lemma 2.8

$$\pi = \pi_{P(n-i,1),n-i-3} \sqcup [n, n - 1, n - i - 3, n - 2, \dots, n - i - 1]$$

und nach Lemma 2.9 gilt für $n + i$ geradzahlig

$$\begin{aligned} j - 2i - 2 &= \frac{3}{2}(n - 2 - i - 1 + 1) + 1, \\ j + i + 4 &= l - \frac{3}{2}(n - 2 - i - 1 - 1) - 1 \end{aligned}$$

und für $i + n$ ungerade:

$$\begin{aligned} j - 2i - 2 &= \frac{3}{2}(n - 2 - i - 1) + 2, \\ j + i + 4 &= l - \frac{3}{2}(n - 2 - i - 1). \end{aligned}$$

Es folgt beide Male $l = 3n - 1$. \square

Wir verbleiben bei der Betrachtung des Falles (2(d)i) und nehmen nun Untersuchungen für $t = s = 0$ und $r > 2$ (Fall (2(d)iB)) vor:

Lemma 2.17 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{a_1, \dots, a_{n-2}, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin A(G_R)$ und $(\pi(n), n) \notin A(G_R)$. Für

$$T_j = ((\emptyset, \emptyset), (\emptyset, \emptyset), (\{n, a_1\}, \{(n, a_1)\}) \cup K_{a_1, a_{n-2}}, (\{\pi(n)\}, \emptyset)),$$

$$\begin{array}{l} A : \\ Z : \\ T_j : \\ \quad H_1 : a_{n-2} \cdots a_1 n \\ \quad H_2 : \pi(n) \end{array}$$

folgt

1.

$$\begin{aligned}
\pi &= \hat{\pi}^{-1} \text{ mit} \\
&\hat{\pi} = \pi_{P(n,2),n-4} \sqcup [n, n-1, n-4, n-2] \text{ oder} \\
&\hat{\pi} = \pi_{P(n-i,1),n-i-3} \sqcup [n-1, n-i-3, n, n-2, \dots, n-i-1], \\
&\quad \text{für } i \in \{2, \dots, n-4\}, \text{ oder} \\
&\hat{\pi} = \pi_{P(n-i,1),n-i-3} \sqcup [n, n-1, n-i-3, n-2, \dots, n-i-1], \\
&\quad \text{für } i \in \{2, \dots, n-5\}, \text{ oder} \\
\text{oder } \pi &= [n, n-1, n-2, \dots, 3, 1, 2] \\
\text{oder } \pi &= [n, n-1, \dots, 3, 2, 1] \\
\text{oder } \pi &= [2, n, n-1, \dots, 4, 1, 3] \\
\text{oder } \pi &= [3, n, n-1, \dots, 4, 2, 1] \\
\text{oder } \pi &= [4, 1, n, n-1, \dots, 5, 3, 2] \\
\text{oder } \pi &= [4, 2, n, n-1, \dots, 5, 3, 1] \text{ und}
\end{aligned}$$

2. $l = 3n - 1$.

Beweis: Vergleichen wir T_j mit der Stapelkonfiguration T_j aus Lemma 2.16, so sind hier nur n und $\pi(n)$ vertauscht worden. Berechnen wir also hier Ausgangs- und Zielstapel, so sind es die gleichen Stapel wie im Lemma 2.16 – nur eben vertauscht. Berücksichtigen wir, daß P normiert ist, so ergibt sich für $\pi(P)$ jeweils die inverse Permutation von $\pi(P)$ aus Lemma 2.16. Man sieht leicht, daß in (1) genau diese inversen Permutationen angegeben sind.

Mit der obigen Argumentation ist klar, daß sich die Anzahl der notwendigen Umstapelungen gegenüber dem Fall in Lemma 2.16 nicht verändert, also ebenfalls $l = 3n - 1$ gilt. \square

Bemerkung: An dieser Stelle soll erwähnt sein, daß im Lemma 2.17 2 Permutationen auftauchen, die wir schon im Lemma 2.16 erhielten:

$$\pi = \hat{\pi}^{-1} \text{ mit } \hat{\pi} = \pi_{P(n,2),n-4} \sqcup [n, n-1, n-4, n-2]$$

ist identisch mit der Permutation

$$\pi = \pi_{P(n-i,1),n-i-3} \sqcup [n-1, n-i-3, n, n-2, \dots, n-i-1]$$

aus Lemma 2.16 für $i = 2$. Daraus folgt, daß auch die Permutation

$$\pi = \hat{\pi}^{-1} \text{ mit } \hat{\pi} = \pi_{P(n-i,1),n-i-3} \sqcup [n-1, n-i-3, n, n-2, \dots, n-i-1]$$

für $i = 2$ aus Lemma 2.17 identisch mit der Permutation

$$\pi = \pi_{P(n,2),n-4} \sqcup [n, n-1, n-4, n-2]$$

aus Lemma 2.16 ist.

Als letztes im Fall (2(d)i) führen wir Betrachtungen für $t = 0$ und $r, s \neq 1$ durch:

Lemma 2.18 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{a_1, \dots, a_r, b_1, \dots, b_s, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin A(G_R)$ und $(\pi(n), n) \notin A(G_R)$. Für

$$T_j : \begin{array}{l} \text{A :} \\ \text{Z :} \\ \text{H}_1 : a_r \cdots a_1 n \\ \text{H}_2 : b_s \cdots b_1 \pi(n) \end{array}$$

folgt dann: $\pi = [n, n-1, \dots, 1]$ und $l = 3n - 1$.

Beweis: Wir betrachten T_{j-1} :

$$T_{j-1} : \begin{array}{l} \text{A : } n \\ \text{Z :} \\ \text{H}_1 : a_r \cdots a_1 \\ \text{H}_2 : b_s \cdots b_1 \pi(n) \end{array} .$$

Nach Voraussetzung existiert die Kante $(\pi(n), n)$ nicht in G_R und es gilt $d_{G_R}^+(\pi(n)) \leq 1$, somit muß ein $u \in \{1, \dots, r\}$ existieren, mit

$$(a_u, n) \in A(G_A). \quad (2.13)$$

Auf dem Hilfsstapel H_2 in der Stapelkonfiguration T_{j-1} liegt $\pi(n)$ auf b_1 , daß heißt, es muß $(\pi(n), b_1) \in A(G_A)$ gelten. Man macht sich leicht klar, daß die Elemente a_1 bis a_r auf den Ausgangsstapel gelegt werden müssen, bevor b_1 dorthin gelangen kann. Somit gilt für alle $a_i \in \{a_1, \dots, a_r\}$:

$$(a_i, \pi(n)) \notin A(G_A). \quad (2.14)$$

Analog betrachten wir T_{j+1} :

$$T_{j+1} : \begin{array}{l} \text{A :} \\ \text{Z : } \pi(n) \\ \text{H}_1 : a_r \cdots a_1 n \\ \text{H}_2 : b_s \cdots b_1 \end{array} .$$

Da wir n nicht auf $\pi(n)$ legen dürfen und $d_{G_R}^+(u) \leq 1$ gilt, muß ein $v \in \{1, \dots, s\}$ existieren, so daß

$$(b_v, \pi(n)) \in A(G_Z) \text{ gilt.} \quad (2.15)$$

Auf dem Hilfsstapel H_1 liegt n auf a_1 , daß heißt, es muß $(n, a_1) \in A(G_Z)$ gelten. Man überlegt sich leicht, daß die Elemente b_1 bis b_s auf den Zielstapel gelegt werden müssen, bevor a_1 dorthin gelangen kann. Somit gilt für alle $b_i \in \{b_1, \dots, b_s\}$:

$$(b_i, n) \notin A(G_Z). \quad (2.16)$$

Kommen wir nun zu T_{j-1} zurück: Angenommen, es gäbe ein $a_i \in \{a_1, \dots, a_r\}$ für das $(a_i, \pi(n)) \in A(G_R)$ gilt. Dann folgt nach (2.14): $(a_i, \pi(n)) \in A(G_Z)$, das wiederum widerspricht (2.15). Demnach ist die Annahme falsch und es gilt für alle $a_i \in \{a_1, \dots, a_r\}$:

$$(a_i, \pi(n)) \notin A(G_R). \quad (2.17)$$

Betrachten wir wieder T_{j+1} : Angenommen, es gäbe ein $b_i \in \{b_1, \dots, b_s\}$, für das $(b_i, n) \in A(G_R)$ gilt, dann folgte nach (2.16) $(b_i, n) \in A(G_A)$, was wiederum (2.13) widerspricht. Demnach ist die Annahme falsch und es gilt für alle $b_i \in \{b_1, \dots, b_s\}$:

$$(b_i, n) \notin A(G_R). \quad (2.18)$$

Gehen wir von T_{j-1} aus und berücksichtigen wir (2.17), so erhalten wir:

$$T_{j-1-r} : \begin{array}{l} A : n a_1 \cdots a_r \\ Z : \\ H_1 : \\ H_2 : b_s \cdots b_1 \pi(n) \end{array},$$

von T_{j+1} ausgehend erhalten wir unter Berücksichtigung von (2.18):

$$T_{j+1+s} : \begin{array}{l} A : \\ Z : \pi(n) b_1 \cdots b_s \\ H_1 : a_r \cdots a_1 n \\ H_2 : \end{array}.$$

Es gilt daher

$$\begin{array}{l} (a_1, a_2), (a_2, a_3), \dots, (a_{r-1}, a_r) \in A(G_Z) \\ \text{und} \quad (b_1, b_2), (b_2, b_3), \dots, (b_{s-1}, b_s) \in A(G_A). \end{array}$$

Mit $(\pi(n), b_1) \in A(G_A)$ und $(n, a_1) \in A(G_Z)$ folgt

$$T_{j-r-2s-2} : \begin{array}{l} A : n a_1 \cdots a_r b_s \cdots b_1 \pi(n) \\ Z : \\ H_1 : \\ H_2 : \end{array}$$

und

$$\begin{array}{l}
 \text{A :} \\
 \text{Z : } \pi(n) b_1 \cdots b_s a_r \cdots a_1 n \\
 T_{j+2r+s+2} : \text{H}_1 : \\
 \text{H}_2 :
 \end{array}
 .$$

Es ist also

$$\begin{aligned}
 G_A &= (\{\pi(n), b_1, b_s, a_r, a_1, n\}, \{(\pi(n), b_1), (b_s, a_r), (a_1, n)\}) \\
 &\quad \cup K_{b_1, b_s} \cup K_{a_r, a_1} \\
 G_Z &= (\{n, a_1, a_r, b_s, b_1, \pi(n)\}, \{(n, a_1), (a_r, b_s), (b_1, \pi(n))\}) \\
 &\quad \cup K_{a_1, a_r} \cup K_{b_s, b_1}.
 \end{aligned}$$

Weil P ein normiertes Problem ist, ergibt sich für $\pi = [n, n-1, \dots, 2, 1]$.

Es ist $T_1 = T_{j-r-2s-2}$ und $T_l = T_{j+2r+s+2}$, damit ist $1 = j - 2s - r - 2$ und $l = j + 2r + s + 2$. Mit $r + s + 2 = n$ folgt $l = 3n - 1$. \square

Wir betrachten nun den Fall (2(d)ii): Auch diesen Fall haben wir wieder zu unterteilen. Beginnen wir mit (2(d)iiA).

Lemma 2.19 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{a_1, \dots, a_r, b_1, \dots, b_s, x_1, \dots, x_t, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin A(G_R)$ und $(\pi(n), n) \notin A(G_R)$. Dann gilt $\pi = [n, 3, 2, n-1, \dots, 4, 1]$ mit $l = 3n - 1$.

Beweis: Nach Lemma 2.12 hat T_j folgende Gestalt:

$$\begin{array}{l}
 \text{A :} \\
 \text{Z :} \\
 T_j : \text{H}_1 : a_r \cdots a_1 n \\
 \text{H}_2 : b_s \cdots b_1 \pi(n) x_t \cdots x_1
 \end{array}
 .$$

Um zu T_l zu gelangen, müssen wir in einer Stapelkonfiguration $\pi(n)$ auf den Zielstapel legen. Wenn wir T_j betrachten, so stellen wir fest, daß wir, um auf $\pi(n)$ zugreifen zu können, zunächst die Elemente x_1 bis x_t auf n auf den Hilfsstapel

H_1 legen müssen. Wir erhalten also

$$T_{j+t} : \begin{array}{l} A : \\ Z : \\ H_1 : a_r \cdots a_1 n x_1 \cdots x_t \\ H_2 : b_s \cdots b_1 \pi(n) \end{array} .$$

Es folgt damit

$$\begin{aligned} (\pi(n), b_1) &\in A(G_A), & (n, a_1) &\in A(G_Z), \\ (x_1, n) &\in A(G_R), & (x_t, \pi(n)) &\in A(G_R) \end{aligned} \quad (2.19)$$

$$\text{und} \quad \begin{aligned} (x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t) &\in A(G_R), \\ (x_t, x_{t-1}), \dots, (x_3, x_2), (x_2, x_1) &\in A(G_R). \end{aligned} \quad (2.20)$$

Jetzt betrachten wir T_{j-1} :

$$T_{j-1} : \begin{array}{l} A : n \\ Z : \\ H_1 : a_r \cdots a_1 \\ H_2 : b_s \cdots b_1 \pi(n) x_t \dots x_1 \end{array} .$$

Welches Element liegt nun im Ausgangsstapel auf n ? Dafür gibt es nur 2 Möglichkeiten:

1. Es liegt ein $a_i \in \{a_1, \dots, a_r\}$ auf n auf dem Ausgangsstapel oder
2. x_1 liegt auf n auf dem Ausgangsstapel.

Wir nehmen zunächst an, es gilt die Möglichkeit (1): Weil nach (2.19) und (2.20) x_1 nur auf n und x_2 liegen darf, aber schon a_i auf n liegt, folgt $(x_1, x_2) \in A(G_A)$. Um aber x_2 auf den Ausgangsstapel bringen zu können, muß x_1 auf den Hilfsstapelplatz H_1 gelegt werden. Wegen $(x_1, x_2), (x_1, n) \in A(G_R)$ und (2.5) geht das jedoch erst, wenn alle Elemente a_1, \dots, a_r auf dem Ausgangsstapel liegen – in welcher Reihenfolge diese Elemente dort liegen interessiert uns vorerst nicht.

$$T_{j-y} : \begin{array}{l} A : n \boxed{a} \\ Z : \\ H_1 : \\ H_2 : b_s \cdots b_1 \pi(n) x_t \dots x_1 \end{array} .$$

Nach (2.19) und (2.20) folgt, daß keines der Elemente x_1 bis x_t auf einem der Elemente a_1 bis a_r liegen darf. Damit erhalten wir:

$$\begin{array}{rcl}
 & \text{A : } & n \boxed{a} \\
 & \text{Z :} & \\
 T_{j-y-t} : & \text{H}_1 : & x_1 \dots x_t \\
 & \text{H}_2 : & b_s \dots b_1 \pi(n)
 \end{array} .$$

Nach (2.19) und (2.6) kann $\pi(n)$ weder auf x_t noch auf dem obersten Element des Ausgangsstapels in T_{j-y-t} liegen. x_t kann in $T_{j-y-t-1}$ nicht auf $\pi(n)$ liegen, weil sonst $\mathcal{T}' = [T_1, \dots, T_{j-y-t-1}, T_{j-y-t+2}, \dots, T_l]$ eine kürzere Umstapelfolge als \mathcal{T} wäre, was der Voraussetzung widerspricht. Somit muß in $T_{j-y-t-1}$ x_t auf der Spitze des Ausgangsstapels und somit auf einem $a_i \in \{a_1, \dots, a_r\}$ liegen. Das widerspricht jedoch (2.5). Somit ist unsere Annahme falsch und es existiert die Möglichkeit (1) nicht.

Weil nur Möglichkeit (2) in Frage kommt, gilt für T_{j-2} :

$$\begin{array}{rcl}
 & \text{A : } & n x_1 \\
 & \text{Z :} & \\
 T_{j-2} : & \text{H}_1 : & a_r \dots a_1 \\
 & \text{H}_2 : & b_s \dots b_1 \pi(n) x_t \dots x_2
 \end{array} .$$

Damit ist $(x_1, n) \in A(G_A)$ und nach (2.20) gilt somit $(x_1, x_2) \in A(G_Z)$. Aus (2.20) folgt mit Lemma 2.6:

$$\begin{array}{rcl}
 & (x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t) \in A(G_Z) & (2.21) \\
 \text{und} & (x_t, x_{t-1}), \dots, (x_3, x_2), (x_2, x_1) \in A(G_A). &
 \end{array}$$

Daher gilt

$$\begin{array}{rcl}
 & \text{A : } & n x_1 \dots x_t \\
 & \text{Z :} & \\
 T_{j-t-1} : & \text{H}_1 : & a_r \dots a_1 \\
 & \text{H}_2 : & b_s \dots b_1 \pi(n)
 \end{array} .$$

Nach (2.6) und wegen $(\pi(n), b_1) \in A(G_A)$ folgt

$$\begin{array}{rcl}
 & \text{A : } & n x_1 \dots x_t \boxed{a} \\
 & \text{Z :} & \\
 T_{j-t-y} : & \text{H}_1 : & \\
 & \text{H}_2 : & b_s \dots b_1 \pi(n)
 \end{array} ,$$

wobei uns die Reihenfolge der Elemente a_1 bis a_r auf dem Ausgangsstapel in T_{j-t-y} nicht interessiert. Aus (2.21) folgt $(x_t, \pi(n)) \in A(G_Z)$. Nach (2.19) wissen

wir, daß $(\pi(n), b_1) \in A(G_A)$ gilt. Damit existiert nach Lemma 2.5 die Kante $(b_1, \pi(n))$ nicht. Wir erhalten so

$$T_{j-t-y-3} : \begin{array}{l} \text{A : } n x_1 \cdots x_t \boxed{a} b_1 \pi(n) \\ \text{Z :} \\ \text{H}_1 : \\ \text{H}_2 : b_s \cdots b_2 \end{array} .$$

Daher muß $(b_1, b_2) \in A(G_Z)$ gelten. Wir betrachten T_{j+t} , mit (2.19) und (2.21) erhalten wir T_{j+2t+1} :

$$T_{j+2t+1} : \begin{array}{l} \text{A :} \\ \text{Z : } \pi(n) x_t \cdots x_1 \\ \text{H}_1 : a_r \cdots a_1 n \\ \text{H}_2 : b_s \cdots b_1 \end{array} .$$

Wegen $(n, a_1) \in A(G_R)$ kann n in T_{j+2t+2} weder auf x_1 noch auf b_1 liegen. Wegen $(b_1, b_2) \in A(G_Z)$ kann auch b_1 in T_{j+2t+2} nicht auf x_1 liegen, somit muß in T_{j+2t+2} b_1 auf n liegen, daß heißt, $(b_1, n) \in A(G_A)$, was wiederum $(x_1, n) \in A(G_A)$ widerspricht. Dieser Widerspruch läßt sich nur auflösen, wenn es kein Element b_2 gibt, also $s = 1$ gilt. Für $s = 1$ können wir nun b_1 auf x_1 legen und erhalten damit

$$T_{j+2t+3} : \begin{array}{l} \text{A :} \\ \text{Z : } \pi(n) x_t \cdots x_1 b_1 \\ \text{H}_1 : a_r \cdots a_1 \\ \text{H}_2 : n \end{array} .$$

Wegen $(x_1, n) \in A(G_A)$ und $(n, a_1) \in A(G_Z)$ kann a_1 in T_{j+2t+4} nicht auf n liegen, damit gilt

$$T_{j+2t+5} : \begin{array}{l} \text{A :} \\ \text{Z : } \pi(n) x_t \cdots x_1 b_1 a_1 n \\ \text{H}_1 : a_r \cdots a_2 \\ \text{H}_2 : \end{array} .$$

Daher wissen wir, daß $(a_1, a_2) \in A(G_A)$ gelten muß. Dann muß aber in der Stapelkonfiguration T_{j-t-2} a_1 auf $\pi(n)$ liegen, mit $(a_1, b_1) \in A(G_Z)$ widerspricht das jedoch (2.5). Auch dieser Widerspruch läßt sich nur dann auflösen, wenn es kein Element a_2 gibt, also $r = 1$ gilt.

Mit $r = s = 1$ gilt sofort $T_{j+2t+5} = T_l$ und $T_{j-t-y-3} = T_1$ mit $y = 2$. Somit ist

$$\begin{aligned} G_A &= (\{a_1, b_1, n, \pi(n), x_1, x_t\}, \{(\pi(n), b_1), (b_1, a_1), (a_1, x_t), (x_1, n)\}) \\ &\quad \cup K_{x_t, x_1} \\ G_Z &= (\{a_1, b_1, n, \pi(n), x_1, x_t\}, \{(n, a_1), (a_1, b_1), (b_1, x_1), (x_t, \pi(n))\}) \\ &\quad \cup K_{x_1, x_t} . \end{aligned}$$

Weil P normiert ist, erhalten wir:

$$\pi = [n, 3, 2, n-1, \dots, 4, 1].$$

Wie oben schon gesagt, gilt $l = j + 2t + 5$ und $1 = j - t - 2 - 3$. Wegen $r = s = 1$ gilt $t = n - 4$ und damit ist $l = 3n - 1$. \square

Im nächsten Schritt betrachten wir den Fall (2(d)iiB):

Lemma 2.20 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{b_1, \dots, b_s, x_1, \dots, x_t, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin A(G_R)$ und $(\pi(n), n) \notin A(G_R)$. Für

$$T_j : \begin{array}{l} \text{A :} \\ \text{Z :} \\ \text{H}_1 : n \\ \text{H}_2 : b_s \cdots b_1 \pi(n) x_t \cdots x_1 \end{array} .$$

folgt:

$$\begin{array}{l} \pi = \pi_{P(s,0),s} \sqcup [n, n-1, \dots, s+2, s] \text{ für } s \in \{2, \dots, n-3\} \\ \text{und } \pi = [2, n, n-1, \dots, 3, 1] \text{ oder } \pi = [n, 2, n-1, \dots, 3, 1] \text{ für } s = 1. \end{array}$$

Dabei gilt $l = 3n - 1$.

Beweis: Um von T_j zu T_l gelangen zu können, müssen wir $\pi(n)$ auf den Zielstapel legen. Um jedoch auf $\pi(n)$ zugreifen zu können, müssen wir zunächst die Elemente x_1 bis x_t auf den Hilfsstapel H_1 legen. Wir erhalten damit

$$T_{j+t+1} : \begin{array}{l} \text{A :} \\ \text{Z : } \pi(n) \\ \text{H}_1 : n x_1 \cdots x_t \\ \text{H}_2 : b_s \cdots b_1 \end{array} .$$

Somit gilt sowohl

$$\begin{array}{l} (x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t) \in A(G_R) \\ \text{als auch } (x_t, x_{t-1}), \dots, (x_3, x_2), (x_2, x_1) \in A(G_R). \end{array} \quad (2.22)$$

Wir betrachten die beiden Fälle

1. $(x_1, n) \in A(G_Z)$ und

2. $(x_1, n) \in A(G_A)$.

Es gelte zunächst $(x_1, n) \in A(G_Z)$. Damit folgt aus (2.22) und mit Lemma 2.6

$$\begin{aligned} & (x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t) \in A(G_A) \\ \text{und} & (x_t, x_{t-1}), \dots, (x_3, x_2), (x_2, x_1) \in A(G_Z). \end{aligned}$$

Damit gilt $(x_t, \pi(n)) \in A(G_A)$.

Um von T_1 zu T_j zu gelangen, muß es daher eine Stapelkonfiguration T_g mit $1 < g < j$ geben, die wie folgt aussieht:

$$T_g : \begin{array}{l} A : n \\ Z : \\ H_1 : x_1 \cdots x_t \\ H_2 : b_s \cdots b_1 \pi(n) \end{array} .$$

In T_{g-1} kann x_t nicht auf n liegen, weil bereits $(x_t, \pi(n)) \in A(G_A)$ gilt. $(\pi(n), n) \in A(G_R)$ ist nach Voraussetzung nicht möglich. Da $\pi(n)$ jedoch bereits auf b_1 liegt, gilt nach (2.6) $(\pi(n), x_t) \notin A(G_R)$. Somit kann diese Situation nicht existieren – es bleibt demnach nur der Fall $(x_1, n) \in A(G_A)$ zu untersuchen:

Für $(x_1, n) \in A(G_A)$ folgt $(x_1, x_2) \in A(G_Z)$, mit (2.22) und Lemma 2.6 folgt damit

$$\begin{aligned} & (x_1, x_2), (x_2, x_3), \dots, (x_{t-1}, x_t) \in A(G_Z) \quad (2.23) \\ \text{und} & (x_t, x_{t-1}), \dots, (x_3, x_2), (x_2, x_1) \in A(G_A). \end{aligned}$$

Damit erhalten wir

$$T_{j-t-1} : \begin{array}{l} A : n x_1 \cdots x_t \\ Z : \\ H_1 : \\ H_2 : b_s \cdots b_1 \pi(n) \end{array} .$$

Somit ist $(x_t, \pi(n)) \in A(G_Z)$. Mit $(\pi(n), b_1) \in A(G_A)$ folgt dann nach Lemma 2.5, daß die Kante $(b_1, \pi(n))$ nicht existiert, somit ergibt sich

$$T_{j-t-4} : \begin{array}{l} A : n x_1 \cdots x_t b_1 \pi(n) \\ Z : \\ H_1 : \\ H_2 : b_s \cdots b_2 \end{array} .$$

Mit $(x_t, \pi(n)) \in A(G_Z)$ und (2.23) folgt

$$T_{j+2t+1} : \begin{array}{l} A : \\ Z : \pi(n) x_t \cdots x_1 \\ H_1 : n \\ H_2 : b_s \cdots b_1 \end{array} .$$

Wegen $(b_1, x_t) \in A(G_A)$ folgt für $s \geq 2$ $(b_1, b_2) \in A(G_Z)$. Nach (2.5) kann b_1 damit in T_{j+2t+1} nicht auf n liegen, es folgt:

$$T_{j+2t+5} : \begin{array}{l} A : \\ Z : \pi(n) x_t \cdots x_1 n b_2 b_1 \\ H_1 : \\ H_2 : b_s \cdots b_3 \end{array} .$$

Wenn wir nun Lemma 2.9 auf die Stapelkonfigurationen T_{j-t-4} und T_{j+2t+5} anwenden, erhalten wir für $s \in \{2, \dots, n-3\}$

$$\begin{aligned} G_A &= (\{\pi(n), n, b_1, x_1, x_t, b_3\}, \{(b_3, \pi(n)), (\pi(n), b_1), (b_1, x_t), (x_1, n))\}) \\ &\quad \cup K_{x_t, x_1} \cup W_{b_2, b_s} \\ G_Z &= (\{b_1, b_2, b_4, n, x_1, x_t, \pi(n)\}, \{(b_4, b_1), (b_1, b_2), (b_2, n), (n, x_1), (x_t, \pi(n))\}) \\ &\quad \cup K_{x_1, x_t} \cup W_{b_3, b_s} . \end{aligned}$$

Weil P normiert ist, gilt mit Lemma 2.8 für $s \in \{2, \dots, n-3\}$

$$\pi = \pi_{P(2,s), s-2} \sqcup [s+1, s-2, n, n-1, \dots, s+2, s] = \pi_{P(s,0), s} \sqcup [n, n-1, \dots, s+2, s] .$$

Für $s = 1$ ist $T_1 = T_{j-t-4}$ und daher

$$G_A = (\{\pi(n), n, b_1, x_1, x_t\}, \{(\pi(n), b_1), (b_1, x_t), (x_1, n)\}) \cup K_{x_t, x_1} .$$

Für T_{j+2t+3} ergeben sich die Möglichkeiten

$$T_{j+2t+3} : \begin{array}{l} A : \\ Z : \pi(n) x_t \cdots x_1 n b_1 \\ H_1 : \\ H_2 : \end{array}$$

oder

$$T_{j+2t+3} : \begin{array}{l} A : \\ Z : \pi(n) x_t \cdots x_1 b_1 n \\ H_1 : \\ H_2 : \end{array} ,$$

und damit ist

$$G_Z = (\{\pi(n), n, b_1, x_1, x_t\}, \{(b_1, n), (n, x_1), (x_t, \pi(n))\}) \cup K_{x_1, x_t}$$

oder

$$G_Z = (\{\pi(n), n, b_1, x_1, x_t\}, \{(n, b_1), (b_1, x_1), (x_t, \pi(n))\}) \cup K_{x_1, x_t} .$$

Damit gilt für $s = 1$

$$\pi = [2, n, n-1, \dots, 3, 1] \quad \text{oder} \quad \pi = [n, 2, n-1, \dots, 3, 1].$$

Für $s = 1$ ist $j - t - 4 = 1$ und $l = j + 2t + 3$, wegen $s = 1$ ist $t = n - 3$ und damit $l = 3n - 1$.

Für $s \in \{2, \dots, n-3\}$ gilt nach Lemma 2.9 für s ungerade

$$j - t - 4 = \frac{3}{2}(s - 2 + 1) + 1 \quad \text{und} \quad j + 2t + 5 = l - \frac{3}{2}(s - 2 - 1) - 1$$

und für s geradzahlig

$$j - t - 4 = \frac{3}{2}(s - 2) + 2 \quad \text{und} \quad j + 2t + 5 = l - \frac{3}{2}(s - 2).$$

Mit $s + t = n - 2$ ergibt sich jeweils $l = 3n - 1$. \square

Nun untersuchen wir den Fall (2(d)ii) mit $s = 0$ und $r > 0$ (2(d)iiC):

Lemma 2.21 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{x_1, \dots, x_t, a_1, \dots, a_r, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapelelement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin A(G_R)$ und $(\pi(n), n) \notin A(G_R)$. Für

$$T_j : \begin{array}{l} A : \\ Z : \\ H_1 : a_r \cdots a_1 n \\ H_2 : \pi(n) x_t \cdots x_1 \end{array} .$$

folgt dann

$$\begin{aligned} \pi &= \hat{\pi}^{-1} \text{ mit } \hat{\pi} = \pi_{P(r,0),r} \sqcup [n, n-1, \dots, r+2, r] \text{ für } r \in \{2, \dots, n-3\} \text{ und} \\ \pi &= [n, 1, n-1, \dots, 3, 2] \quad \text{oder} \quad \pi = [n, 2, n-1, \dots, 3, 1] \text{ für } r = 1. \end{aligned}$$

Dabei gilt $l = 3n - 1$.

Beweis: Um, von T_j ausgehend, T_l zu erreichen, müssen wir $\pi(n)$ auf den Zielstapel legen, um jedoch auf $\pi(n)$ zugreifen zu können, müssen die Elemente x_1 bis x_t zunächst auf n auf den Hilfsstapel H_1 gelegt werden. Wir erhalten somit:

$$T_{j+t} : \begin{array}{l} A : \\ Z : \\ H_1 : a_r \cdots a_1 n x_1 \cdots x_t \\ H_2 : \pi(n) \end{array} .$$

Vergleichen wir T_{j+t} mit der Stapelkonfiguration T_j aus Lemma 2.20, so stellen wir fest, daß hier n und $\pi(n)$ vertauscht wurden und sich die Anordnung der Elemente x_1 bis x_t umgekehrt hat. Kehren wir im Beweis von Lemma 2.20 in T_1 und T_l die Reihenfolge der Elemente x_1 bis x_t um, so verändern sich dabei die berechneten Permutationen nicht. Wir können also Ausgangs- und Zielstapel aus Lemma 2.20 übernehmen und – weil n und $\pi(n)$ vertauscht wurden – vertauschen sie. Wir erhalten demnach die inversen Permutationen aus Lemma 2.20. Auch die Anzahl der dazu notwendigen Umstapelungen bleibt offensichtlich genauso groß, wie im Fall (2(d)iiB), nämlich $l = 3n - 1$. \square

Als letzter zu untersuchender Fall verbleibt nun noch (2(d)iiD).

Lemma 2.22 Es sei $P = (G_A, G_Z, G_R, 2)$ ein einfaches, normiertes, induziertes und zweistufig lösbares Stapelproblem mit $n := |V(P)| > 4$, $\pi := \pi(P)$, $\pi(n) \neq n$ und $V(P) = \{x_1, \dots, x_{n-2}, \pi(n), n\}$. $\mathcal{T} = [T_1, \dots, T_l]$ sei eine minimale zweistufige Umstapelfolge und j sei der kleinste Index, für den in der Stapelkonfiguration T_j kein Stapелеlement mehr auf dem Ausgangsstapel liegt. Es gelte $(n, \pi(n)) \notin A(G_R)$ und $(\pi(n), n) \notin A(G_R)$. Für

$$T_j = ((\emptyset, \emptyset), (\emptyset, \emptyset), (\{n\}, \emptyset), (\{x_{n-2}, \pi(n)\}, \{(x_{n-2}, \pi(n))\}) \cup K_{x_1, x_{n-2}}) :$$

$$T_j : \begin{array}{l} \text{A :} \\ \text{Z :} \\ \text{H}_1 : n \\ \text{H}_2 : \pi(n) x_{n-2} \cdots x_1 \end{array} .$$

folgt $\pi = [n, n-1, \dots, 2, 1]$ mit $l = 3n - 1$.

Beweis: Aus der Struktur von T_j erfahren wir, daß

$$\begin{aligned} (x_1, x_2), (x_2, x_3), \dots, (x_{n-3}, x_{n-2}) &\in A(G_R), \\ \text{sowie } (x_{n-2}, \pi(n)) &\in A(G_R) \quad \text{gilt.} \end{aligned} \tag{2.24}$$

Um von T_j zu T_l zu gelangen, müssen wir $\pi(n)$ auf den Zielstapel legen, das ist jedoch erst möglich, wenn die Elemente x_1 bis x_{n-2} auf n auf dem Hilfsstapel H_1 liegen, es gilt also für T_{j+n-1} :

$$T_{j+n-1} : \begin{array}{l} \text{A :} \\ \text{Z : } \pi(n) \\ \text{H}_1 : n x_1 \cdots x_{n-2} \\ \text{H}_2 : \end{array}$$

und damit

$$\begin{aligned} & (x_{n-2}, x_{n-3}), \dots, (x_2, x_1) \in A(G_R), \\ \text{sowie} \quad & (x_1, n) \in A(G_R). \end{aligned} \tag{2.25}$$

Wir untersuchen nun die beiden Fälle

1. $(x_1, n) \in A(G_Z)$ und
2. $(x_1, n) \in A(G_A)$.

Angenommen, es gilt $(x_1, n) \in A(G_Z)$, dann folgt sofort $(x_1, x_2) \in A(G_A)$. Aus (2.24) und (2.25) folgt mit Lemma 2.6 sofort

$$(x_{n-2}, x_{n-3}), \dots, (x_2, x_1) \in A(G_Z). \tag{2.26}$$

Um diesen Bedingungen zu genügen, muß $(n, \pi(n)) \in A(G_Z)$ gelten. Das widerspricht jedoch unserer Voraussetzung, somit ist die Annahme falsch, es kann nur $(x_1, n) \in A(G_A)$ gelten.

Im Fall $(x_1, n) \in A(G_A)$ folgt sofort $(x_1, x_2) \in A(G_Z)$. Aus (2.24) und (2.25) folgt dann mit Lemma 2.6 sofort

$$\begin{aligned} & (x_1, x_2), (x_2, x_3), \dots, (x_{n-3}, x_{n-2}) \in A(G_Z), \\ & (x_{n-2}, x_{n-3}), \dots, (x_3, x_2), (x_2, x_1) \in A(G_A) \\ \text{und} \quad & (x_{n-2}, \pi(n)) \in A(G_Z). \end{aligned} \tag{2.27}$$

Damit erhalten wir sofort

$$\begin{aligned} & A : \\ T_{j+2(n-2)+2} : & \quad Z : \pi(n) x_{n-2} \cdots x_1 n \\ & \quad H_1 : \\ & \quad H_2 : \end{aligned}$$

und

$$\begin{aligned} & A : n x_1 \cdots x_{n-2} \pi(n) \\ T_{j-n} : & \quad Z : \\ & \quad H_1 : \\ & \quad H_2 : \end{aligned} ,$$

damit ist

$$\begin{aligned} G_A &= (\{\pi(n), n, x_1, x_{n-2}\}, \{(\pi(n), x_{n-2}), (x_1, n)\}) \cup K_{x_{n-2}, x_1} \\ \text{und } G_Z &= (\{\pi(n), n, x_1, x_{n-2}\}, \{(n, x_1), (x_{n-2}, \pi(n))\}) \cup K_{x_1, x_{n-2}}. \end{aligned}$$

Da P ein normiertes Problem ist, erhalten wir

$$\pi = [n, n-1, \dots, 2, 1].$$

Wegen $T_1 = T_{j-n}$ und $T_l = T_{j+2(n-2)+2}$ folgt $l = 3n - 1$. \square

2.3.5 Einfache Stapelprobleme sind leicht zu lösen

Definition 2.20 Wir vereinbaren nun Bezeichnungen für die in den Lemmata 2.10 bis 2.22 gefundenen Permutationen:

$$\begin{aligned}
\pi_1 &= [n-2, n-3, \dots, 2, 1, n-1, n], & \pi_2 &= [n-1, n-2, n-3, \dots, 2, 1, n], \\
\pi_3 &= [n-2, \dots, 1, n, n-1], & \pi_4 &= [2, n, n-1, 1, n-2, \dots, 3], \\
\pi_5 &= [n, 2, n-1, 1, n-2, \dots, 3], & \pi_6 &= [n, n-1, 1, n-2, \dots, 2], \\
\pi_7 &= [n-1, n, n-2, \dots, 1], & \pi_8 &= [4, 1, n, n-1, \dots, 5, 3, 2], \\
\pi_9 &= [4, 2, n, n-1, \dots, 5, 3, 1], & \pi_{10} &= [3, n, n-1, \dots, 4, 2, 1], \\
\pi_{11} &= [n, n-1, n-2, \dots, 3, 1, 2], & \pi_{12} &= [n, n-1, \dots, 1], \\
\pi_{13} &= [n, 3, 2, n-1, \dots, 4, 1], & \pi_{14} &= [2, n, n-1, \dots, 3, 1], \\
\pi_{15} &= [n, 2, n-1, \dots, 3, 1], & \pi_{16} &= [n, 1, n-1, \dots, 3, 2], \\
\pi_{17} &= [n-1, 1, n, n-2, \dots, 2], & \pi_{18} &= [2, n, n-1, \dots, 4, 1, 3],
\end{aligned}$$

Weiterhin:

$$\begin{aligned}
\pi_{19} &= \pi_{P(2,n),n-2} \sqcup [n, n-2], \\
\pi_{20} &= \pi_{P(1,n),n-1} \sqcup [n-1], \\
\pi_{21} &= \pi_{P(n,2),n-4} \sqcup [n, n-1, n-4, n-2], \\
\pi_{20+i} &= \pi_{P(n-i,1),n-i-3} \sqcup [n-1, n-i-3, n, n-2, \dots, n-i-1], \\
&\quad \text{für } i \in \{2, \dots, n-4\}, \\
\pi_{15+n+i} &= \pi_{P(n-i,1),n-i-3} \sqcup [n, n-1, n-i-3, n-2, \dots, n-i-1], \\
&\quad \text{für } i \in \{2, \dots, n-5\}, \\
\pi_{9+2n+i} &= \pi_{P(n-1,i),n-i-1} \sqcup [n-1, n-2, \dots, n-i+1, n-i-1, n] \\
&\quad \text{für } i \in \{2, \dots, n-2\}, \\
\pi_{6+3n+i} &= \pi_{P(i,0),i} \sqcup [n, n-1, \dots, i+2, i], \quad \text{für } i \in \{2, \dots, n-3\}, \\
\pi_{1+4n+i} &= \pi_{20+i}^{-1}, \quad \text{für } i \in \{3, \dots, n-4\}, \\
\pi_{5n-4+i} &= \pi_{9+2n+i}^{-1}, \quad \text{für } i \in \{2, \dots, n-2\}, \\
\pi_{6n-7+i} &= \pi_{15+n+i}^{-1}, \quad \text{für } i \in \{2, \dots, n-5\}, \\
\pi_{7n-13+i} &= \pi_{6+3n+i}^{-1}, \quad \text{für } i \in \{2, \dots, n-3\}.
\end{aligned}$$

Wie bereits im Anschluß an Lemma 2.17 erwähnt, gibt es Permutationen, die mehrmals auftauchen. In Definition 2.20 treten diese Permutationen natürlich nur einmal auf.

Satz 2.23 Für $n > 4$ gilt: $L_2(\Pi_n) = \{\pi_1, \dots, \pi_{8n-16}\}$.

Beweis: $L_2(\Pi_n) \subseteq \{\pi_1, \dots, \pi_{8n-16}\}$: Es sei $\pi \in L_2(\Pi_n)$, das heißt, es existiert ein zweistufiges lösbares Problem P , für das $\pi = \pi(P)$ gilt.

Da wir in den Lemmata 2.10 bis 2.22 alle möglichen Fälle für zweistufig lösbare Probleme untersucht haben, gleicht π damit einer der dabei gefundenen Permutationen, somit gilt $\pi \in \{\pi_1, \dots, \pi_{8n-16}\}$.

$\{\pi_1, \dots, \pi_{8n-16}\} \subseteq L_2(\Pi_n)$ Sei $\pi \in \{\pi_1, \dots, \pi_{8n-16}\}$, dann haben wir in mindestens einem der Lemmata 2.10 bis 2.22 gezeigt, daß es ein zweistufig lösbares Problem $P \in \mathcal{P}_{n,2}$ gibt, für das $\pi = \pi(P)$ gilt. Damit ist $\pi \in L_2(\Pi_n)$. \square

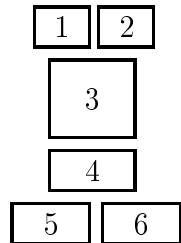
Bemerkung: Mit Satz 2.23 erfahren wir nicht nur die genaue Anzahl der Zielstapel von zweistufig lösbaren Problemen ($8n - 16$), für eine feste Anzahl n von Stapелеlementen, sondern auch deren genaue Struktur. Aus den Lemmata 2.10 bis 2.22 geht außerdem hervor, daß zur Umstapelung nur eine lineare Anzahl an Operationen ($5n - 6$, $5n - 5$ bzw. $3n - 1$) notwendig ist.

Kapitel 3

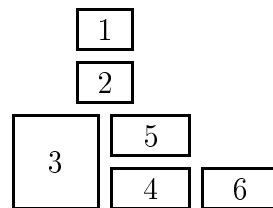
Verzweigte Stapel

Wir trennen uns nun von der simplen „Geradeausstruktur“ der Stapel und wenden uns Stapeln zu, bei denen ein Element auf mehreren anderen liegen darf, bzw. mehrere Stapелеlemente auf einem einzigen liegen dürfen. Folgende Ausgangs- und Zielsituationen sind nun beispielsweise zulässig:

Ausgangssituation:



gewünschte Zielsituation:



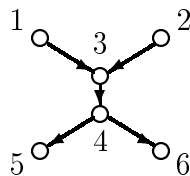
Wieder wählen wir gerichtete Graphen als mathematische Repräsentation für diese Stapel. Diesmal richten wir jedoch die Kanten des Zielstapels von „unten nach oben“:

$$Q = (G_A, G_Z, k)$$

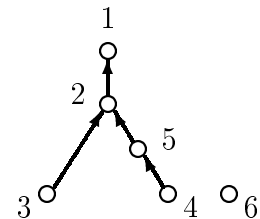
$$G_A = (\{1, 2, 3, 4, 5, 6\}, \{(1, 3), (2, 3), (3, 4), (4, 5), (4, 6)\})$$

$$\text{und } G_Z = (\{1, 2, 3, 4, 5, 6\}, \{(4, 5), (3, 2), (5, 2), (2, 1)\})$$

Ausgangsstapel G_A :



Zielstapel G_Z :



Hier stellt sich nun die Frage nach der kleinsten Anzahl k an Hilfsstapelplätzen,

für die das Stapelproblem $Q = (G_A, G_Z, k)$ lösbar ist. Das entspricht der Anzahl der Stapelelemente, die ausgestapelt werden müssen, damit Q gelöst werden kann. Die Hilfsstapelanzahl erfährt hier keine Beschränkung.

Die kleinste Anzahl an Ausstapelungen, für die Q lösbar bleibt, ist $k = 2$. Eine Umstapelung:

$$S_1 \xrightarrow{1} S_2 \xrightarrow{2} S_3 \xrightarrow{3} S_4 \xrightarrow{4} S_5 \xrightarrow{5} S_6 \xrightarrow{2} S_7 \xrightarrow{1} S_8 \xrightarrow{6} S_9,$$

mit $S_1 = (G_A, (\emptyset, \emptyset), \emptyset, \emptyset)$ und $S_9 = ((\emptyset, \emptyset), G_Z, \emptyset, \{1, 2\})$.

Definition 3.1 $Q = (G_A, G_Z, k)$ heißt genau dann *verzweigtes Stapelproblem*, wenn $G_A = (V, A_A)$ und $G_Z = (V, A_Z)$ verzweigte Stapel sind und k eine natürliche Zahl ist. G_A nennen wir *Ausgangsstapel*, G_Z nennen wir *Zielstapel*; k gibt die Anzahl der *Hilfsstapelplätze* an. Die Knoten v aus V nennen wir *Stapelelemente*.

Definition 3.2 Ein Stapel $G = (V, A)$ ist genau dann von der Stapelhöhe h , wenn der längste einfache gerichtete Pfad in G die Länge h hat.

Ein verzweigtes Stapelproblem $Q = (G_A, G_Z, k)$ ist genau dann von der *Stapelhöhe* h , wenn

$$h = \max\{h_A, h_Z\}$$

gilt, wobei h_A die Stapelhöhe von G_A und h_Z die Stapelhöhe von G_Z ist.

Definition 3.3 $S = (G_1, G_2, H, M)$ heißt genau dann *Stapelkonfiguration eines verzweigten Stapelproblems* $Q = (G_A, G_Z, k)$, wenn gilt:

- i) $G_1 = (V_1, A_1)$ ist der von V_1 in G_A induzierte Untergraph und $G_2 = (V_2, A_2)$ ist der von V_2 in G_Z induzierte Untergraph.
- ii) $H \subseteq M \subset V(G_A)$, $|M| \leq k$,
- iii) $H \cup V_1 \cup V_2 = V(G_A)$ und
- iv) H , V_1 und V_2 sind paarweise disjunkt.

Eine solche Stapelkonfiguration S repräsentiert die Zustände des Ausgangs- und des Zielstapelplatzes G_1 und G_2 und gibt die Menge H aller gegenwärtig ausgestapelten Elemente an, die auf den k Hilfsstapelplätzen liegen. Dabei darf auf jedem Hilfsstapelplatz nur ein Stapелеlement liegen. M ist die Menge aller Elemente, die bisher auf einen Hilfsstapelplatz gelegt wurden. Jedem Hilfsstapelplatz wird somit genau ein Stapелеlement zugewiesen – kein anderes Element darf mehr auf diesem Platz liegen ($|M| \leq k$ nach Definition 3.3).

Wird vom Ausgangsstapel ein Element weggenommen und auf einen anderen Stapel gelegt oder von einem der Hilfsstapelplätze ein Element auf den Zielstapel gebracht, so ergibt sich eine neue Stapelkonfiguration. Diesen Vorgang nennen wir *Umstapelung*.

Definition 3.4 Gegeben seien 2 Stapelkonfigurationen $S = (G_1, G_2, H, M)$ und $S' = (G'_1, G'_2, H', M')$ eines verzweigten Stapelproblems $Q = (G_A, G_Z, k)$. S' heißt genau dann *Umstapelung von S* , wenn ein Stapелеlement $v \in V(G_A)$ existiert, für das eine der folgenden Bedingungen erfüllt ist:

- i) $d_{G_1}^-(v) = 0$, $H' = H$, $M' = M$, $G'_2 = G_2 \cup (\{v\}, \{(u, v) : (u, v) \in A(G_Z)\})$,
 $G'_1 = G_1 \setminus \{v\}$ und für alle $(u, v) \in A(G_Z) \Rightarrow u \in V(G_2)$,
- ii) $d_{G_1}^-(v) = 0$, $H' = H \cup \{v\}$, $G'_1 = G_1 \setminus \{v\}$, $G'_2 = G_2$, $M' = M \cup \{v\}$,
- iii) $v \in H$, $H' = H \setminus \{v\}$, $M' = M$, $G'_2 = G_2 \cup (\{v\}, \{(u, v) : (u, v) \in A(G_Z)\})$,
 $G'_1 = G_1$ und für alle $(u, v) \in A(G_Z) \Rightarrow u \in V(G_2)$.

Ist S' Umstapelung von S , so schreiben wir auch $S \xrightarrow{v} S'$. Wird v dabei auf den Zielstapelplatz gelegt, dann nennen wir diesen Vorgang auch *Einstapelung*. Wird v auf einen Hilfsstapelplatz gelegt, dann nennen wir diesen Vorgang *Ausstapelung*.

An dieser Stelle sei auf die Veränderung des Modells bezüglich des Zielstapels gegenüber dem Modell für einfache Stapelprobleme hingewiesen. Aus Definition 3.4 geht hervor, daß bei der Modellierung von verzweigten Stapelproblemen die Kanten des Zielstapels entgegengesetzt denen des Ausgangsstapels gerichtet sind. Im Ausgangsstapel sind die Kanten von „oben nach unten“ gerichtet – im Zielstapel von „unten nach oben“. Zur Illustration dient das Beispiel auf Seite 65.

Definition 3.5 Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes Stapelproblem. Eine Folge $\mathcal{S} = [S_1, \dots, S_l]$ von Stapelkonfigurationen $S_i = (G_1^i, G_2^i, H^i, M^i)$ heißt genau dann *zulässige Stapelfolge von Q* , wenn gilt:

- i) $G_1^1 = G_A$, $G_2^1 = (\emptyset, \emptyset)$, $M^1 = \emptyset$ und
- ii) für $i \in \{1, \dots, l-1\}$ ist S_{i+1} Umstapelung von S_i .

Die Stapelkonfigurationen S_i nennen wir dann zulässige Stapelkonfigurationen von Q .

Definition 3.6 Ein verzweigtes Stapelproblem $Q = (G_A, G_Z, k)$ heißt genau dann *lösbar*, wenn eine zulässige Stapelfolge $\mathcal{S} = [S_1, \dots, S_l]$ existiert, so daß für $S_l = (G_1^l, G_2^l, H^l, M^l)$

$$G_2^l = G_Z$$

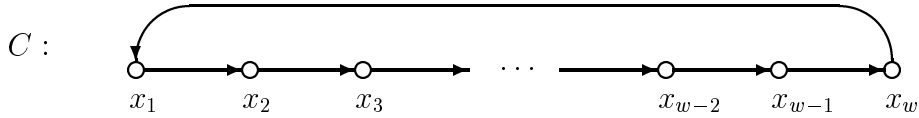
gilt. \mathcal{S} heißt dann *l -Umstapelfolge für Q* .

3.1 Lösbarkeit von verzweigten Stapelproblemen

Im folgenden werden wir die Struktur des Problems näher untersuchen, besonders interessiert uns dabei ein graphentheoretisches Kriterium zur Lösbarkeit verzweigter Stapelprobleme.

Lemma 3.1 Es sei $Q = (G_A, G_Z, k)$ ein lösbares, verzweigtes Stapelproblem. Dann folgt: Für jeden Kreis C in $G_A \cup G_Z$ gibt es einen Knoten x_C mit (x_C, y_C) in C und $(x_C, y_C) \in A(G_A)$, der zur Lösung von Q ausgestapelt werden muß.

Beweis: Wir betrachten einen Kreis C in $G_A \cup G_Z$:



Angenommen, keiner der Knoten x_1, x_2, \dots, x_w muß zur Lösung von Q ausgestapelt werden. Um das Element x_1 auf den Zielstapel zu bringen, muß x_w vor x_1 vom Ausgangsstapel genommen werden, da auch im Fall von $(x_w, x_1) \in A(G_Z)$ x_w nicht ausgestapelt werden darf. Bevor jedoch x_w auf den Zielstapelplatz gebracht werden darf, muß x_{w-1} vom Ausgangsstapel entnommen werden. Setzen wir diese Argumentation fort, so erhalten wir, daß x_1 vor x_2 , x_2 vor x_3 , \dots , x_{w-1} vor x_w und x_w vor x_1 vom Ausgangsstapel entnommen werden muß. Da aber nicht gleichzeitig x_1 vor x_w und x_w vor x_1 vom Ausgangsstapel entnommen werden kann, muß ein Knoten $x_i \in \{x_1, \dots, x_w\}$ existieren, durch dessen Ausstapelung der Kreis C aufgeschnitten wird. Da durch Ausstapelungen aber nur Kanten aus G_A gelöscht werden, folgt $(x_i, x_{i+1}) \in A(G_A)$. \square

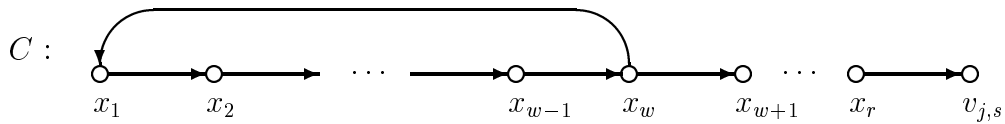
Satz 3.2 Ein verzweigtes Stapelproblem $Q = (G_A, G_Z, k)$ ist genau dann lösbar, wenn einfache Pfade $P_1 = [v_{1,1}, \dots, v_{1,i_1}]$, $P_2 = [v_{2,1}, \dots, v_{2,i_2}]$, \dots , $P_t = [v_{t,1}, \dots, v_{t,i_t}]$ mit $t \leq k$ in G_A existieren, für die folgende Bedingungen erfüllt sind:

- i) Für alle $j \in \{1, \dots, t\}$ liegt $v_{j,1}$ auf der Spitze von G_A ($d_{G_A}^-(v_{j,1}) = 0$),
- ii) $G_Z \cup (G_A \setminus A(G_A(N_{G_A}[\{v_{1,i_1}, \dots, v_{t,i_t}\}])))$ ist kreisfrei und
- iii) für alle $j \in \{1, \dots, t\}$ gilt: In $G_Z \cup (G_A \setminus A(G_A(N_{G_A}[\{v_{1,i_1}, \dots, v_{j-1,i_{j-1}}\}])))$ führt kein Weg von einem Kreis C zu einem der Knoten $v_{j,1}, \dots, v_{j,i_j}$ und diese Knoten treten auch selbst in C nicht auf.

Beweis: („ \Rightarrow “): Es sei $Q = (G_A, G_Z, k)$ ein lösbares, verzweigtes Stapelproblem. Dann existiert eine minimale Menge $V' \subseteq V(G_A)$ mit $|V'| \leq k$ von Knoten, die zur Lösung von Q ausgestapelt werden müssen. Weiterhin gibt es eine zugehörige minimale Umstapelfolge \mathcal{S} , in der die Knoten u_1, \dots, u_t aus V' in dieser Reihenfolge ausgestapelt werden. Damit existieren Pfade P_1, \dots, P_t in G_A , die zu den Knoten u_1, \dots, u_t führen. Damit ist Punkt (1) trivialerweise erfüllt.

Zeigen wir nun Punkt (2). Angenommen, $G_Z \cup (G_A \setminus A(G_A(N_{G_A}[V'])))$ ist nicht kreisfrei. Dann existiert ein Kreis $C = [x_1, \dots, x_w, x_{w+1} = x_1]$. Nach Lemma 3.1 existiert ein Knoten $x_j \in V(C)$ mit $(x_j, x_{j+1}) \in A(G_A)$, der auf einen Hilfsstapelplatz gebracht werden muß. Damit ist x_j auch Element von V' – das widerspricht jedoch der Wahl von C , weil in C keine ausgehenden Kante aus G_A eines Knotens aus V' auftreten können. Somit ist die Annahme falsch.

Kommen wir nun zu Punkt (3): Wir betrachten eine minimale Umstapelfolge \mathcal{S} von Q , in der die auszustapelnden Elemente in der Reihenfolge u_1, u_2, \dots, u_t vom Ausgangsstapelplatz auf Hilfsstapelplätze gelegt werden. Es seien bereits die Knoten u_1, u_2, \dots, u_{j-1} ausgestapelt worden. Entsprechend der Umstapelfolge \mathcal{S} wollen wir nun u_j auf einen Hilfsstapelplatz bringen. Wir nehmen an, es existiert ein Knoten $v_{j,s}$ aus P_j und ein Kreis C in $G_Z \cup (G_A \setminus A(G_A(N_{G_A}[\{u_1, \dots, u_{j-1}\}])))$ so daß ein Pfad von C nach $v_{j,s}$ führt. Um u_j ausstapeln zu können, müssen die Knoten $v_{j,1}, v_{j,2}, \dots, v_{j,s}, \dots, v_{j,i_j-1}$ des Pfades P_j vom Ausgangsstapelplatz auf den Zielstapelplatz gebracht werden.



Bevor $v_{j,s}$ jedoch vom Ausgangsstapelplatz auf den Zielstapelplatz gebracht werden kann, müssen die Elemente x_1, \dots, x_r dorthin gelegt werden. Nach Lemma 3.1 muß einer der Knoten x_1, \dots, x_w ausgestapelt werden. Da jedoch u_j der nächste auszustapelnde Knoten sein soll, existiert ein $x_i \in \{x_1, \dots, x_w\}$ mit $x_i = u_j$. Das bedeutet jedoch, daß u_j ausgestapelt wird, während $v_{j,s}$ noch auf dem Ausgangsstapelplatz liegt. Das ist nicht möglich, da in G_A ein Pfad von $v_{j,s}$ nach u_j führt (P_j).

Der Beweis für den Fall $v_{j,s} \in V(C)$ verläuft analog (da P_j ein einfacher Pfad ist, gilt $v_{j,s} \neq u_j$). Damit ist die Annahme falsch.

(„ \Leftarrow “): Angenommen, das Stapelproblem Q sei nicht lösbar, dann existiert eine Stapelfolge $\mathcal{S} = [S_1, \dots, S_l]$ maximaler Länge von Q . Maximal soll in diesem Sinn bedeuten, daß eine mögliche direkte Einstapelung in den Zielstapel einer Ausstapelung vorgezogen wird und l maximal ist. Da \mathcal{S} maximal und Q nicht lösbar ist, gilt für $S_l = (G_1^l, G_2^l, H^l, M^l)$:

$$G_2^l \neq G_Z$$

und es gibt keine Stapelkonfiguration S_{l+1} , die Umstapelung von S_l ist. Es läßt sich demnach kein Element mehr vom Ausgangsstapelplatz oder den Hilfsstapelplätzen auf den Zielstapelplatz bringen. Wir führen folgende Bezeichnungen ein:

$$V_l := \{v : d_{G_1^l}^-(v) = 0\} \cup H^l \quad \text{und} \quad U_l := \{v : d_{G_1^l}^-(v) > 0\}.$$

Die Stapelelemente aus V_l sind demnach diejenigen, auf die sofort zugegriffen werden kann. (Die Spitze von G_1^l und die Elemente, die in S^l auf den Hilfsstapelplätzen liegen.) Da nun keine Knoten aus V_l mehr auf den Zielstapelplatz gebracht werden können, gilt:

Für alle $v \in V_l$ existiert ein $u \in U_l$, so daß in G_Z ein Pfad von u nach v führt.

Das heißt, bevor irgendein v aus V_l auf den Zielstapelplatz gelegt werden kann, muß ein $u \in U_l$ dorthin gebracht werden. Andererseits wissen wir nach Definition von U_l :

Für alle $u \in U_l$ existiert ein $v \in V_l$, so daß in G_1^l ein Pfad von v nach u führt.

Zu jedem Knoten aus U_l gibt es von einem Knoten aus V_l einen Weg in $G_1^l \cup G_Z$. Ebenso gibt es zu jedem Knoten aus V_l von einem Knoten aus U_l einen Weg in $G_1^l \cup G_Z$. Da U_l und V_l aber endliche Mengen sind, muß $G_1^l \cup G_Z$ einen Kreis C enthalten. Nach Punkt (2) ist $G_Z \cup (G_A \setminus A(G_A(N_{G_A}[\{v_{1,i_1}, \dots, v_{t,i_t}\}])))$ kreisfrei, somit existiert ein $v_{r,i_r} \in \{v_{1,i_1}, \dots, v_{t,i_t}\}$ mit $v_{r,i_r} \in V(G_1^l)$. Wir wählen dieses v_{r,i_r} so, daß r der kleinstmögliche Index ist, also für alle v_{j,i_j} mit $j < r$ $v_{j,i_j} \notin V(G_1^l)$ gilt. Wenn die Knoten $v_{1,i_1}, \dots, v_{r-1,i_{r-1}}$ bereits aus dem Ausgangsstapel entfernt wurden, gilt offensichtlich

$$G_1^l \subseteq G_A \setminus A(G_A(N_{G_A}[\{v_{1,i_1}, \dots, v_{r-1,i_{r-1}}\}]})), \quad (3.1)$$

da aus G_1^l mehr Knoten als $v_{1,i_1}, \dots, v_{r-1,i_{r-1}}$ entfernt werden konnten.

Nach Voraussetzung existiert ein Pfad $P_r = [v_{r,1}, \dots, v_{r,i_r}]$ in G_A wobei $v_{r,1}$ auf der Spitze von G_A liegt ($d_{G_A}^-(v_{r,1}) = 0$). Dieser Pfad muß demnach einen Knoten v der Spitze von G_1^l enthalten. Es gilt demnach $v \in V_l$, damit existiert ein $u \in U_l$, so daß in $G_Z \cup G_1^l$ ein Pfad R_1 von u nach v führt. Für u wiederum gibt es ein $v^1 \in V_l$, so daß in $G_Z \cup G_1^l$ ein Pfad R_2 von v^1 nach u führt. Für v^1 existiert wiederum ein $u^1 \in U_l$ für welches es einen Pfad R_3 von u^1 nach v^1 in $G_Z \cup G_1^l$ gibt. Diese Argumentation läßt sich beliebig lang fortsetzen. Da V_l und U_l jedoch endliche Mengen sind, gibt es in diesem Pfad $R_1 \cup R_2 \cup R_3 \cup \dots$ einen Kreis C . Somit liegt entweder v in diesem Kreis oder es führt ein Pfad von C zu v :

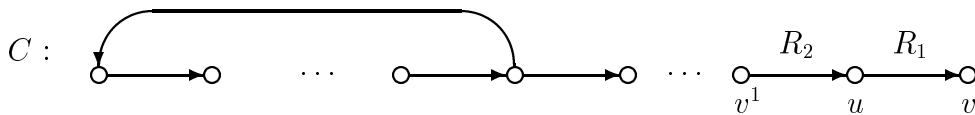


Abb. 3.1:

C liegt in $G_1^l \cup G_Z$. Wegen (3.1) liegt C auch in $G_Z \cup (G_A \setminus A(G_A(N_{G_A}[\{v_{1,i_1}, \dots, v_{r,i_{r-1}}\}])))$. Mit Abbildung 3.1 widerspricht das der Voraussetzung (3). Somit ist die Annahme falsch – Q muß lösbar sein. \square

Betrachten wir nun Stapelprobleme ohne Hilfsstapelplätze in bezug auf ihre Lösbarkeit. Es gilt:

Satz 3.3 *Ein verzweigtes Stapelproblem $Q = (G_A, G_Z, 0)$ ist genau dann lösbar, wenn $G_A \cup G_Z$ kreisfrei ist.*

Beweis: („ \Rightarrow “) Folgt unmittelbar aus Lemma 3.1.

(„ \Leftarrow “) Ergibt sich aus Satz 3.2. \square

3.2 NP-Vollständigkeit

In diesem Abschnitt werden wir die NP-Vollständigkeit des Problems nachweisen. Zu allgemeinen Begriffen der Kompliziertheitstheorie verweisen wir auf das Buch von GAREY und JOHNSON [1].

Definition 3.7 Wir definieren die folgenden Problemklassen. SP sei die Menge aller lösbaren verzweigten Stapelprobleme und SP-2 die Menge aller lösbaren verzweigten Stapelprobleme der Höhe 2:

$$\text{SP-2} := \{ Q = (G_A, G_Z, k) \text{ lösbares verzweigtes} \\ \text{Stapelproblem der Höhe 2} \}$$

$$\text{SP} := \{ Q = (G_A, G_Z, k) \text{ lösbares} \\ \text{verzweigtes Stapelproblem} \}$$

$$\text{FEEDBACK-VERTEX-SET} := \{ (G, k) : G = (V, E) \text{ ist gerichteter Graph,} \\ V' \subseteq V, |V'| \leq k, \text{ so daß } V' \text{ mindestens} \\ \text{einen Knoten jeden Kreises von } G \text{ enthält.} \}$$

Es stellt sich heraus, daß die Klasse SP, ja sogar SP-2 NP-vollständig sind:

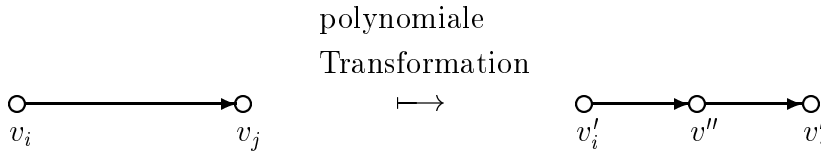
Satz 3.4 (SCHMIEDEL) *SP-2 ist NP-vollständig.*

Um Satz 3.4 zu beweisen, führen wir eine Reduktion auf das Problem FEEDBACK-VERTEX-SET durch, das nach GAREY und JOHNSON [1] NP-vollständig ist.

Beweis: Wir haben zunächst zu zeigen, daß $\text{SP-2} \in \text{NP}$ gilt:

Für ein gegebenes lösbares verzweigtes Stapelproblem $Q = (G_A, G_Z, k)$ der Höhe 2 kann eine nichtdeterministische Turingmaschine in Polynomialzeit eine Umstapelfolge \mathcal{S} berechnen, bei der nur k Hilfsstapelplätze benötigt werden.

Nun zeigen wir, daß FEEDBACK-VERTEX-SET polynomial auf SP-2 reduziert werden kann. Es sei $G = (V, E)$ ein gerichteter Graph und k eine natürliche Zahl, so daß $(G, k) \in \text{FEEDBACK-VERTEX-SET}$ gilt. Wir ordnen nun jedem Knoten $v \in V$ einen neuen Knoten v' zu. Weiterhin ordnen wir jeder Kante $(v_i, v_j) \in E$ zwei Kanten (v'_i, v'') und (v'', v'_j) zu, wobei v'' wiederum ein neuer Knoten ist.



Wir definieren folgende Mengen:

$$N'' := \{v'' : (v_i, v_j) \in E\}, N' := \{v' : v \in V\} \text{ und } N := N' \cup N'', \text{ sowie} \\ E_1 := \{(v'_i, v'') : (v_i, v_j) \in E\} \text{ und } E_2 := \{(v'', v'_j) : (v_i, v_j) \in E\}.$$

Diese Transformationen sind offensichtlich in Polynomialzeit durchführbar. Nun sei $V' \subseteq V$ mit $|V'| \leq k$ eine Knotenmenge, so daß $G \setminus V'$ kreisfrei ist. Genau dann ist $Q = ((N, E_1), (N, E_2), k) \in \text{SP-2}$.

Begründung: U' sei die Menge der Knoten, die durch die Transformation den Knoten aus V' zugeordnet wurden. Q ist nach Definition von N, E_1 und E_2 ein verzweigtes Stapelproblem und alle Knoten aus U' liegen auf der Spitze von (N, E_1) und $|U'| \leq k$. $(N, E_2 \cup E_1 \setminus \{(u, v) : \forall u \in U', \forall v : (u, v) \in E_1\})$ ist kreisfrei. Nach Satz 3.2 ist Q lösbar.

Andererseits besitzt nach Satz 3.2 jedes lösbare verzweigte Stapelproblem $Q = ((N, E_1), (N, E_2), k)$ eine Knotenmenge $U' \subseteq N$ mit $|U'| \leq k$, so daß $(N, E_2 \cup E_1 \setminus \{(u, v) : \forall u \in U', \forall v : (u, v) \in E_1\})$ kreisfrei ist. Es sei V' die Menge ursprünglicher Knoten, denen durch die Transformation die Knoten aus U' zugeordnet wurden. Dann gilt offensichtlich: $G \setminus V'$ ist kreisfrei.

Somit ist FEEDBACK-VERTEX-SET polynomial auf SP-2 reduzierbar. \square

Satz 3.5 *SP ist NP-vollständig.*

Beweis: Eine nichtdeterministische Turingmaschine wird auch für ein lösbares verzweigtes Stapelproblem $Q = (G_A, G_Z, k)$ beliebiger Höhe in Polynomialzeit eine Umstapelfolge \mathcal{S} raten, bei der nur k Hilfsstapelplätze benötigt werden. Somit gilt $\text{SP} \in \text{NP}$.

Es sei $G_A = (V, A_A)$ und $G_Z = (V, A_Z)$. Um zu zeigen, daß SP NP-vollständig ist, reduzieren wir SP-2 auf SP. Es sei $Q = (G_A, G_Z, k) \in \text{SP-2}$. Wir führen folgende – offensichtlich polynomiale – Transformation durch:

Für jede Kante $(x, y) \in A(G_A)$ fügen wir Knoten x_1, x_2, \dots, x_{r_x} ein, so daß $(x, x_1), (x_1, x_2), \dots, (x_{r_x}, y) \in A(G'_A)$ gilt. Gleichzeitig fügen wir diese neuen

Knoten als isolierte Knoten zu G_Z hinzu:

Für $\begin{array}{c} \circ \longrightarrow \circ \\ x \qquad y \end{array}$ in G_A , folgt $\begin{array}{c} \circ \longrightarrow \circ \longrightarrow \circ \longrightarrow \cdots \longrightarrow \circ \longrightarrow \circ \\ x \quad x_1 \quad x_2 \qquad \qquad \qquad x_{r_x} \quad y \end{array}$ in G'_A .

$$\begin{aligned} G'_A = (V', A'_A) : \quad V' &:= V \cup \bigcup_{(x,y) \in A_A} \{x_1, \dots, x_{r_x}\}, \\ A'_A &:= A_A \cup \bigcup_{(x,y) \in A_A} \{(x, x_1), \dots, (x_{r_x}, y)\}, \\ G'_Z = (V', A'_Z) : \quad A'_Z &:= A_Z. \end{aligned}$$

Analog verändern wir G_Z (jetzt G'_Z):

Für $\begin{array}{c} \circ \longrightarrow \circ \\ x \qquad y \end{array}$ in G'_Z , folgt $\begin{array}{c} \circ \longrightarrow \circ \longrightarrow \circ \longrightarrow \cdots \longrightarrow \circ \longrightarrow \circ \\ x \quad x_1 \quad x_2 \qquad \qquad \qquad x_{r_x} \quad y \end{array}$ in G''_Z .

$$\begin{aligned} G''_Z = (V'', A''_Z) : \quad V'' &:= V' \cup \bigcup_{(x,y) \in A'_Z} \{x_1, \dots, x_{r_x}\}, \\ A''_Z &:= A'_Z \cup \bigcup_{(x,y) \in A'_Z} \{(x, x_1), \dots, (x_{r_x}, y)\}, \\ G''_A = (V'', A''_A) : \quad A''_A &:= A'_A. \end{aligned}$$

Wir zeigen nun, daß $Q'' = (G''_A, G''_Z, k) \in \text{SP}$ ist. Wenn wir eine Lösung für Q haben, dann ist das ebenfalls eine Lösung für Q'' .

Begründung: Stößt man beim Ausstapeln von G_A auf „neue“ Stapelelemente, so können diese sofort auf den Zielstapel gebracht werden, weil diese Knoten in G_Z isolierte Knoten sind. Müssen beim Einstapeln in den Zielstapel neue Stapelelemente gelegt werden, so stehen diese auf dem Ausgangsstapel sofort zur Verfügung, da diese Knoten dort isoliert sind. Aus dem gleichen Grund ist eine Lösung von Q'' auch eine Lösung von Q . \square

3.3 Lösungsstrategien

3.3.1 Obere und untere Schranken

Durch Satz 3.3 haben wir erfahren, daß Kreise in der Vereinigung aus Ausgangs- und Zielgraph dafür verantwortlich sind, daß Elemente auf Hilfsstapelplätze gebracht werden müssen. Sehen wir uns nun die Struktur dieser Kreise etwas genauer an: Da sowohl G_A als auch G_Z kreisfrei sind, muß jeder Kreis C in $G_A \cup G_Z$ Kanten aus G_A und auch aus G_Z enthalten. In einem solchen Kreis C finden sich demnach abwechselnd Pfade aus G_A und G_Z :

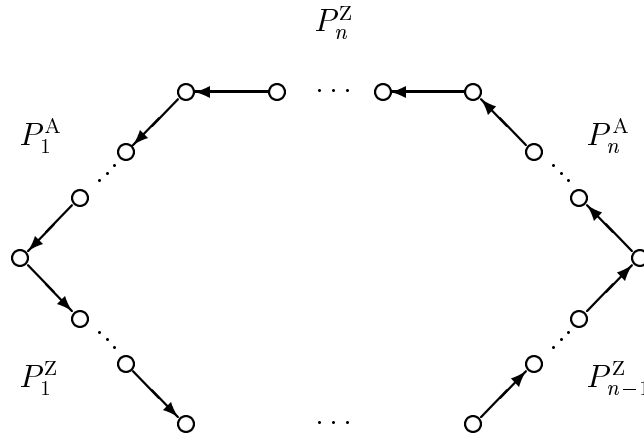


Abb. 3.2:

Um ein verzweigtes Stapelproblem $Q = (G_A, G_Z, k)$, für das die Vereinigung $G_A \cup G_Z$ nicht kreisfrei ist, lösen zu können, muß jeder derartige Kreis (siehe Abbildung 3.2) aufgeschnitten werden (Lemma 3.1). Dieses Aufschneiden kann nur durch das Ausstapeln eines Elementes des Ausgangsstapels auf einen Zwischenstapelplatz realisiert werden. Es ist demnach in geschickter Weise für jeden solchen Kreis ein Pfad P_i^A in C zu wählen, dessen Startknoten v ($d_{P_i^A}^-(v) = 0$) dann auf einen Hilfsstapelplatz gebracht wird. Dabei kommen natürlich nur solche Knoten v in Frage, die während des Lösungsprozesses auf der Spitze des Ausgangsstapels liegen, da auf darunterliegende Knoten erst später zugegriffen werden kann.

SCHMIEDEL[13] fand hierzu ein wesentliches Kriterium:

Besteht ein Kreis statt wie in Abbildung 3.2 jeweils nur aus einem Pfad P^A aus G_A und P^Z aus G_Z , so kommt nur ein Knoten v (nämlich der Startknoten von P^A) zum Aufschneiden des Kreises in Frage. Von diesem Knoten v wissen wir, daß er auf einen Zwischenstapelplatz gebracht werden muß.

Lemma 3.6 SCHMIEDEL[13] Ist $Q = (G_A, G_Z, k)$ ein lösbares, verzweigtes Stapelproblem und existiert ein Kreis $C = [a_1, a_2, \dots, a_x, b_1, b_2, \dots, b_y, a_1]$ in $G_A \cup G_Z$ für den

$$\begin{aligned} & (a_i, a_{i+1}), (a_x, b_1) \in A(G_A) \text{ für } i = 1, \dots, x-1 \\ \text{und} & (b_i, b_{i+1}), (b_y, a_1) \in A(G_Z) \text{ für } i = 1, \dots, y-1 \end{aligned}$$

gilt, dann folgt für jede l -Umstapelfolge \mathcal{S} von Q : $a_1 \in M^l$.

Beweis: Für eine Stapelkonfiguration S^i sind in der Menge M^i all diejenigen Elemente eingetragen, die während des vorangegangenen Umstapelprozesses auf

einem der Hilfsstapelplätze gelegen haben. Nach Lemma 3.1 muß C durch das Ausstapeln eines der Elemente a_1, \dots, a_x aufgeschnitten werden. Bevor jedoch die Elemente a_2, a_3, \dots, a_x vom Ausgangsstapel entnommen werden können, muß a_1 vom Ausgangsstapelplatz entnommen werden. Damit muß a_1 in jedem Fall auf einen Hilfsstapelplatz gelegt werden. Damit gilt $a_1 \in M^l$. \square

Definition 3.8 Ein gerichteter Graph $G = (V, A)$ heißt genau dann stark zusammenhängend, wenn es für alle Knotenpaare (x, y) aus V einen gerichteten Pfad von x nach y in G gibt.

Ein Teilgraph $G' \subseteq G$ von G heißt genau dann *stark zusammenhängende Komponente* von G , wenn G' stark zusammenhängend ist und G' maximal ist, d.h., wenn alle Teilgraphen $G'' \subseteq G$ mit $G' \subset G''$ keine stark zusammenhängenden Komponenten von G sind.

Lemma 3.7 Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes Stapelproblem, wobei $G_A \cup G_Z$ stark zusammenhängend sei. Weiterhin gelte $n := |V(G_A)|$ und $m := |A(G_A \cup G_Z)|$. Dann existiert ein $k \leq m - n + 1$, so daß Q lösbar ist.

Beweis: Wir beweisen Lemma 3.7 mit Hilfe vollständiger Induktion über die Knotenzahl der stark zusammenhängenden Komponente.

Induktionsanfang: Für $|V(G_A) \cup V(G_Z)| = 1$ ist zur Lösung von Q keine Ausstapelung erforderlich, das bedeutet, es gilt $k = 0$. Gleichzeitig gilt aber auch $m = 0$ und damit $m - n + 1 = 0 = k$.

Induktionsschritt: Wir betrachten Q . Beim Ausstapeln eines Stapелеlementes v auf einen Hilfsstapelplatz zerfällt $G_A \cup G_Z$ in r stark zusammenhängende Komponenten Q_i (einschließlich v selbst). Jede dieser Komponenten Q_i enthalte n_i Knoten und m_i Kanten. Außerdem ist nach Induktionsvoraussetzung jedes dieser neu entstandenen Stapelprobleme Q_i mit $m_i - n_i + 1$ Hilfsstapelplätzen lösbar. In Q bildeten alle Q_i eine einzige stark zusammenhängende Komponente, somit muß es von jedem Q_i einen gerichteten Pfad zu jedem Q_j gegeben haben. Dazu sind mindestens r gerichtete Kanten außerhalb der Komponenten Q_i notwendig. Damit gilt folgende Abschätzung:

$$\sum_{i=1}^r m_i \leq m - r \quad (3.2)$$

Berechnen wir nun die im schlechtesten Fall notwendige Anzahl an Hilfsstapelplätzen: Um v auszustapeln benötigen wir einen solchen Platz – für jede Komponente Q_i höchstens $m_i - n_i + 1$ Hilfsstapelplätze:

$$1 + \sum_{i=1}^r (m_i - n_i + 1) \leq 1 + \sum_{i=1}^r m_i - \sum_{i=1}^r n_i + \sum_{i=1}^r 1.$$

Wegen $\sum_{i=1}^r n_i = n$ und (3.2) gilt:

$$1 + \sum_{i=1}^r (m_i - n_i + 1) \leq 1 + m - r - n + r = m - n + 1$$

Damit ist das Lemma bewiesen. \square

Satz 3.8 *Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes lösbares Stapelproblem, dabei sei k die kleinste Anzahl an Hilfsstapelplätzen, für die Q lösbar ist. dc sei die Anzahl knotendisjunkter Kreise in $G_A \cup G_Z$ und l sei die Anzahl nichttrivialer stark zusammenhängender Komponenten Q^1, \dots, Q^l in $G_A \cup G_Z$. Eine solche Komponente Q^i enthalte n_i Knoten und m_i Kanten. Dann gilt*

$$\sum_{i=1}^l \min\{m_i - n_i + 1, n_i - 1\} \geq k \geq dc \geq l.$$

Beweis: Da es in einer nichttrivialen stark zusammenhängenden Komponente mehrere knotendisjunkte Kreise geben kann – und es wenigstens einen geben muß –, jedoch ein Kreis nicht in mehreren stark zusammenhängenden Komponenten liegen kann, gilt $dc \geq l$.

Nach Lemma 3.1 muß jeder Kreis durch das Ausstapeln eines Knotens aufgeschnitten werden. Durch das Ausstapeln von k Knoten können aber nur k knotendisjunkte Kreise aufgetrennt werden. Somit gilt $k \geq dc$.

Für eine stark zusammenhängende Komponente mit n_i Elementen und m_i Kanten müssen offensichtlich höchstens $n_i - 1$ Knoten ausgestapelt werden. Gleichzeitig sind nach Lemma 3.7 höchstens $m_i - n_i + 1$ Hilfsstapelplätze notwendig. Somit können wir für jede stark zusammenhängende Komponente von Q das Minimum dieser beiden Werte wählen.

Damit gilt $\sum_{i=1}^l \min\{m_i - n_i + 1, n_i - 1\} \geq k$. \square

Aus der Argumentation zum Beweis von Satz 3.8 läßt sich sofort schlußfolgern, daß es ausreicht, die stark zusammenhängenden Komponenten eines Stapelproblems Q zu betrachten, um dessen Lösbarkeit zu bestimmen. Für einen Lösungsalgorithmus ist es daher sinnvoll, ein Stapelproblem Q in Teilprobleme Q^1, \dots, Q^l zu zerlegen, die sich bei der Bestimmung der stark zusammenhängenden Komponenten ergeben.

Definition 3.9 Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes Stapelproblem. Die Menge der Stapelprobleme $\mathcal{Q} := \{Q^i = (G_A^i, G_Z^i, k_i) : i = 1, \dots, l\}$ heißt genau dann *scc-Dekomposition von Q* , wenn $G_A^i \cup G_Z^i$ für alle $i \in \{1, \dots, l\}$ stark zusammenhängende Komponente von $G_A \cup G_Z$ ist, $V(G_A) = \cup_{i=1}^l V(G_A^i)$ und $V(G_Z) = \cup_{i=1}^l V(G_Z^i)$ gilt und für alle $i \in \{1, \dots, l\}$ G_A^i und G_Z^i induzierte Untergraphen von G_A bzw. von G_Z sind.

Lemma 3.9 Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes Stapelproblem und $\mathcal{Q} := \{Q^i = (G_A^i, G_Z^i, k_i) : i = 1, \dots, l\}$ eine scc-Dekomposition von Q so daß für alle $Q^i = (G_A^i, G_Z^i, k_i)$ gilt: k_i ist minimal, d. h. Q^i ist lösbar, $Q^i = (G_A^i, G_Z^i, k_i - 1)$ jedoch nicht. Dann gilt:

$$Q \text{ ist genau dann lösbar, wenn } k \geq \sum_{i=1}^l k_i \text{ gilt.}$$

Beweis: („ \Rightarrow “) $G_A \cup G_Z$ besteht aus den stark zusammenhängenden Komponenten $G_A^1 \cup G_Z^1, \dots, G_A^l \cup G_Z^l$, wobei zur Lösung von Q aus der Komponente $G_A^i \cup G_Z^i$ genau c_i Knoten ausgestapelt werden ($i = 1, \dots, l$). Insgesamt werden k Elemente ausgestapelt, d. h. $\sum_{i=1}^l c_i = k$. Angenommen, Q ist lösbar und es gilt $k < \sum_{i=1}^l k_i$. Somit gilt: $\sum_{i=1}^l c_i < \sum_{i=1}^l k_i$. Deshalb existiert wenigstens eine Komponente $G_A^j \cup G_Z^j$, für die c_j Knoten v_1, \dots, v_{c_j} ausgestapelt werden und für die $Q^j = (G_A^j, G_Z^j, c_j)$ mit $c_j < k_j$ gilt und die lösbar ist, da das gesamte Problem Q lösbar ist. Das widerspricht jedoch der Voraussetzung, daß k_j minimal ist. Aus diesem Grund muß die Annahme falsch sein.

(„ \Leftarrow “) Es gelte $k \geq \sum_{i=1}^l k_i$. Dann ist Q lösbar, denn wenn jede stark zusammenhängende Komponente für sich gelöst wird, wird auch das gesamte Problem gelöst, da es keine Kreise geben kann, die durch mehrere stark zusammenhängende Komponenten verlaufen. Damit werden zur Lösung nur $\sum_{i=1}^l k_i \leq k$ Hilfsstapelplätze zur Lösung von Q benötigt. \square

Es ergeben sich zwei Ansätze, die zur algorithmischen Bearbeitung des Problems herangezogen werden können. Einerseits kann das Problem exakt durch vollständige Enumeration gelöst werden, was aufgrund der NP-Vollständigkeit sehr lange dauern wird. Andererseits ist die Bearbeitung durch ein heuristisches Verfahren denkbar, welches wesentlich schneller – insbesondere in Polynomialzeit – arbeiten kann, jedoch im allgemeinen nur eine Näherungslösung liefert.

In den folgenden Abschnitten sollen Algorithmen zu diesen Lösungsansätzen erläutert und die Ergebnisse ihrer Implementierungen vorgestellt werden.

3.3.2 Branching & Bounding

Um algorithmisch die kleinste notwendige Anzahl an Hilfsstapelplätzen zu bestimmen, bedienen wir uns einer simplen Technik: Wir simulieren einfach das schrittweise Umstapeln von Elementen des Ausgangsstapels auf den Zielstapelplatz. Immer dann, wenn kein Element der Spitze des Ausgangsstapelplatzes auf den Zielstapelplatz gelegt werden kann, benötigen wir einen Hilfsstapelplatz. Nach Lemma 3.9 reicht es aus, alle Stapelprobleme Q^i der scc-Dekomposition von Q zu betrachten. Die kleinste nötige Anzahl an Hilfsstapelplätzen, die zur Lösung von Q benötigt werden, ergibt sich dann aus der Summe der Lösungen der Q^i . Wir zerlegen demnach das Problem $Q = (G_A, G_Z, k)$ in die Teilprobleme $Q^i = (G_A^i, G_Z^i, k_i)$ indem wir die stark zusammenhängenden Komponenten der Vereinigung $G_A \cup G_Z$ bilden.

Nachdem wir getestet haben, daß die zu untersuchende Komponente Q^i nicht trivial ist – nicht nur aus einem isolierten Knoten besteht – überprüfen wir, ob sich für Q^i ein Kreis finden läßt, der ein Element v der Spitze des Ausgangsstapelplatzes enthält, so daß Lemma 3.6 erfüllt ist. Sollte das der Fall sein, so stapeln wir v auf einen Hilfsstapelplatz aus. Durch das Entfernen von v kann

es passieren, daß die stark zusammenhängende Komponente zerfällt. An dieser Stelle haben wir nun die Möglichkeit die zerfallene Komponente wieder in stark zusammenhängende Komponenten zu zerlegen oder aber nach weiteren Knoten v zu suchen, die Lemma 3.6 erfüllen. Für unsere Untersuchung wurde die zweite Möglichkeit gewählt: Durch das Ausstapeln von v entsteht nun auf dem Ausgangsstapelplatz eine neue Spitze. Wir können nun nach Knoten auf dieser Spitze suchen, die sich direkt auf den Zielstapelplatz ein stapeln lassen oder Lemma 3.6 erfüllen. Diese Schritte wiederholen wir solange, bis sich keine solchen Knoten mehr finden lassen. An diesem Punkt angelangt, müssen wir einen Knoten der Spitze auf einen Hilfsstapelplatz bringen – da wir aber nicht wissen welchen, bleibt uns nur, nach und nach jeden dieser Knoten auszustapeln, den verbleibenden Rest von Q^i rekursiv zu lösen und uns die entsprechende Lösung – Anzahl der benötigten Hilfsstapelplätze – zu merken. Dieses einfache Backtracking-Verfahren wird zum Branch-and-Bound Algorithmus, indem wir uns das jeweils beste erzielte Ergebnis merken und die Berechnung für einen ausgestapelten Knoten sofort abbrechen, sobald die in dieser Verzweigung benötigte Anzahl an Hilfsstapelplätzen größer wird, als dieses bisher beste Ergebnis. Als Resultat erhalten wir die minimal notwendige Anzahl an Hilfsstapelplätzen, die zur Lösung von Q^i benötigt wird.

3.3.3 Heuristische Lösungsverfahren

Ein mögliches heuristisches Verfahren verläuft wie der Branch-and-Bound Algorithmus. Jedoch an der Stelle, an der weder Knoten der Spitze existieren, die Lemma 3.6 erfüllen, noch Knoten direkt auf den Zielstapelplatz gebracht werden können, muß hier genau ein Knoten ausgewählt werden, der auf den Hilfsstapelplatz gebracht wird. Da diese Entscheidung welcher Knoten ausgestapelt wird, nicht wieder korrigiert werden kann, müssen wir diese Wahl von einem definierten Kriterium abhängig machen. Dazu bewerten wir alle in Frage kommenden Knoten (die Knoten der Spitze des Ausgangsstapelplatzes) durch eine Funktion. Damit wird deutlich, daß die Effizienz und Qualität eines heuristischen Verfahrens von einer solchen Bewertungsfunktion abhängt. Im folgenden sollen nun zwei mögliche Bewertungsfunktionen vorgestellt werden. Zuerst wurde die Funktion f_1 betrachtet:

Definition 3.10 Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes Stapelproblem und $Q^i = (G_A^i, G_Z^i, k_i)$ ein Stapelproblem aus der scc-Dekomposition von Q . Weiterhin sei $\{v_1, \dots, v_l\}$ die Menge der Elemente auf der Spitze von G_A^i . Für jeden Knoten v_t der Spitze von G_A^i seien $C_{t,1}, \dots, C_{t,r_t}$ alle diejenigen Kreise in $G_A^i \cup G_Z^i$, für die $v_t \in V(C_{t,j})$ für $j \in \{1, \dots, r_t\}$ gelte. Dann ist f_1 wie folgt definiert:

$$f_1(v_t) := |\{v_1, \dots, v_l\} \cap \bigcup_{j=1}^{r_t} V(C_{t,j})|.$$

Demnach bestimmt $f_1(v_t)$ die Anzahl derjenigen Knoten der Spitze, die in Kreisen in $G_A^i \cup G_Z^i$ enthalten sind in denen auch v_t auftritt. Derjenige Knoten mit dem größten Funktionswert wird schließlich auf einen Hilfsstapelplatz gebracht. Der Rest der betrachteten stark zusammenhängenden Komponente Q^i wird nun separat rekursiv untersucht.

Mit Hilfe dieser Bewertungsfunktion f_1 konnten in ersten Untersuchungen auch akzeptable Ergebnisse erzielt werden. Eine echte Auswahl eines der Elemente der Spitze im Ausgangsstapel findet jedoch nur dann statt, wenn durch direkte Einstapelungen in den Zielstapel oder durch Ausstapelungen von Knoten nach Lemma 3.6 die zu untersuchende stark zusammenhängende Komponente bereits zerlegt wurde. Weiterhin ist die Berechnung von f_1 algorithmisch recht aufwendig. Aus diesem Grund wurde nach einer weiteren möglichen Bewertungsfunktion gesucht, die in erster Linie weniger Rechenzeit benötigt und die mit f_1 erzielten Ergebnisse eventuell verbessert. Hier sollen nur einige dieser untersuchten Funktionen erwähnt werden:

Definition 3.11 Es sei $Q = (G_A, G_Z, k)$ ein verzweigtes Stapelproblem und $Q^i = (G_A^i, G_Z^i, k_i)$ ein Stapelproblem aus der scc-Dekomposition von Q . Weiterhin sei $\{v_1, \dots, v_l\}$ die Menge der Elemente auf der Spitze von G_A^i . Für jedes Element v_t der Spitze von G_A^i sind folgende Funktionen definiert:

- $f_2(v_t) := d_{G_A^i}^+(v_t) + d_{G_Z^i}^-(v_t)$,
- $f_3(v_t) := |\{v \in N_{G_A^i}^+(v_t) : N_{G_A^i}^-(v) = \{v_t\} \text{ und } v \text{ kann direkt auf den Zielstapel gelegt werden.}\}|$,
- $f_4(v_t) := |\{v \in N_{G_A^i}^+(v_t) : N_{G_A^i}^-(v) = \{v_t\} \text{ und } v \text{ erfüllt Lemma 3.6.}\}|$,
- $f_5(v_t) := f_3(v_t) + f_4(v_t)$.

Die Funktion f_5 zählt somit diejenigen Nachbarknoten von v_t , die nach dem Ausstapeln von v_t entweder direkt auf den Zielstapel gebracht werden können oder nach Lemma 3.6 auf jeden Fall ausgestapelt werden müssen. Die mit Abstand besten Ergebnisse wurden mit dieser Bewertungsfunktion f_5 erzielt.

3.3.4 Zeitkompliziertheit der Heuristik

Wir wollen nun die wichtigsten Algorithmen des Verfahrens vorstellen und eine Zeitabschätzung der Prozeduren vornehmen. Wie bereits in Lemma 3.9 ersichtlich war, zerlegen wir ein zu untersuchendes Stapelproblem Q in seine stark zusammenhängenden Komponenten.

Gegeben: Stapelproblem $Q = (G_A, G_Z, |V(G_A)|)$.

Gesucht: Natürliche Zahl k , so daß $Q' = (G_A, G_Z, k)$ lösbares Stapelproblem und k minimal ist.

procedure AUSSTAPELZAHL($Q = (G_A, G_Z, |V(G_A)|)$);
begin
 wähle einen Knoten v aus der Spitze des Ausgangsstapels G_A ;
 bilde die stark zusammenhängende Komponente, in der v liegt;

- $Q^1 = (G_A^1, G_Z^1, |V(G_A^1)|)$ sei Stapelproblem, für das $G_A^1 \cup G_Z^1$ die stark zusammenhängende Komponente bildet, in der v liegt,
- $Q^{\text{Rest}} = (G_A^{\text{Rest}}, G_Z^{\text{Rest}}, |V(G_A^{\text{Rest}})|)$ sei dasjenige Stapelproblem, das aus Q nach Abtrennung von Q^1 hervorgeht;

$\text{AUSSTAPELZAHL}(Q) := \text{KOMONENTENAUSSTAPELZAHL}(Q^1)$
 $\quad\quad\quad + \text{AUSSTAPELZAHL}(Q^{\text{Rest}});$

end;

Algorithmus 1: AUSSTAPELZAHL(Q)

Es sei $n := |V(G_A)| = |V(G_Z)|$ und $m := |A(G_A)| + |A(G_Z)|$. Für das Stapelproblem Q^1 der stark zusammenhängenden Komponente gelte: $n_1 := |V(G_A^1)|$, $m_1 := |A(G_A^1)| + |A(G_Z^1)|$. Die zur Abspaltung der stark zusammenhängenden Komponente Q^1 von Q benötigte Rechenzeit $T_D(n, m)$ beträgt $O(n + m)$, auf diesen Algorithmus soll hier nicht näher eingegangen werden. Damit beträgt die notwendige Rechenzeit T_A von AUSSTAPELZAHL(Q):

$$T_A(n, m) = T_D(n, m) + T_{KA}(n_1, m_1) + T_A(n - n_1, m - m_1),$$

wobei T_{KA} die Rechenzeit der Prozedur KOMONENTENAUSSTAPELZAHL(Q) bezeichnet. Diese Prozedur berechnet nun die Anzahl notwendiger Ausstapelungen für ein Stapelproblem Q^i der scc-Dekomposition von Q :

Gegeben: Stapelproblem $Q^i = (G_A^i, G_Z^i, |V(G_A^i)|)$, so daß
 $G_A^i \cup G_Z^i$ stark zusammenhängend ist.
 Gesucht: Natürliche Zahl k_i , so daß $Q^{i'} = (G_A^i, G_Z^i, k_i)$ lösbares
 Stapelproblem und k_i minimal ist.

procedure KOMONENTENAUSSTAPELZAHL($Q^i = (G_A^i, G_Z^i, |V(G_A^i)|)$);
begin
if ($|V(G_A^i)| = 0$ **or** $|V(G_A^i)| = 1$)
then KOMONENTENAUSSTAPELZAHL(Q^i) := 0
else begin
 $k := 0$;
while es existiert ein Knoten v , der auf einem der Hilfsstapelplätze
 oder auf der Spitze von G_A^i liegt ($d_{G_A^i}^-(v) = 0$)
 und der Lemma 3.6 erfüllt oder sich in den Zielstapel ein stapeln läßt

```

do begin
  if  $v$  genügt Lemma 3.6 then  $k := k + 1$ ;
  Staple  $v$  auf einen Hilfs- bzw. Zielstapelplatz ( $G_A^i \rightarrow G_A^{i'}, G_Z^i \rightarrow G_Z^{i'}$ );
end;
Bewerte die Knoten  $v$  auf der Spitze von  $G_A^i$  mit Hilfe der Funktion  $f_5$ ;
Staple einen Knoten  $v$ , für den  $f_5(v)$  maximal ist,
  auf einen Hilfsstapelplatz aus ( $G_A^{i'} \rightarrow G_A^{i''}$ );
 $k := k + 1$ ;
KOMONENTENAUSSTAPELZAHL( $Q^i$ ):= $k$ 
  +AUSSTAPELZAHL( $Q^{i'} = (G_A^{i'}, G_Z^{i'}, |V(G_A^{i''})|$ ));
end;
end;

```

Algorithmus 2: KOMONENTENAUSSTAPELZAHL(Q^i)

Die Überprüfung der Bedingung in der **while**-Schleife für einen zu betrachtenden Knoten v und das Ausstapeln dieses Knotens v benötigt eine Rechenzeit von $O(n + m)$. Zur Bewertung der Knoten aus der Spitze von G_A^i durch f_5 müssen $O(n + m_i)$ Schritte ausgeführt werden. Für die zur Berechnung von KOMONENTENAUSSTAPELZAHL(Q^i) benötigte Zeit $T_{KA}(n_i, m_i)$ gilt damit:

$$T_{KA}(n_i, m_i) = O(n_{i_1}(n + m) + n + m_{i_1}) + T_A(n_i - n_{i_1}, m_i - m_{i_1}),$$

wobei n_{i_1} die Anzahl derjenigen Knoten angibt, die innerhalb der **while**-Schleife auf den Zielstapel gebracht wurden – m_{i_1} ist die Summe der in G_A^i aus diesen Knoten ausgehenden Kanten. Wir wollen nun den Rechenzeitaufwand T_A der Prozedur AUSSTAPELZAHL bestimmen. Im schlechtesten Fall handelt es sich beim zu untersuchenden Stapelproblem Q um ein Problem, das sich nicht weiter in stark zusammenhängende Komponenten zerlegen läßt. T_A läßt sich somit wie folgt abschätzen:

$$T_A(n, m) \leq T_D(n, m) + T_{KA}(n, m).$$

Somit ist die Prozedur KOMONENTENAUSSTAPELZAHL nur auf ein Stapelproblem – nämlich Q selbst anzuwenden:

$$\begin{aligned} T_{KA}(n, m) &= O(n_1(n + m) + m_1 + n) + T_A(n - n_1, m - m_1) \\ &\leq O(n_1(n + m) + m_1 + n) + T_D(n - n_1, m - m_1) \\ &\quad + T_{KA}(n - n_1, m - m_1). \end{aligned}$$

Die Rekursionstiefe zur Lösung von Q betrage t , $t \leq n$. Somit gilt:

$$\begin{aligned} T_{KA}(n, m) &\leq O(n_1(n + m) + m + n) + T_D(n - n_1, m - m_1) \\ &\quad + O(n_2(n + m) + m - m_1 + n) + T_D(n - n_1 - n_2, m - m_1 - m_2) \\ &\quad + \end{aligned}$$

$$\begin{aligned} & \vdots \\ & + O\left(n_t(n+m) + m - \sum_{i=1}^{t-1} m_i + n\right) + T_D\left(n - \sum_{i=1}^t n_i, m - \sum_{i=1}^t m_i\right). \end{aligned}$$

Damit folgt:

$$\begin{aligned} T_{KA}(n, m) &\leq O\left(\sum_{i=1}^t (n_i(n+m)) + \sum_{i=1}^t \left(m - \sum_{j=1}^{i-1} m_j\right) + \sum_{i=1}^t n\right) \\ &+ \sum_{i=1}^t T_D\left(\left(n - \sum_{j=1}^i n_j\right), \left(m - \sum_{j=1}^i m_j\right)\right). \end{aligned}$$

Es gilt $t \leq n$, $\sum_{i=1}^t n_i \leq n$, $\sum_{i=1}^t m_i \leq m$, $T_D(n, m) = O(n+m)$. Wir können folgende Abschätzung vornehmen:

$$\begin{aligned} T_{KA}(n, m) &\leq O\left(\sum_{i=1}^t n_i(n+m) + \sum_{i=1}^t m + \sum_{i=1}^t n\right) + \sum_{i=1}^t T_D(n, m) \\ &\leq O(n^2 + nm + nm + n^2) + nT_D(n, m) \\ &= O(n^2 + nm) = O(n^3). \end{aligned}$$

Für T_A erhalten wir so:

$$T_A(n, m) \leq O(n+m) + O(n^2 + nm) = O(n^2 + nm) = O(n^3).$$

Dies ist auch gleichzeitig die benötigte Rechenzeit des gesamten Lösungsalgorithmus, da zur Berechnung derjenigen Knoten, die Lemma 3.6 genügen, ebenfalls nur $O(n^2 + nm)$ Schritte benötigt werden.

3.3.5 Erfahrungen mit den Algorithmen

Zur Durchführung von Experimenten sind zunächst Beispielgraphen zu generieren, die Ausgangs- und Zielstapel eines Stapelproblems repräsentieren. Die Knoten eines gerichteten, kreisfreien Graphen können in gewisse Ebenen eingeordnet werden, zwei Knoten liegen genau dann in einer Ebene, wenn die Länge der kürzesten Pfade, die von der Spitze des Graphen zum Knoten führen, gleich ist. Die *durchschnittliche Breite* eines solchen Graphen ergibt sich aus dem arithmetischen Mittel der Knotenanzahl in diesen Ebenen des Graphen. Die Höhe eines gerichteten, kreisfreien Graphen G ist die Anzahl dieser Ebenen – nach Definition 3.2 die Länge des längsten Pfades in G .

Obwohl das Generieren solcher zufälligen Graphen kein triviales Problem ist, soll hier nicht näher darauf eingegangen werden. Im wesentlichen verwenden wir

die Methode, die HEMPEL und VASILKOV [15] erläutern. Dabei lassen sich unter anderem die durchschnittliche Breite und Höhe eines zu generierenden Graphen vorgeben sowie dessen Kantendichte – d. h. die Wahrscheinlichkeit, mit der mögliche Kanten im Graphen existieren.

Bei den ersten Experimenten fiel auf, daß ein sehr hoher Anteil der untersuchten Probleme in vernünftiger Zeit lösbar ist. Algorithmisch bedeutet das für diese Beispiele: Im Branch-and-Bound Verfahren ist keine Verzweigung nötig, im heuristischen Verfahren müssen keine Knoten durch f_5 bewertet werden. Schritt für Schritt finden sich immer wieder Knoten, die entweder den Voraussetzungen von Lemma 3.6 genügen oder sich direkt auf den Zielstapelplatz ein stapeln lassen, solange, bis der Ausgangsstapel vollständig abgearbeitet wurde. Das Branch-and-Bound Verfahren und das heuristische Verfahren liefern in diesem Fall natürlich die gleichen Ergebnisse. In Tabelle 3.1 wird der Anteil derjenigen Beispiele (gemessen an allen untersuchten Problemen) angegeben, welche ohne Verzweigung lösbar sind. Die betrachteten Graphen haben maximal 100 Knoten, die Kantendichte beträgt 25 %. Es wurde die Abhängigkeit von der Struktur der Graphen – Verhältnis von Höhe zu Breite – untersucht.

Kantendichte: 25 %, $ V \leq 100$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel			
	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	97%	94%	91%
$r_A = 1$	96%	90%	83%
$r_A = 0.1$	90%	80%	63%

Tabelle 3.1: Anteil der Beispiele, die in akzeptabler Zeit lösbar sind, gemessen an allen untersuchten Stapelproblemen mit den angegebenen Parametern.

Dabei scheint diese Struktur der Stapel eine wesentliche Rolle zu spielen, so daß auch in den folgenden Experimenten diese Struktur als Parameter variiert wurden. Dabei muß erwähnt werden, daß im weiteren Text nur noch über Beispiele berichtet wird, für deren Lösung im Branch-and-Bound Algorithmus wirklich verzweigt werden muß – im heuristischen Verfahren demnach eine Bewertung durch die Funktion f_5 stattfindet.

Qualität der Heuristik

In mehreren Versuchsreihen wurden die Ergebnisse des heuristischen Verfahrens für 5,10,15,20 und 25 Prozent Kantendichte bewertet. Dabei wurden wieder jeweils drei verschiedene Strukturen von Ausgangs- und Zielstapel gegenübergestellt. In Tabelle 3.2 wird angegeben, in wieviel Prozent der untersuchten Fälle die Heuristik das Optimum (O) und eine „gute“ (G) Lösung (Abweichung vom

Optimum um maximal einen Knoten) liefert. Die Genauigkeit, mit der das heuristische Verfahren arbeitet, übertraf alle Erwartungen.

Kantendichte: 25 %, $ V \leq 50$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel				Kantendichte: 15 %, $ V \leq 50$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel			
	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$		$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	O: 93% G: 99%	O: 94% G: 99%	O: 88% G: 98%	$r_A = 10$	O: 86% G: 99%	O: 81% G: 98%	O: 82% G: 98%
$r_A = 1$	O: 91% G: 99%	O: 88% G: 99%	O: 82% G: 99%	$r_A = 1$	O: 81% G: 97%	O: 77% G: 96%	O: 77% G: 98%
$r_A = 0.1$	O: 84% G: 98%	O: 78% G: 97%	O: 74% G: 97%	$r_A = 0.1$	O: 81% G: 96%	O: 76% G: 96%	O: 82% G: 95%
Kantendichte: 10 %, $ V \leq 50$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel				Kantendichte: 5 %, $ V \leq 50$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel			
	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$		$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	O: 71% G: 94%	O: 71% G: 95%	O: 79% G: 96%	$r_A = 10$	O: 75% G: 94%	O: 75% G: 94%	O: 84% G: 98%
$r_A = 1$	O: 70% G: 91%	O: 76% G: 94%	O: 79% G: 96%	$r_A = 1$	O: 79% G: 99%	O: 86% G: 97%	O: 91% G: 99%
$r_A = 0.1$	O: 77% G: 96%	O: 79% G: 94%	O: 81% G: 97%	$r_A = 0.1$	O: 87% G: 98%	O: 88% G: 99%	O: 95% G: 99%

Tabelle 3.2: Qualität der Heuristik für verschiedene Kantendichten

Rechenzeit der Algorithmen im Vergleich

Der Vergleich der benötigten Rechenzeiten des Branch-and-Bound und des heuristischen Verfahrens fällt recht eigentümlich aus: Die betrachteten Beispiele lassen sich in zwei Gruppen einteilen: In der ersten Gruppe – der überwiegenden Anzahl der Beispiele – gibt es kaum einen Unterschied zwischen den beiden Verfahren, die Rechenzeit betreffend. So liegt die benötigte Rechenzeit $T_{\text{Heuristik}}(Q)$ des Näherungsverfahrens für Stapelprobleme Q aus dieser Gruppe unter einer Sekunde¹. Die Rechenzeit $T_{\text{Branch-and-Bound}}(Q)$ des exakten Verfahrens beträgt ebenfalls weniger als eine Sekunde. In der zweiten – deutlich kleineren – Gruppe steigt der Aufwand des exakten Verfahrens enorm an, hier finden wir Rechenzeiten von 30 Sekunden bis zu mehreren Stunden, während die gleichen Beispiele mit dem Näherungsverfahren in wenigen Sekunden gelöst werden können.

¹Alle angegebenen Rechenzeiten beziehen sich auf einen Pentium 133 mit 32 MB RAM Hauptspeicher.

Für Probleme größerer Ordnung ($n \geq 200$) ist diese Einteilung nicht ohne weiteres möglich. Hier arbeitet die Heuristik zwar ebenfalls sehr schnell, jedoch lassen sich keine nachprüfbaren Aussagen über die Qualität der Ergebnisse machen, da es praktisch unmöglich ist, die exakten Werte zu bestimmen.

Mit Tabelle 3.3 geben wir einen Überblick über die benötigten Rechenzeiten des exakten Verfahrens in Abhängigkeit von der Struktur der untersuchten Stapel. Dabei wurden dieselben Beispiele ausgewertet die bereits in Tabelle 3.2 betrachtet wurden.

Rechenzeit	Kantendichte:			
	5%	10%	15%	25%
<1s	87	74	91	59
1 – 5s	9	17	6.7	36
6 – 120s	3	5	1.7	4
121 – 7200s	0.8	3	0.4	0.6
> 7200s	0.2	1	0.2	0.4
Anzahl der berechneten Beispiele	1240	2785	3043	1623

Tabelle 3.3: Prozentuale Verteilung der untersuchten Stapelprobleme bezüglich Kantendichte und benötigter Rechenzeit für den Branch-and-Bound Algorithmus

Tabelle 3.4 zeigt den Anteil „schwieriger“ Beispiele an den untersuchten Stapelproblemen. Schwierig soll in diesem Fall bedeuten, daß die zur Lösung eines Stapelproblems Q benötigte Zeit $T_{\text{Branch-and-Bound}}(Q)$ mehr als zehnmal größer ist, als die Zeit $T_{\text{Heuristik}}(Q)$, die der heuristische Algorithmus benötigt:

$$T_{\text{Branch-and-Bound}}(Q) > 10 \cdot T_{\text{Heuristik}}(Q).$$

Unter diesen Beispielen ist jedoch auch der Faktor 1000 keine Seltenheit.

Kantendichte: 20 %, $ V \leq 50$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel				Kantendichte: 10 %, $ V \leq 50$ r_A =Höhe/Breite im Ausgangsstapel r_Z =Höhe/Breite im Zielstapel			
	$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$		$r_Z = 10$	$r_Z = 1$	$r_Z = 0.1$
$r_A = 10$	0.7%	0.1%	0.1%	$r_A = 10$	8%	6%	4%
$r_A = 1$	0.5%	0.3%	0.4%	$r_A = 1$	11%	10%	4%
$r_A = 0.1$	2.6%	3.5%	7.5%	$r_A = 0.1$	5%	8%	2%

Tabelle 3.4: Anteil „schwieriger“ Stapelprobleme für verschiedene Kantendichten

Ursprünglich bestand die Vorstellung, daß Stapelprobleme mit einer sehr hohen Kantendichte (25%) sehr schwierig zu lösen seien. Die Auswertung der Expe-

perimente ergab jedoch eher das Gegenteil. HEMPEL und VASILKOV [15] berichten ebenfalls von steigender Schwierigkeit bei abnehmender Kantendichte des Graphen. Hierfür ist die geringe Wahrscheinlichkeit für die Existenz von Kreise verantwortlich zu machen, die Lemma 3.6 erfüllen – je geringer die Kantendichte, desto größer werden die Kreise in $G_A \cup G_Z$.

Gleichzeitig muß festgestellt werden, daß der Einfluß der Struktur der Graphen auf Güte und Geschwindigkeit der Heuristik nicht so groß ist, wie zunächst vermutet. Ebenfalls aus [15] entnehmen wir, daß die Zerlegung des Problems in stark zusammenhängende Komponenten nicht immer den erhofften Effekt bringt – in der Regel finden wir isolierte Knoten vor sowie eine sehr große Komponente. Bei näherer Betrachtung ist dies nicht weiter verwunderlich: Zwei größere stark zusammenhängende Komponenten A und B verschmelzen zu einer einzigen, sobald es nur einen Pfad gibt, der von A nach B führt und einen der von B nach A führt.

Abschließend können wir sagen, daß für Probleme mit weniger als 100 Stapel-elementen eine Kombination der beiden vorgestellten Verfahren sehr sinnvoll ist: Erst wenn das Branch-and-Bound Verfahren in einem vorgegebenen Zeitintervall keine Lösung liefert, sollte das heuristische Verfahren auf das Problem angesetzt werden.

Kapitel 4

Stapelprobleme mit nicht eineindeutiger Elementezuordnung

4.1 Perfekte Matchings in paaren Graphen

Definition 4.1 Ein Graph $G = (V, E)$ heißt genau dann *paarer Graph*, wenn $V = X \cup Y$ ist, $X \cap Y = \emptyset$ ist, und X und Y unabhängige Mengen in G sind. Für G schreiben wir dann auch $G = (X, Y, E)$.

Definition 4.2 Es sei $G = (V, E)$ ein ungerichteter Graph. Eine Menge $M \subseteq E$ von Kanten dieses Graphen heißt genau dann *Matching von G* , wenn keine zwei Kanten aus M mit einem gemeinsamen Knoten inzident sind.

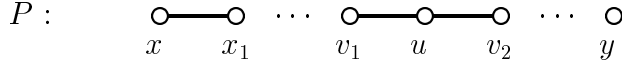
Ein Matching M eines Graphen G heißt genau dann *maximum Matching von G* , wenn für alle anderen Matchings M' von G $|M'| \leq |M|$ gilt. Ein maximum Matching M von G heißt genau dann *perfekt*, wenn $|M| = |V|/2$ gilt.

Lemma 4.1 Es seien $G_1 = (X, Y, E_1)$ und $G_2 = (X, Y, E_2)$ ungerichtete, paare Graphen. Weiterhin sei M_1 ein perfektes Matching von G_1 und M_2 perfektes Matching von G_2 . Dann liegt jeder Knoten aus $G_3 := (X, Y, M_1 \cup M_2)$ dessen Valenz zwei beträgt, in einem Kreis aus G_3 .

Beweis: Angenommen, Lemma 4.1 gilt nicht. Dann existieren Knoten $u, v_1, v_2 \in X \cup Y$ mit $v_1 \neq v_2$, für die gilt:

$$\{u, v_1\}, \{u, v_2\} \in M_1 \cup M_2 \text{ und } u \text{ liegt in keinem Kreis.}$$

Offensichtlich liegen die Kanten $\{u, v_1\}$ und $\{u, v_2\}$ in einem Pfad P in G_3 . Es seien x und y die Anfangs- bzw. Endknoten dieses Pfades.



Da x_1 die Valenz zwei hat, liegt $\{x, x_1\}$ entweder in M_1 oder in M_2 – o.B.d.A. nehmen wir $\{x, x_1\} \in M_1$ an. Da $\{x, x_1\} \notin M_2$ gilt und x in G_3 nur die Valenz eins hat, gibt es im Matching M_2 keine Kante, die mit x inzident ist. Das widerspricht jedoch der Voraussetzung, daß M_1 und M_2 perfekte Matchings sind. Damit ist die Annahme falsch und das Lemma bewiesen. \square

Lemma 4.2 Es seien $G_1 = (X, Y, E_1)$ und $G_2 = (X, Y, E_2)$ ungerichtete, paare Graphen. Weiterhin sei M_1 ein perfektes Matching von G_1 und M_2 perfektes Matching von G_2 . Der Graph $G_3 := (X, Y, M_1 \cup M_2)$ ist genau dann kreisfrei, wenn $M_1 = M_2$ gilt.

Beweis: („ \Leftarrow “): Trivial. („ \Rightarrow “): Angenommen, es gilt $M_1 \neq M_2$: Dann existieren voneinander verschiedene Knoten $u, v_1, v_2 \in X \cup Y$, für die $\{u, v_1\} \in M_1$ und $\{u, v_2\} \in M_2$ gilt. Damit hat der Knoten u Valenz zwei in G_3 – nach Lemma 4.1 besitzt G_3 somit einen Kreis. Das widerspricht der Voraussetzung. \square

Lemma 4.3 Es seien $G_1 = (X, Y, E_1)$ und $G_2 = (X, Y, E_2)$ ungerichtete, paare Graphen. Weiterhin sei M_1 ein perfektes Matching von G_1 und M_2 perfektes Matching von G_2 . Besitzt der Graph $G_3 := (X, Y, M_1 \cup M_2)$ genau einen Kreis, dann gibt es für G_3 außer M_1 und M_2 kein weiteres perfektes Matching.

Beweis: Angenommen, die Aussage des Lemmas sei falsch: Dann existieren Kanten $\{u, v\}$ und $\{x, y\}$ in einem perfekten Matching M_3 , für die

$$\{u, v\} \in M_1, \{u, v\} \notin M_2 \text{ und } \{x, y\} \in M_2, \{x, y\} \notin M_1 \text{ gilt.} \tag{4.1}$$

Da M_1 und M_2 perfekt sind, müssen u, v, x und y die Valenz zwei in G_3 haben. Damit liegt nach Lemma 4.1 jeder dieser Knoten in einem Kreis von G_3 – da es in G_3 nach Voraussetzung nur einen Kreis C gibt, liegen die Kanten $\{u, v\}$ und $\{x, y\}$ alle in C . Da es sich bei M_1 und M_2 um Matchings handelt, liegen die Kanten aus C abwechselnd in M_1 und M_2 . Da M_3 ein Matching ist, liegen die Kanten aus C ebenso abwechselnd in M_3 und nicht in M_3 . Somit sind die Kanten aus M_3 in C entweder alle aus M_1 oder alle aus M_2 . Somit gilt entweder $\{u, v\}, \{x, y\} \in M_1$ oder $\{u, v\}, \{x, y\} \in M_2$, was (4.1) widerspricht. \square

Satz 4.4 Es seien $G_1 = (X, Y, E_1)$ und $G_2 = (X, Y, E_2)$ ungerichtete, paare Graphen. M_1 sei ein perfektes Matching von G_1 und M_2 perfektes Matching von G_2 . Die Anzahl der Kreise im Graphen $G_3 := (X, Y, M_1 \cup M_2)$ betrage c . Dann gibt es genau 2^c verschiedene perfekte Matchings in G_3 .

Beweis: Wir führen den Beweis mittels vollständiger Induktion über die Anzahl der Kreise.

Induktionsanfang: Es sei $c = 0$, d. h. G_3 ist kreisfrei. Nach Lemma 4.2 gilt daher $M_1 = M_2$, das bedeutet, für G_3 existiert nur ein perfektes Matching: $2^0 = 1$. Damit ist der Induktionsanfang gesichert.

Induktionsschritt: Der Satz 4.4 gelte für $c = k$. Wir betrachten nun G_3 mit $c = k + 1$ Kreisen – C sei einer dieser Kreise. Für $G_3 \setminus C$ gilt:

Die Anzahl der perfekten Matchings in $G_3 \setminus C$ beträgt 2^k .

Nach Lemma 4.3 kommen für C nur die beiden Matchings M_1 und M_2 als perfekte Matchings in Frage. Für G_3 gibt es somit einmal die 2^k Matchings für $G_3 \setminus C$ und das Matching M_1 für C und das andere Mal die 2^k Matchings für $G_3 \setminus C$ und das Matching M_2 für C . Damit beträgt die Anzahl der perfekten Matchings für G_3 $2^k + 2^k = 2^{k+1} = 2^c$. \square

4.2 Eigenschaften nicht eineindeutiger Stapelprobleme

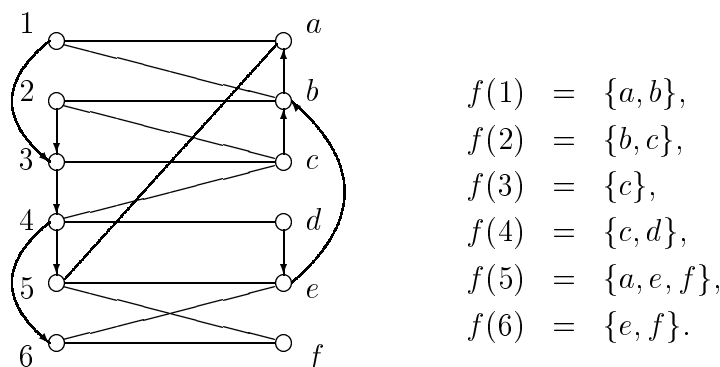
Bei den in den vorangegangenen Kapiteln untersuchten Stapelproblemen wurde immer eine eineindeutige Zuordnung der Elemente des Ausgangsstapels zu den Elementen des Zielstapels vorausgesetzt. Das hatte zur Folge, daß wir nicht zwischen Stapелеlementen und Plätzen im Stapel, auf die sie gebracht werden dürfen, unterscheiden mußten. Natürlich ist ein wesentlich allgemeineres Modell denkbar, bei dem diese Zuordnung keineswegs eineindeutig – ja noch nicht einmal eindeutig zu sein braucht.

Den praktischen Hintergrund eines solchen Modells bilden Stapелеlemente bzw. Positionen im Stapel, die auf eine definierte Weise bezüglich einer Position im Stapel bzw. eines Stapелеlements gleichberechtigt sind. In einem solchen Modell muß nun streng zwischen den Elementen des Ausgangsstapels und den Positionen im Zielstapel unterschieden werden.

Definition 4.3 $R = (G_A = (V_A, A_A), G_Z = (V_Z, A_Z), f, k)$ heißt genau dann *nicht eineindeutiges, verzweigtes Stapelproblem*, wenn

- i) G_A und G_Z verzweigte Stapel sind,
- ii) $k \geq 0$ eine natürliche Zahl ist,
- iii) V_A und V_Z disjunkt sind und $|V_A| = |V_Z|$ gilt und
- iv) f von V_A in die Potenzmenge von V_Z abbildet: $f : V_A \mapsto 2^{V_Z}$.

Eine solche Zuordnungsfunktion können wir uns als ungerichtete Kanten in einem Graphen denken, dessen Knotenmenge einerseits aus den Elementen des Ausgangsstapels und andererseits aus den Elementen des Zielstapels besteht. Zur Veranschaulichung diene hier wieder das Beispiel von Seite 65 mit der Zuordnungsfunktion $f : \{1, 2, 3, 4, 5, 6\} \mapsto 2^{\{a,b,c,d,e,f\}}$:



Definition 4.4 Ein nicht eineindeutiges, verzweigtes Stapelproblem $R = (G_A = (V_A, A_A), G_Z = (V_Z, A_Z), f, k)$ ist genau dann lösbar, wenn

- i) eine eindeutige Funktion $g : V_A \mapsto V_Z$ existiert, so daß für alle $v \in V_A$ $g(v) \in f(v)$ gilt und
- ii) $Q := (G_A, G' := (V_A, A), k)$ lösbares (eindeutiges) Stapelproblem ist, mit $A := \{(u, v) : (g(u), g(v)) \in A_Z\}$.

Wir führen somit die Lösbarkeit von nicht eineindeutigen Stapelproblemen auf eindeutige Probleme zurück, indem wir eine eindeutige Zuordnung von Elementen des Ausgangsstapels zu Elementen im Zielstapel fordern, so daß das resultierende eindeutige Stapelproblem lösbar ist.

Definition 4.5 Es seien $G_A = (V_A, A_A)$ und $G_Z = (V_Z, A_Z)$ Stapel und f mit $f : V_A \mapsto 2^{V_Z}$ eine Zuordnungsfunktion. V_A und V_Z seien disjunkt, dann ist der Zuordnungsgraph $G(f) := (V, E)$ bezüglich f wie folgt definiert:

- i) $V := V_A \cup V_Z$ und
- ii) $E := \{(u, v) : u \in V_A, v \in V_Z \text{ und } v \in f(u)\}.$

Definition 4.6 Es sei $G = (X, Y, A)$ ein paarer Graph und M ein perfektes Matching von G . Dann ist die Matchingfunktion f_M wie folgt definiert:

Für alle $u \in X$ mit $\{u, v\} \in M$ gilt $f_M(u) := v$.

Zur Lösung eines nicht eineindeutigen Stapelproblems R muß die Funktion f also eine eineindeutige Zuordnung g von Stapелеlementen des Ausgangsstapels zu Plätzen im Zielstapel zulassen. Eine solche Funktion g repräsentiert somit ein perfektes Matching im Zuordnungsgraphen $G(f)$. Damit können wir folgenden Satz formulieren.

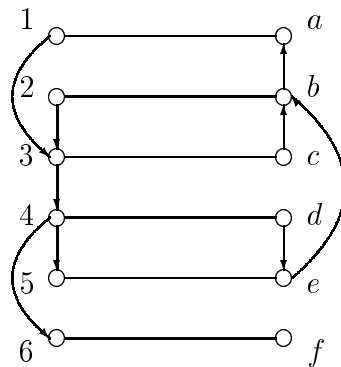
Satz 4.5 *Es sei $R = (G_A, G_Z, f, k)$ ein lösbares, nicht eineindeutiges, verzweigtes Stapelproblem. Dann gibt es in $G(f)$ ein perfektes Matching.*

Beweis: Nach Definition 4.4 gibt es eine eineindeutige Funktion g für die gilt:

- $g : V(G_A) \mapsto V(G_Z)$,
- für alle $v \in V(G_A)$ gilt $g(v) \in f(v)$.

Die Kantenmenge $M := \{\{u, g(u)\} : u \in V(G_A)\}$ ist offensichtlich ein perfektes Matching für $G(f)$. \square

Die Existenz eines solchen perfekten Matchings ist somit eine notwendige Voraussetzung für die Lösbarkeit eines nicht eineindeutigen, verzweigten Stapelproblems R . Das Beispiel von Seite 90 erfüllt diese Bedingung, denn es existiert eine eineindeutige Funktion $g : V(G_A) \mapsto V(G_Z)$ so, daß für alle $v \in V(G_A)$ $g(v) \in f(v)$ erfüllt ist:



$$\begin{aligned} g(1) &= a, \\ g(2) &= b, \\ g(3) &= c, \\ g(4) &= d, \\ g(5) &= e, \\ g(6) &= f. \end{aligned}$$

4.3 Auswahl einer eineindeutigen Zuordnung

Bevor also ein konkret formuliertes Stapelproblem auf seine Lösbarkeit hin untersucht wird, sollte überprüft werden, ob die vorgegebene Zuordnungsfunktion f eine Lösung überhaupt zuläßt. Nach einem Algorithmus von HOPCROFT und KARP [4] kann für paare Graphen $G = (V, E)$ ein maximum Matching M in $O(|V|^{\frac{1}{2}}|E|)$ Schritten bestimmt werden. Sollte für dieses gefundene Matching M $|M| = |V|/2$ gelten, so handelt es sich in unserem Fall – da $|V_A| = |V_Z|$ gilt – auch um ein perfektes Matching.

Satz 4.6 *Die Problemklasse*

$$NSP := \{R = (G_A = (V_A, A_A), G_Z = (V_Z, A_Z), f, k); \\ R \text{ lösbares, nicht eineindeutiges, verzweigtes Stapelproblem.}\}$$

ist NP-vollständig.

Beweis: Satz 4.6 folgt unmittelbar aus Satz 3.5. \square

Zur Lösung des vorgegebenen Problems kann dieses auf eine Anzahl von Problemen reduziert werden, bei denen die Zuordnung der Elemente zu den Plätzen im Stapel eindeutig ist. Da es jedoch im schlechtesten Fall ($G(f) = K_{n,n}$) $n!$ viele perfekte Matchings in $G(f)$ gibt, ist bereits an dieser Stelle ein exponentieller Zeitaufwand notwendig. Im folgenden wird uns nun interessieren, ob sich auf andere Art in polynomialer Zeit ein perfektes Matching M für $G(f)$ finden läßt, für das das daraus resultierende eineindeutige Stapelproblem mit minimaler Hilfsstapelanzahl lösbar ist.

Lemma 4.7 Es sei $R = (G_A = (V_A, A_A), G_Z = (V_Z, A_Z), f, k)$ ein (formal) nicht eineindeutiges, verzweigtes Stapelproblem, so daß die Kanten von $G(f)$ ein perfektes Matching E für $G(f)$ bilden. Dann gibt es eine *Erweiterung* $R' = (G'_A = (V'_A, A'_A), G'_Z = (V'_Z, A'_Z), f', k)$ von R , so daß R' ebenfalls ein nicht eineindeutiges, verzweigtes Stapelproblem ist, das die folgenden Eigenschaften erfüllt:

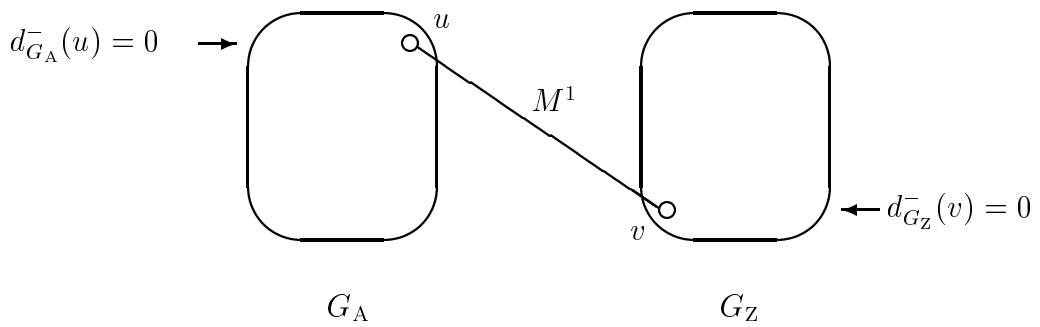
1. $G(f')$ hat maximal zwei voneinander verschiedene perfekte Matchings E' und M' (gibt es in $G(f')$ nur ein perfektes Matching, so gilt $E' = M'$).
2. Es gilt $V_A \subseteq V'_A, V_Z \subseteq V'_Z$ und $E \subseteq E'$.
3. $R'' = (G'_A, G'_Z, f_{E'}, k)$ ist genau dann lösbar, wenn R lösbar ist, wobei $f_{E'}$ die Matchingfunktion von E' ist.
4. $R''' = (G'_A, G'_Z, f_{M'}, 0)$ ist lösbar, wobei $f_{M'}$ die Matchingfunktion von M' ist,
5. R', E' und M' können in Polynomialzeit bestimmt werden.

Beweis: Um Lemma 4.7 zu beweisen, zeigen wir, wie sich perfekte Matchings E' und M' mit den geforderten Eigenschaften beschaffen lassen.

M^0 sei zunächst leer. Sukzessive fügen wir nun Kanten zu M^0 hinzu. Dazu betrachten wir G_A und G_Z und wählen ein Element u der Spitze des Ausgangsstapels G_A und ein Element v , das im Zielstapel G_Z ganz unten liegt. Wir ordnen dem Element u den Platz von v zu, das heißt, wir fügen die Kante $\{u, v\}$ zur Menge M^0 hinzu, falls

- dadurch im Graphen $G(f_E) \cup G(f_{M^0 \cup \{u,v\}})$ der durch die Kantenmenge $E \cup M^0 \cup \{u, v\}$ repräsentiert wird, kein Kreis entsteht oder
- u das einzige Element des Ausgangsstapels und v das einzige Element des Zielstapels ist.

$$M^1 := M^0 \cup \{u, v\}, \quad E^1 := E$$

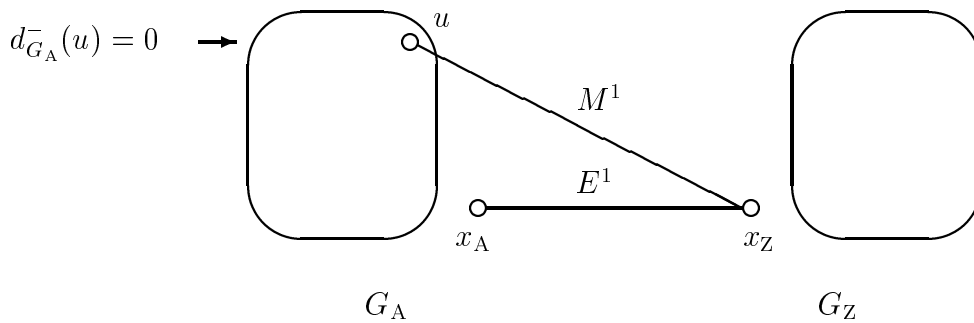


Sollte sich kein solches Paar u, v finden, so wählen wir einen beliebigen Knoten u der Spitze des Ausgangsstapels G_A und fügen ein neues Element x_A zum Ausgangsstapel G_A und ein neues Element x_Z zum Zielstapel G_Z hinzu:

$$G_A := G_A \cup (\{x_A\}, \emptyset), \quad G_Z := G_Z \cup (\{x_Z\}, \emptyset).$$

Diese Knoten sind jeweils isolierte Knoten in den Graphen. Entsprechend müssen nun E und M^0 erweitert werden:

$$E^1 := E \cup \{\{x_A, x_Z\}\}, \quad M^1 := M^0 \cup \{\{u, x_Z\}\}.$$



Durch diese Konstruktion wird gesichert, daß im Graphen $G(f_{E^1}) \cup G(f_{M^1})$ der durch die Matchingkanten in $E^1 \cup M^1$ bestimmt wird, kein Kreis entsteht.

Die mit der zu M' hinzugefügten Kante inzidenten Knoten u und v bzw. u und x_Z werden im Ausgangsstapel G_A und im Zielstapel G_Z mit allen inzidenten Kanten gelöscht:

$$G_A^1 := G_A \setminus \{u\}, \quad G_Z^1 := G_Z \setminus \{v\} \quad \text{bzw.} \quad G_Z^1 := G_Z \setminus \{x_Z\}.$$

Auf diese Art läßt sich Schritt für Schritt für jedes Element der Stapel eine inzidente Kante in einem Matching M^l beschaffen. Dabei kann das Verfahren so angewendet werden, daß nach j Schritten nur noch neue Elemente vom Typ x_A im Ausgangsstapel G_A^j auftreten. Diese können dann ohne Rücksicht auf Ausstapelungen (die x_A liegen ja isoliert in G_A^j) so den Elementen im Zielstapel G_Z^j zugeordnet werden, daß keine Kreise in dem Graphen, der durch diese Zuordnungskanten und die Kanten aus E^j und M^j definiert ist, entstehen. (Da jeder Knoten mit maximal 2 Kanten aus $E^j \cup M^j$ inzident ist, können in $E^j \cup M^j$ nur Pfade, keine Baumstrukturen auftreten.) Durch dieses Vorgehen wird gesichert, daß das Verfahren nach l Schritten abbricht und $G(f_{E^l}) \cup G(f_{M^l})$ maximal einen Kreis enthält. Die so erhaltenen, um neue Knoten vom Typ x_A und x_Z erweiterten Ausgangs- und Zielstapel G_A^l und G_Z^l nennen wir G'_A und G'_Z . Weiterhin sei $E' := E^l$ und $M' := M^l$.

Gleichzeitig wurde sichergestellt, daß mit der Zuordnung der Elemente des Ausgangsstapels zu den Elementen des Zielstapels, die das perfekte Matching M' repräsentiert, der Ausgangsstapel in den Zielstapel ohne Ausstapelungen überführt werden kann, da durch die mit M' definierte Zuordnung immer ein Element der Spitze des Ausgangsstapels direkt auf den Zielstapel gelegt werden kann:

$$R''' := (G'_A, G'_Z, f_{M'}, 0) \text{ ist lösbar.}$$

Das perfekte Matching E für das ursprüngliche Stapelproblem R wurde so erweitert, daß die jeweils zu Ausgangs- und Zielstapel neu hinzugekommenen Elemente x_A und x_Z in E' einander zugeordnet wurden. Da diese Elemente sowohl im Ausgangsstapel G'_A als auch im Zielstapel G'_Z immer isoliert liegen, ist der neue Ausgangsstapel G'_A mit der gleichen Anzahl an Ausstapelungen in G'_Z überführbar, wie G_A in G_Z überführt werden kann:

$$R'' := (G'_A, G'_Z, f_{E'}, k) \text{ ist lösbar} \iff R \text{ ist lösbar.}$$

Wir definieren nun f' so, daß

$$\text{für alle } v \in V(G'_A) \quad f'(v) := \{f_{E'}(v), f_{M'}(v)\} \text{ gilt.}$$

E' und M' sind offensichtlich in polynomialer Zeit berechenbar. Damit kann natürlich auch $R' := (G'_A, G'_Z, f', k)$ in Polynomialzeit bestimmt werden. \square

Das folgende Beispiel soll die Konstruktion, welche im Beweis von Lemma 4.7

verwendet wird, illustrieren:

Beispiel:

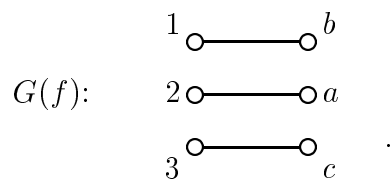
$$G_A = (\{1, 2, 3\}, \{(1, 2), (2, 3)\}), \quad G_Z = (\{a, b, c\}, \{(a, b), (b, c)\}),$$



$$\begin{aligned} f(1) &= \{b\}, & E &= \{\{1, b\}, \{2, a\}, \{3, c\}\}, & k &= 1 \\ f(2) &= \{a\}, \\ f(3) &= \{c\}. \end{aligned}$$

Dann ist $R = (G_A, G_Z, f, k)$ ein (formal) nicht eindeutiges Stapelproblem und E bildet ein perfektes Matching für $G(f)$:

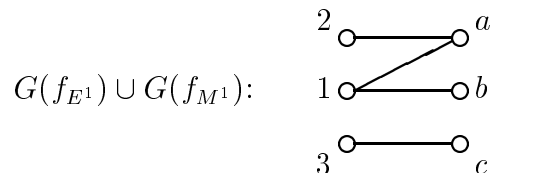
$$G(f) = (\{1, 2, 3, a, b, c\}, \{\{1, b\}, \{2, a\}, \{3, c\}\}),$$



Es sei $M^0 = \emptyset$. Nun fügen wir folgende Kante zu M hinzu: $M^1 := M^0 \cup \{\{1, a\}\}$. Das perfekte Matching E bleibt unverändert: $E^1 := E$. Löschen wir die Knoten 1 und a mit allen inzidenten Kanten aus G_A und G_Z , so erhalten wir:

$$G_A^1 := (\{2, 3\}, \{(2, 3)\}) \quad \text{und} \quad G_Z^1 := (\{b, c\}, \{(b, c)\}).$$

Der Zuordnungsgraph $G(f_{E^1}) \cup G(f_{M^1})$, der durch die Kanten der Mengen E^1 und M^1 gebildet wird, bleibt kreisfrei:

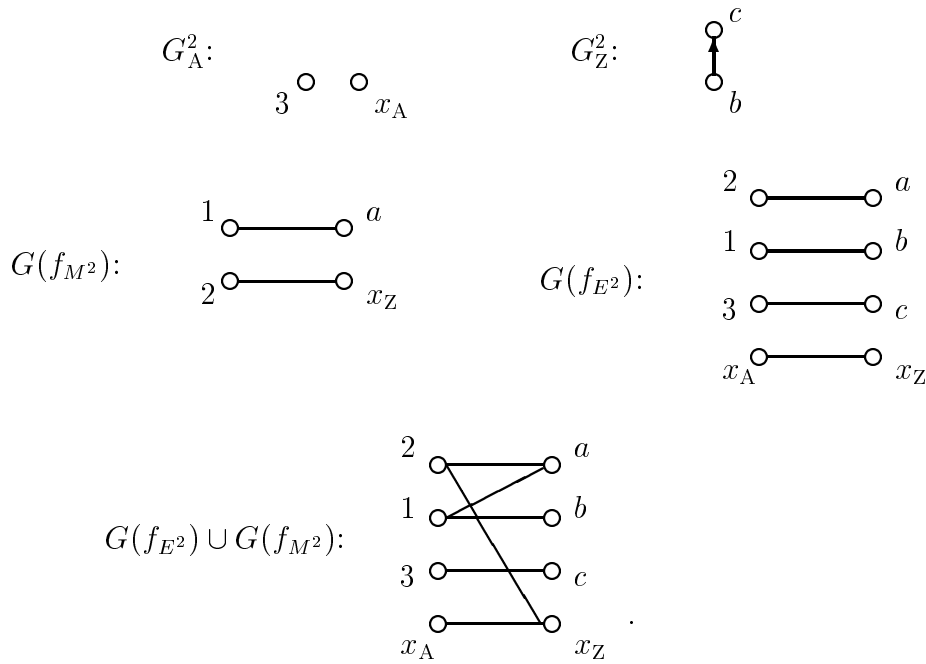


Für die neuen Ausgangs- und Zielstapel findet sich zwar wieder ein Elementepaar $(2, b)$, so daß 2 auf der Spitze des Ausgangsstapels G_A und b im Zielstapel G_Z ganz unten liegt, jedoch würde der Zuordnungsgraph $G(f_{E^1}) \cup G(f_{M^1 \cup \{2, b\}})$ einen Kreis

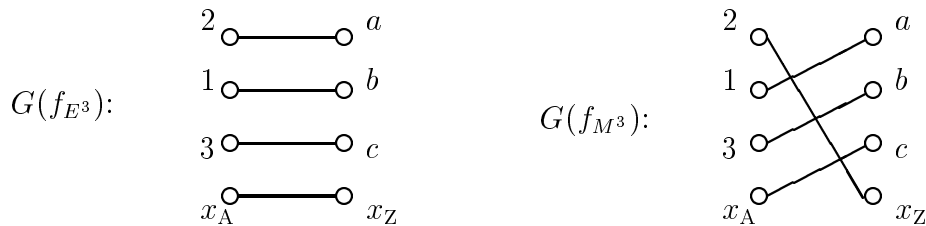
enthalten. Entsprechend dem Beweis von Lemma 4.7 fügen wir nun ein Element x_A zum Ausgangsstapel G_A^1 und ein Element x_Z zum Zielstapel G_Z^1 hinzu:

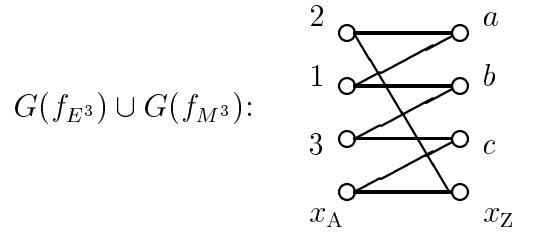
$$G_A^1 := G_A^1 \cup (\{x_A\}, \emptyset), \quad G_Z^1 := G_Z^1 \cup (\{x_Z\}, \emptyset).$$

Anschließend nehmen wir die Kante $\{x_A, x_Z\}$ im Matching E^1 auf: $E^2 := E^1 \cup \{\{x_A, x_Z\}\}$. Gleichzeitig können wir die Kante $\{x_A, b\}$ zu M^1 hinzufügen, ohne daß sich in $G(f_{E^2}) \cup G(f_{M^1 \cup \{x_A, b\}})$ dadurch ein Kreis bildet: $M^2 := M^1 \cup \{x_A, b\}$. Nun löschen wir die Knoten 2 und x_Z mit allen inzidenten Kanten aus G_A^1 bzw G_Z^1 und erhalten



Nun liegen alle in G_A^2 verbleibenden Elemente „ganz oben“ im Stapel. Im Zielstapel G_Z^2 steht b als ganz unten liegendes Element zur Verfügung. Wir können nun entweder die Kante $\{3, b\}$ oder die Kante $\{x_A, b\}$ in M^2 aufnehmen. Wir entscheiden uns für die Kante $\{3, b\}$: $M^3 := M^2 \cup \{\{3, b\}\}$. Nach dem Löschen der Knoten 3 und b in den Stapeln G_A^2 und G_Z^2 verbleiben nun noch jeweils ein isolierter Knoten x_A im Ausgangsstapel G_A^3 und das Element c im Zielstapel G_Z^3 . Da diese beiden Elemente die einzigen sind, können wir die Kante $\{x_A, c\}$ in M^2 aufnehmen, obwohl damit im Graphen $G(f_{E^3}) \cup G(f_{M^2 \cup \{x_A, c\}})$ ein Kreis entsteht. Mit $M^3 := M^2 \cup \{\{x_A, c\}\}$ ergibt sich somit

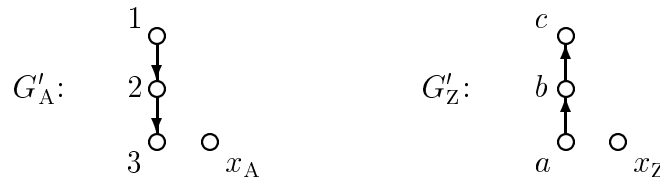




Nun sind $G'_A := G_A \cup (\{x_A\}, \emptyset)$, $G'_Z := G_Z \cup (\{x_Z\}, \emptyset)$, $E' := E^3$, $M' := M^3$ und f' mit

$$\text{für alle } v \in V(G'_A) \text{ gilt } f'(v) := \{f_{E'}(v), f_{M'}(v)\}$$

die im Lemma 4.7 beschriebenen Stapel, Matchings und Zuordnungsfunktion.



Die gewünschte Erweiterung ist $R' := (G'_A, G'_Z, f', 1)$. Begründung: Nach Lemma 4.3 enthält $G(f')$ nur die beiden perfekten Matchings E' und M' . Das nicht-eindeutige Stapelproblem $R'' := (G'_A, G'_Z, f'_{E'}, 1)$ ist lösbar, da, um das Element x_A des Ausgangsstapels G'_A auf die Position x_Z des Zielstapels zu bringen, keine zusätzliche Ausstapelung notwendig ist. Gleichzeitig ist das nicht-eindeutige Stapelproblem $R''' := (G'_A, G'_Z, f'_{M'}, 0)$ lösbar, da hier mit der Zuordnung, die durch M' festgelegt ist, keine Ausstapelung erforderlich wird, um den Ausgangsstapel G_A in den Zielstapel G_Z zu überführen.

Lemma 4.8 Es sei $R = (G_A = (V_A, A_A), G_Z = (V_Z, A_Z), f, k)$ ein (formal) nicht-eindeutiges, verzweigtes Stapelproblem, so daß die Kanten von $G(f)$ ein perfektes Matching E für $G(f)$ bilden. Dann gibt es eine *Erweiterung* $R' = (G'_A = (V'_A, A'_A), G'_Z = (V'_Z, A'_Z), f', 2k)$ von R , so daß R' ebenfalls ein nicht-eindeutiges, verzweigtes Stapelproblem ist, das die folgenden Eigenschaften erfüllt:

1. $G(f')$ hat maximal zwei voneinander verschiedene perfekte Matchings E' und M' (gibt es in $G(f')$ nur ein perfektes Matching, so gilt $E' = M'$).
2. Es gilt $V_A \subseteq V'_A$, $V_Z \subseteq V'_Z$ und $E \subseteq E'$.
3. $R'' = (G'_A, G'_Z, f_{E'}, 2k)$ ist genau dann lösbar, wenn R lösbar ist, wobei $f_{E'}$ die Matchingfunktion von E' ist.
4. $R''' = (G'_A, G'_Z, f_{M'}, 2k + 1)$ ist lösbar, wobei $f_{M'}$ die Matchingfunktion von M' ist und k minimal.

5. R', E' und M' können in Polynomialzeit bestimmt werden.

Beweis: Zunächst ersetzen wir G_A durch zwei Kopien von G_A und ersetzen gleichzeitig G_Z durch zwei Kopien von G_Z . Damit erhalten wir ein nicht eindeutiges Stapelproblem $R^0 = (H_A, H_Z, f, 2k)$:

$$G'_A, G''_A \cong G_A, \quad V(G'_A) \cap V(G''_A) = \emptyset, \quad H_A := G'_A \cup G''_A$$

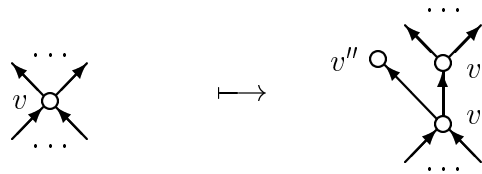
$$G'_Z, G''_Z \cong G_Z, \quad V(G'_Z) \cap V(G''_Z) = \emptyset, \quad H_Z := G'_Z \cup G''_Z, \text{ mit}$$

$$(V(G'_Z) \cup V(G''_Z)) \cap (V(G'_A) \cup V(G''_A)) = \emptyset,$$

wobei die Zuordnungsfunktion f ebenfalls um die entsprechenden Zuordnungen erweitert wurde, so daß R^0 eine „Verdoppelung“ des Problems R ist. Offensichtlich ist R^0 genau dann lösbar, wenn R es ist.

Nun läßt sich Lemma 4.7 auf R^0 anwenden und wir erhalten eine Erweiterung $R^1 = (G^1_A, G^1_Z, f^1, 2k)$ von R^0 , so daß $G(f^1)$ maximal zwei voneinander verschiedene perfekte Matchings E^1 und M^1 enthält, $H_A \subseteq G^1_A$ und $H_Z \subseteq G^1_Z$ und $E \subseteq E^1$ gilt. Desweiteren ist $R^{1'} := (G^1_A, G^1_Z, f_{E^1}, 2k)$ genau dann lösbar, wenn R^0 lösbar ist und außerdem ist $R^{1''} := (G^1_A, G^1_Z, f_{M^1}, 0)$ lösbar.

Nun erweitern wir das Problem R^1 wie folgt: Sofern es im Matching M^1 eine Kante $\{u, v\}$ gibt, die nicht in E^1 liegt, wählen wir diese, anderenfalls betrachten wir eine beliebige Kante $\{u, v\} \in M^1 = E^1$. Dabei soll u aus dem Ausgangsstapel G^1_A und v aus dem Zielstapel G^1_Z stammen. Wir fügen nun Elemente u' und u'' zum Ausgangsstapel G^1_A und Elemente v' und v'' zum Zielstapel G^1_Z hinzu. Dies soll derart geschehen, daß u'' im Ausgangsstapel auf u' liegt und die entstehende gerichtete Kante (u'', u') im Ausgangsstapel G^1_A isoliert bleibt. Im Zielstapel G^1_Z soll v'' nur auf v liegen – v' dagegen soll auf v liegen und unter allen denjenigen Elementen des Zielstapels G^1_Z , die ursprünglich auf v lagen. Das neue Element v' „übernimmt“ gewissermaßen alle ursprünglich aus v ausgehenden Kanten:

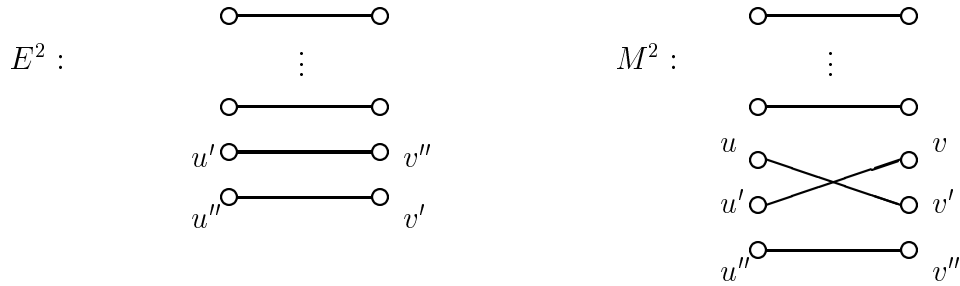


Durch die Hinzunahme dieser vier neuen Elemente erhalten wir einen neuen Ausgangsstapel G^2_A und einen neuen Zielstapel G^2_Z . Nun müssen die neuen Elemente u', u'', v' und v'' sowohl in das perfekte Matching M^1 als auch in E^1 aufgenommen werden:

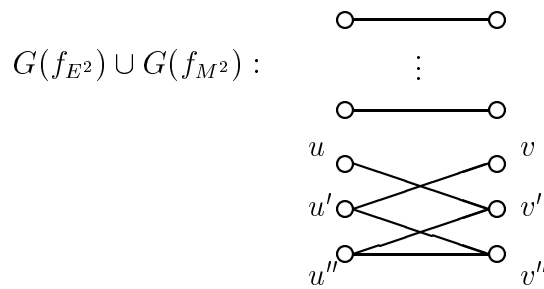
$$E^2 := E^1 \cup \{\{u', v''\}, \{u'', v'\}\}.$$

Das perfekte Matching M^1 erhalten wir wie folgt: In M^1 streichen wir die Kante $\{u, v\}$ und fügen stattdessen $\{u, v'\}, \{u', v\}$ und $\{u'', v''\}$ hinzu:

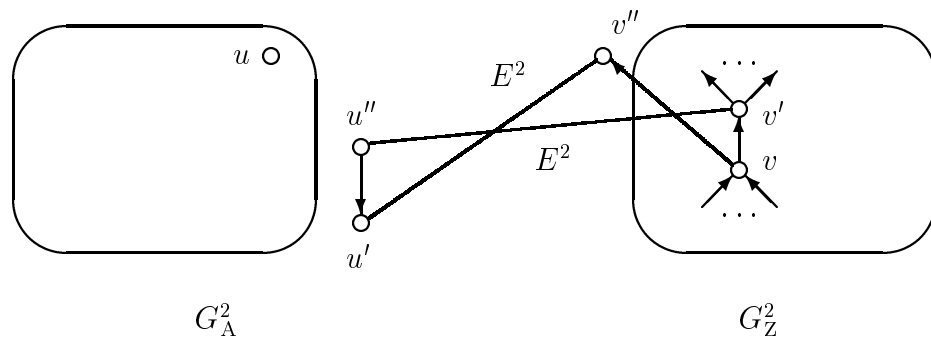
$$M^2 := (M^1 \setminus \{u, v\}) \cup \{\{u, v'\}, \{u', v\}, \{u'', v''\}\}.$$



Dadurch wird sichergestellt, daß im Graphen $G(f_{E^2}) \cup G(f_{M^2})$, der durch die beiden Matchings M^2 und E^2 repräsentiert wird, nur dann ein Kreis entstehen kann, wenn ursprünglich $E^1 = M^1$ galt. Somit enthält $G(f_{E^2}) \cup G(f_{M^2})$ nur genau dann diesen einen Kreis. Gilt jedoch $\{u, v\} \notin E$, so wird kein zusätzlicher Kreis erzeugt, da nur die Kante $\{u, v\}$ mit Hilfe neuer Knoten zu einem Pfad $[u, v', u'', v'', u', v]$ erweitert wurde.

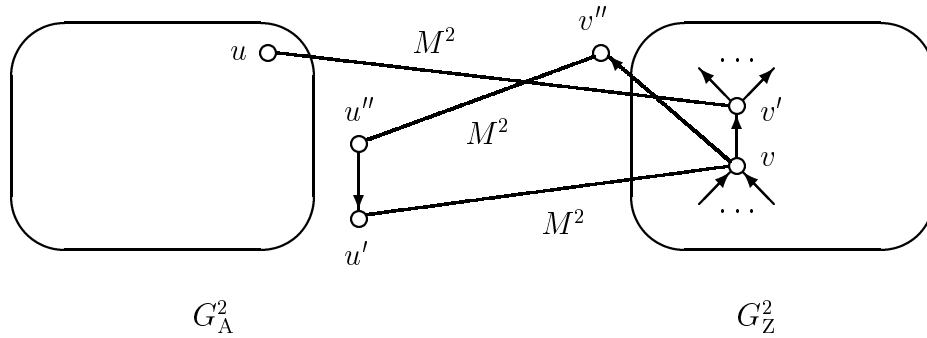


Die Anzahl notwendiger Ausstapelungen für $R^{2'} := (G_A^2, G_Z^2, f_{E^2}, 2k)$ hat sich gegenüber R^1 durch die oben beschriebene Operation nicht geändert:



Das Element u'' kann zu jeder Zeit vom Ausgangsstapel entnommen werden. Um den neuen Zielstapel vollständig aufzubauen, genügt es, u' zuletzt auf G_Z^2 zu legen. Ansonsten kann wie zur Lösung von R^1 verfahren werden. Daher sind keine zusätzlichen Hilfsstapelplätze zur Lösung von $R^{2'}$ notwendig.

Zur Lösung von $R^{2''} := (G_A^2, G_Z^2, f_{M^2}, 1)$ erhöht sich dagegen die Anzahl notwendiger Ausstapelungen um einen Knoten gegenüber $R^{1''}$:



An der Stelle in einer für das Problem $R^{1''}$ optimalen Umstapelfolge, an der u auf den Zielstapelplatz v gebracht wurde, ist nun zunächst das Element u' dorthin zu bringen. Das ist jedoch nur möglich, wenn vorher u'' auf einen Hilfsstapelplatz ausgestapelt wurde. Anschließend kann weiter wie zur Lösung von $R^{1''}$ verfahren werden – zum Schluß wird u'' auf den Zielstapel gebracht.

Durch diese Erweiterung von R^1 zu $R^2 := (G_A^2, G_Z^2, f^2, 2k)$ mit f^2 so, daß

$$\text{für alle } v \in V(G_A^2) \quad f^2(v) := \{f_{E^2}(v), f_{M^2}(v)\} \text{ gilt,}$$

wurde erreicht, daß sich die Anzahl notwendiger Ausstapelungen zur Lösung von $R^{2''}$ gegenüber $R^{1''}$ um eins erhöht, während $R^{2'}$ genau dann lösbar bleibt, wenn $R^{1'}$ lösbar ist.

Durch wiederholte Anwendung dieses Verfahrens können wir Schritt für Schritt zu nicht eineindeutigen Stapelproblemen

$$R^{2k+1'} = (G_A^{2k+1}, G_Z^{2k+1}, f_{E^{2k+1}}, 2k) \text{ und } R^{2k+1''} = (G_A^{2k+1}, G_Z^{2k+1}, f_{M^{2k+1}}, 2k + 1)$$

gelangen, so daß $R^{2k+1'}$ genau dann lösbar ist, wenn $R^{1'}$ lösbar ist und $R^{2k+1''}$ mit genau $2k + 1$ Ausstapelungen gelöst werden kann. Gleichzeitig bleibt gesichert, daß der Graph, der durch $E^{2k+1} \cup M^{2k+1}$ repräsentiert wird, höchstens einen Kreis enthält. Damit sind die im Lemma 4.8 geforderten Stapel $G'_A := G_A^{2k+1}$ und $G'_Z := G_Z^{2k+1}$ gefunden. Die gesuchte Zuordnungsfunktion f' ergibt sich wie folgt:

$$\text{Für alle } v \in V(G_A^{2k+1}) \text{ sei } f'(v) := \{f_{E^{2k+1}}(v), f_{M^{2k+1}}(v)\}.$$

Nach der Konstruktion und Lemma 4.3 ist klar, daß $G(f')$ genau die beiden perfekten Matchings $E' := E^{2k+1}$ und $M' := M^{2k+1}$ enthält. Damit ist $R' := (G'_A, G'_Z, f', 2k)$ die geforderte Erweiterung von R . \square

Wir wollen nun an einem Beispiel verdeutlichen, wie für ein gegebenes, formal

nicht eindeutiges Stapelproblem R die in Lemma 4.8 beschriebene Erweiterung R' nach der Konstruktion des Beweises bestimmen können.

Beispiel: Wir betrachten ein Stapelproblem R , das durch die folgenden Ausgangs- und Zielstapel



sowie durch die Zuordnungsfunktion f definiert ist:

$$f(1) = \{b\}, \quad f(2) = \{a\}.$$

Damit sieht E wie folgt aus: $E = \{\{1, b\}, \{2, a\}\}$.

$$R := (G_A = (\{1, 2\}, \{(1, 2)\}), G_Z = (\{a, b\}, \{(a, b)\}), f, 1)$$

G_A kann in G_Z überführt werden – zur Lösung dieses Problems ist nur eine Ausstapelung notwendig, deshalb gilt $k = 1$. Wie im Beweis von Lemma 4.8 beschrieben, ersetzen wir R durch eine „Verdoppelung“ des Problems:

$$R^0 := (G_A = (\{1, 2, 3, 4\}, \{(1, 2), (3, 4)\}), G_Z = (\{a, b, c, d\}, \{(a, b), (c, d)\}), f^0, 2)$$

mit

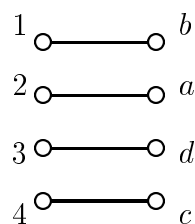
$$f^0(1) = \{b\}, \quad f^0(2) = \{a\}, \quad f^0(3) = \{d\}, \quad f^0(4) = \{c\}.$$



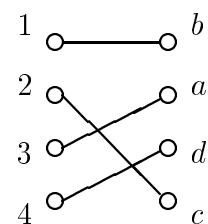
Für E gilt dann entsprechend $E = \{\{1, b\}, \{2, a\}, \{3, d\}, \{4, c\}\}$. Um R^0 lösen zu können sind damit doppelt so viele Ausstapelungen notwendig, wie zur Lösung von R , damit gilt $k = 2$. Die Anwendung von Lemma 4.7 auf R^0 gestaltet sich recht einfach: Es müssen keine zusätzlichen Elemente zu den Stapeln hinzugefügt werden, um die perfekten Matchings E^1 und M^1 zu erhalten, wie sie in Lemma 4.7 gefordert werden.

$R^1 := (G_A^1 := G_A, G_Z^1 := G_Z, f^1, 2)$, wobei $G(f^1)$ genau diejenigen Kanten enthält, die in E^1 und M^1 auftreten:

$$E^1 := E:$$



$$M^1 := \{\{1, b\}, \{2, c\}, \{3, a\}, \{4, d\}\}:$$



$$f^1(1) = \{b\}, f^1(2) = \{a, c\}, f^1(3) = \{a, d\}, f^1(4) = \{c, d\}.$$

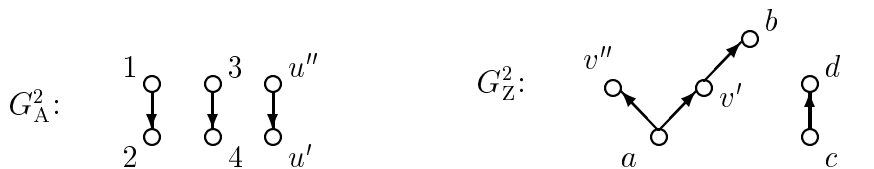
Der Zuordnungsgraph $G(f^1)$, der durch die Kantenmenge $E^1 \cup M^1$ beschrieben wird, enthält genau einen Kreis: $[2, a, 3, d, 4, c, 2]$. Da $R^{1'} := (G_A^1, G_Z^1, f_{E^1}, 2)$ mit dem Problem R^0 identisch ist, ist es auch genau dann lösbar, wenn es R^0 ist. $R^{1''} := (G_A^1, G_Z^1, f_{M^1}, 0)$ ist ohne Ausstapelungen lösbar, weil das Element 3 des Ausgangsstapels sofort auf den Platz a im Zielstapel gelegt werden kann, anschließend können die Elemente 1 auf den Platz b , 2 auf den Platz c und 4 auf den Platz d im Zielstapel gelegt werden.

Nun erweitern wir R^1 im Sinne von Lemma 4.8: Als Matchingkanten $\{u, v\}$, die in M^1 aber nicht in E^1 liegen, kommen hier $\{2, c\}$, $\{3, a\}$ und $\{4, d\}$ in Frage. Wir entscheiden uns für die Kante $\{3, a\}$. Fügen wir die Elemente u' und u'' zum Ausgangsstapel G_A^1 und v' sowie v'' zum Zielstapel G_Z^1 in der im Beweis von Lemma 4.8 beschriebenen Weise hinzu, so erhalten wir

$$R^2 := (G_A^2, G_Z^2, f^2, 2) \text{ mit}$$

$$G_A^2 := (\{1, 2, 3, 4, u', u''\}, \{(1, 2), (3, 4), (u'', u')\})$$

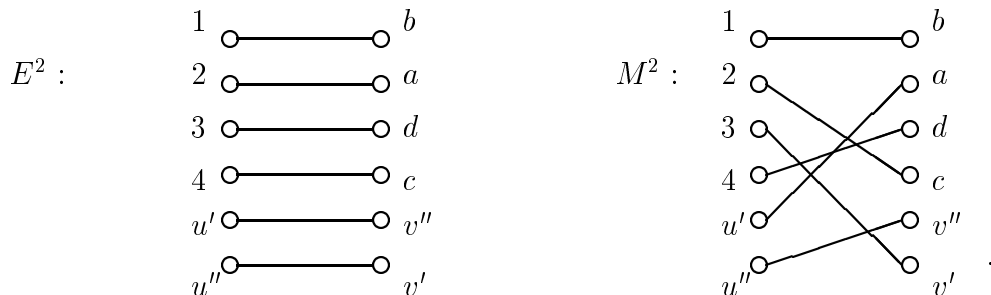
und $G_Z^2 := (\{a, b, c, d, v', v''\}, \{(a, v'), (a, v''), (v', b), (c, d)\})$



Entsprechend der Vorgehensweise im Beweis von Lemma 4.8 erhalten wir

$$E^2 := \{\{1, b\}, \{2, a\}, \{3, d\}, \{4, c\}, \{u', v''\}, \{u'', v'\}\}$$

und $M^2 := \{\{1, b\}, \{2, c\}, \{3, v'\}, \{4, d\}, \{u', a\}, \{u'', v''\}\} :$



Für f^2 gilt: Für alle $v \in V(G_A^2)$ ist $f^2(v) := \{f_{E^2}(v), f_{M^2}(v)\}$.

$$\begin{aligned} f^2(1) &= \{b\}, & f^2(2) &= \{a, c\}, & f^2(3) &= \{d, v'\} \\ f^2(4) &= \{c, d\}, & f^2(u') &= \{a, v''\}, & f^2(u'') &= \{v', v''\} \end{aligned}$$

Der Graph der durch die Kantenmenge $E^2 \cup M^2$ beschrieben wird, enthält genau einen Kreis: $[2, a, u', v'', u'', v', 3, d, 4, c, 2]$.

$R^{2'} := (G_A^2, G_Z^2, f_{E^2}, 2)$ ist genau dann lösbar, wenn $R^{1'}$ es ist: Zunächst werden die Elemente 1 und 3 des Ausgangsstapels G_A^2 ausgestapelt, anschließend kann z.B. Element 2 nach Platz a im Zielstapel, 4 nach c , 3 nach d , u'' nach v' und u' nach v'' gebracht werden. Zum Schluß kann das Element 1 von einem Hilfsstapelplatz auf die Position b im Zielstapel gelegt werden. Zur Lösung von $R^{2'}$ sind somit genau 2 Ausstapelungen notwendig.

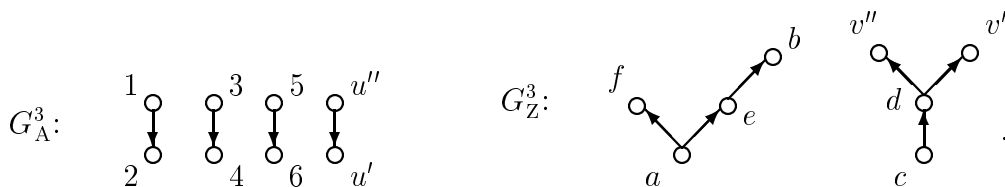
Um das Problem $R^{2''} := (G_A^2, G_Z^2, f_{M^2}, 1)$ zu lösen, ist jetzt eine Ausstapelung notwendig: Das Element 3 des Ausgangsstapels kann jetzt nicht mehr auf Position a im Zielstapel gebracht werden – dorthin kann jetzt nur u' gelegt werden, so daß u'' ausgestapelt werden muß.

Aus Gründen der Übersichtlichkeit nennen wir die Elemente u'' und u' jetzt 5 und 6, sowie v' und v'' jetzt e und f :

$$\begin{aligned} G_A^2 &:= (\{1, 2, 3, 4, 5, 6\}, \{(1, 2), (3, 4), (5, 6)\}) \\ \text{und } G_Z^2 &:= (\{a, b, c, d, e, f\}, \{(a, e), (a, f), (e, b), (c, d)\}). \end{aligned}$$

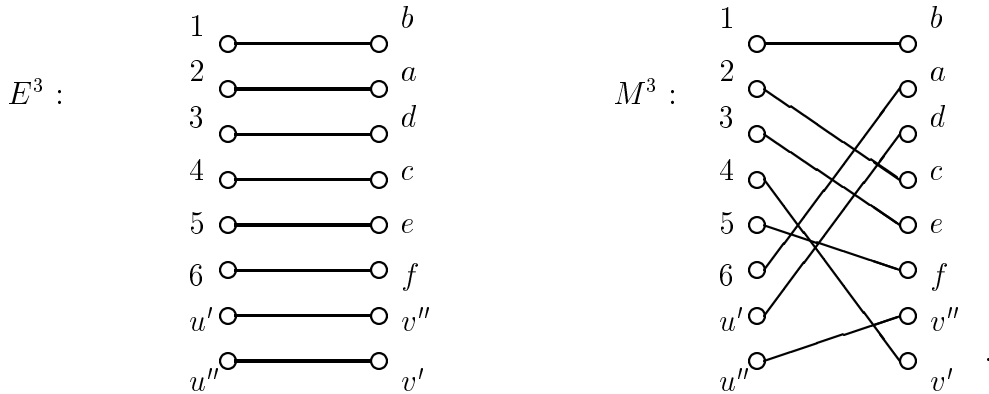
Nun erweitern wir R^2 nochmals im Sinne von Lemma 4.8: Als Matchingkante $\{u, v\}$, die in M^2 aber nicht in E^2 liegt, wählen wir nun beispielsweise $\{4, d\}$. Wiederum fügen wir neue Elemente u', u'' zum Ausgangsstapel und v', v'' zum Zielstapel hinzu und erhalten $R^3 := (G_A^3, G_Z^3, f^3, 2)$ mit

$$\begin{aligned} G_A^3 &:= (\{1, 2, 3, 4, 5, 6, u', u''\}, \{(1, 2), (3, 4), (5, 6), (u'', u')\}) \\ \text{und } G_Z^3 &:= (\{a, b, c, d, e, f, v', v''\}, \{(a, e), (e, b), (a, f), (c, d), (d, v'), (d, v'')\}) \end{aligned}$$



Die perfekten Matchings M^2 und E^2 werden wie folgt ergänzt:

$$\begin{aligned} E^3 &:= \{\{1, b\}, \{2, a\}, \{3, d\}, \{4, c\}, \{5, e\}, \{6, f\}, \{u', v''\}, \{u'', v'\}\} \\ \text{und } M^3 &:= \{\{1, b\}, \{2, c\}, \{3, e\}, \{4, v'\}, \{5, f\}, \{6, a\}, \{u', d\}, \{u'', v''\}\} : \end{aligned}$$



Für f^3 gilt: Für alle $v \in V(G_A^3)$ ist $f^3(v) := \{f_{E^3}(v), f_{M^3}(v)\}$:

$$\begin{aligned}
 f^3(1) &= \{b\}, & f^3(2) &= \{a, c\}, & f^3(3) &= \{d, e\}, & f^3(4) &= \{c, v'\} \\
 f^3(5) &= \{e, f\}, & f^3(6) &= \{a, f\}, & f^3(u') &= \{v'', d\}, & f^3(u'') &= \{v', v''\}.
 \end{aligned}$$

Der durch die Kantenmenge $E^3 \cup M^3$ repräsentierte Graph enthält genau einen Kreis: $[2, a, 6, f, 5, e, 3, d, u', v'', u'', v', 4, c, 2]$.

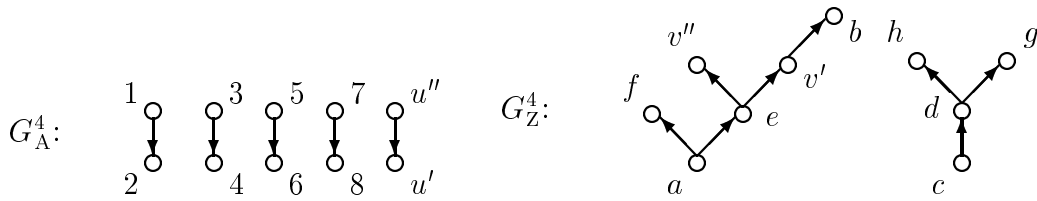
$R^{3'} := (G_A^3, G_Z^3, f_{E^3}, 2)$ ist – genau wie $R^{2'}$ – mit 2 Ausstapelungen lösbar: Die Elemente 1 und 3 des Ausgangsstapels müssen auf Hilfsstapelplätze ausgestapelt werden, anschließend können alle Elemente auf die durch E^3 zugeordneten Positionen im Zielstapel gelegt werden.

Um den Ausgangsstapel des Problems $R^{3''} := (G_A^3, G_Z^3, f_{M^3}, 2)$ in den Zielstapel überführen zu können, ist nun eine Ausstapelung mehr als zur Lösung des Problems $R^{2''}$ notwendig: Bevor das Element 6 des Ausgangsstapels G_A^3 auf die Position a in G_Z^3 gelegt werden kann, muß das Element 5 aus G_A^3 ausgestapelt werden und bevor u' auf Position d gebracht werden kann, muß u'' ausgestapelt werden. Auch hier benennen wir wieder der Übersichtlichkeit wegen die Elemente u'' und u' in 7 und 8 und v' und v'' in g und h um:

$$\begin{aligned}
 G_A^3 &:= (\{1, 2, 3, 4, 5, 6, 7, 8\}, \{(1, 2), (3, 4), (5, 6), (7, 8)\}) \\
 \text{und } G_Z^3 &:= (\{a, b, c, d, e, f, g, h\}, \{(a, e), (e, b), (a, f), (c, d), (d, g), (d, h)\}).
 \end{aligned}$$

Führen wir nun die Erweiterung aus Lemma 4.8 ein letztes Mal durch, so erhalten wir für $\{u, v\} = \{3, e\}$ $R^4 := (G_A^4, G_Z^4, f^4, 3)$ mit

$$\begin{aligned}
 G_A^4 &:= (\{1, 2, 3, 4, 5, 6, 7, 8, u', u''\}, \{(1, 2), (3, 4), (5, 6), (7, 8), (u'', u')\}) \\
 \text{und } G_Z^4 &:= (\{a, b, c, d, e, f, g, h, v', v''\}, \{(a, e), (e, v''), (e, v'), (a, f), (v', b), \\
 &\hspace{15em} (c, d), (d, g), (d, h)\}) :
 \end{aligned}$$

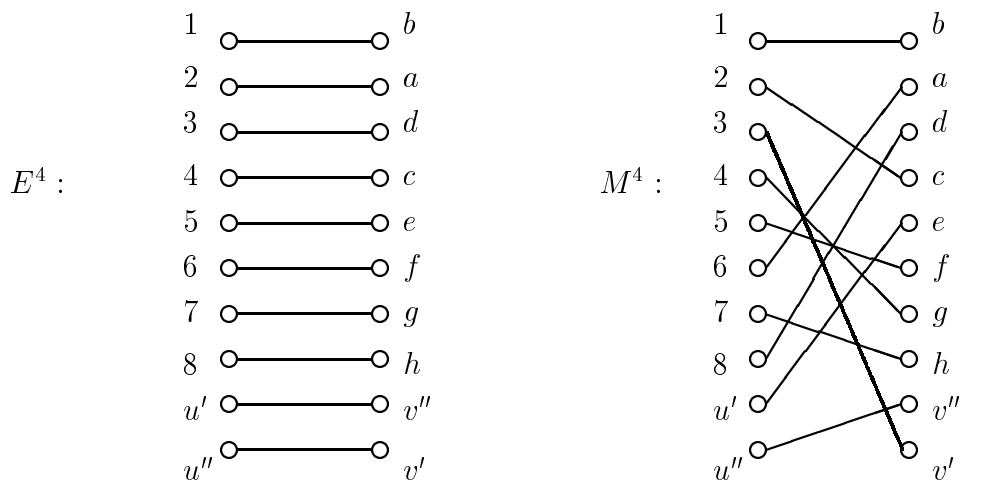


Nach Beweis von Lemma 4.8 sehen E^4 und M^4 wie folgt aus:

$$E^4 := \{\{1, b\}, \{2, a\}, \{3, d\}, \{4, c\}, \{5, e\}, \{6, f\}, \{7, g\}, \{8, h\}, \{u', v''\}, \{u'', v'\}\}$$

und

$$M^3 := \{\{1, b\}, \{2, c\}, \{3, v'\}, \{4, g\}, \{5, f\}, \{6, a\}, \{7, h\}, \{8, d\}, \{u'e\}, \{u'', v''\}\} :$$



Für f^4 gilt ebenso: Für alle $v \in V(G_A^4)$ ist $f^4(v) := \{f_{E^4}(v), f_{M^4}(v)\}$. Auch enthält der durch $E^4 \cup M^4$ repräsentierte Graph außer $[2, a, 6, f, 5, e, u', v'', u'', v', 3, d, 8, h, 7, g, 4, c, 2]$ keinen Kreis.

$R^{4'} := (G_A^4, G_Z^4, f_{E^4}, 2)$ ist genau wie $R^{3'}$ mit 2 Ausstapelungen lösbar: Die Elemente 1 und 3 des Ausgangsstapels G_A^4 müssen zur Lösung ausgestapelt werden. Um das Problem $R^{4''} := (G_A^4, G_Z^4, f_{M^4}, 3)$ zu lösen, sind nun drei Ausstapelungen notwendig: Es müssen die Elemente 5, 7 und u'' des Ausgangsstapels auf Hilfsstapelplätze gelegt werden. Damit ist $R' := R^{4'}$ die in Lemma 4.8 beschriebene Erweiterung von R , $E' := E^4$ und $M' := M^4$ sind die entsprechenden perfekten Matchings.

Aus der Kompliziertheitstheorie (siehe GAREY and JOHNSON [1]) ist folgende Äquivalenz bekannt:

Lemma 4.9 Ein Problem Π ist NP-schwer, wenn Π' ein NP-schweres Problem ist und aus $\Pi \in P \implies \Pi' \in P$ folgt.

Satz 4.10 *Es sei $R := (G_A, G_Z, f, k)$ ein lösbares nicht eineindeutiges verzweigtes Stapelproblem. Die Bestimmung eines perfekten Matchings $M \subseteq A(G(f))$ für $G(f)$, für das die Anzahl notwendiger Ausstapelungen zur Lösung von $R' := (G_A, G_Z, f_M, k)$ minimal ist, ist NP-schwer.*

Bevor wir mit dem eigentlichen Beweis des Satzes 4.10 beginnen, soll kurz die Beweisidee erläutert werden: Wir wollen zeigen, daß für ein nicht eineindeutiges Stapelproblem R die Berechnung einer eineindeutigen Zuordnung der Elemente des Ausgangsstapels zu den Positionen im Zielstapel, so daß die Anzahl der zur Lösung dieses Stapelproblems R notwendigen Hilfsstapelplätze minimal ist, NP-schwer ist.

Dazu nehmen wir an, es gäbe einen Algorithmus PA , der eine solche Zuordnung (perfektes Matching) in polynomialer Zeit berechnet. Wir zeigen dann, daß wir mit Hilfe eines solchen Algorithmus PA die Lösbarkeit eines eineindeutigen Stapelproblems Q mit k Hilfsstapelplätzen in Polynomialzeit bestimmen könnten. Da jedoch die Entscheidung der Lösbarkeit von Q nach Satz 3.5 NP-vollständig ist, ist damit nach Lemma 4.9 der Satz bewiesen.

Wie kann nun mit Hilfe von PA die Lösbarkeit von Q in Polynomialzeit bestimmt werden? Dazu verwandeln wir Q zunächst in ein formal nicht eineindeutiges Stapelproblem R , dessen Zuordnungsfunktion genau der (implizit definierten) Zuordnung der Elemente in Q entspricht, so daß R genau dann lösbar ist, wenn Q lösbar ist. Mit Hilfe von Lemma 4.8 können wir eine Erweiterung R' von R berechnen, deren Zuordnungsfunktion maximal zwei verschiedene eineindeutige Zuordnungen E und M zuläßt. Für die sich durch diese eineindeutige Zuordnungen ergebenden eineindeutigen Stapelprobleme R'' und R''' gilt: R'' ist genau dann lösbar, wenn Q lösbar ist, R''' ist mit genau $2k + 1$ Ausstapelungen lösbar.

Wenn wir nun den (Polynomialzeit-)Algorithmus PA auf R' anwenden, erhalten wir als Ergebnis entweder die Zuordnung E oder M . Lautet das Ergebnis E , so ist R' und damit Q lösbar. Ist das Ergebnis von PA jedoch M , dann heißt das, daß zur Lösung von R' mindestens $2k + 1$ Ausstapelungen notwendig sind und R' und damit Q nicht lösbar sind.

Beweis von Satz 4.10: Den Satz 4.10 beweisen wir mit Hilfe von Lemma 4.9. Dazu wählen wir als Problem Π' die Bestimmung der Lösbarkeit eines eineindeutigen, verzweigten Stapelproblems $Q = (G_A, G_Z, k)$. Nach Satz 3.5 ist dies ein NP-schweres Problem. Uns bleibt nun zu zeigen, daß, wenn wir Π (die Fragestellung des Satzes 4.10) in polynomialer Zeit lösen könnten, Π' ebenfalls polynomial lösbar wäre. Dazu nehmen wir nun an, es gäbe einen Polynomialzeitalgorithmus PA , der das in Satz 4.10 geforderte perfekte Matching M berechnet. Es sei $Q = (G_A, G_Z, k)$ ein eineindeutiges, verzweigtes Stapelproblem. Dann sei

$R^1 := (G_A, G_Z, f_I, k)$, wobei für alle $v \in V(G_A)$ $f_I(v) := \{v\}$ gelte. Dann gilt offensichtlich

$$R^1 := (G_A, G_Z, f_I, k) \text{ ist lösbar} \iff Q = (G_A, G_Z, k) \text{ ist lösbar,} \quad (4.2)$$

weil R^1 nur formal ein nichteindeutiges Stapelproblem ist: Die Zuordnungsfunktion f_I läßt ja nur genau eine Zuordnung der Elemente des Ausgangsstapels zu den Positionen im Zielstapel zu, die dem Problem Q analog ist. Wir beschaffen uns nun eine Erweiterung $R^{1'} := (G'_A, G'_Z, f', 2k)$ von R^1 im Sinne von Lemma 4.8, so daß $G(f')$ maximal zwei verschiedene perfekte Matchings M und E enthält und

$$R^{1''} := (G'_A, G'_Z, f_E, 2k) \text{ genau dann lösbar ist, wenn } R^1 \text{ lösbar ist und} \quad (4.3)$$

$R^{1'''} := (G'_A, G'_Z, f_M, 2k + 1)$ lösbar ist – wobei k minimal ist.

Nun wenden wir den Algorithmus PA auf $R^{1'}$ an. Dieser Algorithmus liefert uns nach der Annahme in polynomial vielen Schritten dasjenige Matching von $G(f')$ für das das Problem $R^{1'}$ mit minimaler Ausstapelzahl lösbar ist. Da $G(f')$ außer E und M keine weiteren Matchings enthält, liefert PA als Lösung entweder E oder M . Liefert PA als Lösung E , dann sind mit dem Matching E weniger als $2k + 1$ Ausstapelungen zur Lösung von $R^{1'}$ nötig. (Die Anzahl der notwendigen Ausstapelungen kann nach Konstruktion von $R^{1'}$ im Beweis von Lemma 4.8 nur geradzahlig sein, da das ursprüngliche Problem R^1 „verdoppelt“ wurde.) Damit ist $R^{1''}$ lösbar. Nach (4.3) ist somit auch R^1 lösbar und damit wegen (4.2) auch Q . Lautet das Ergebnis von PA jedoch M , so sind zur Lösung von $R^{1'}$ $2k + 1$ Ausstapelungen notwendig, damit ist $R^{1''}$ nicht lösbar. Somit ist Q nach (4.3) und (4.2) ebenfalls nicht lösbar. Nach Lemma 4.8 ist $R^{1'}$ in Polynomialzeit berechenbar, nach der Annahme soll PA ebenfalls in polynomialer Zeit arbeiten. Somit konnte die Lösbarkeit des Stapelproblems Q in Polynomialzeit entschieden werden.

Mit Lemma 4.9 ist Π damit NP-schwer. \square

Die Aussage in Satz 4.10 stellt kein überraschendes Ergebnis dar, rechtfertigt jedoch die Absicht, ein solches Matching M approximativ zu finden, bzw. die Anzahl notwendiger Ausstapelungen für ein gegebenes, nicht eindeutiges Stapelproblem $R = (G_A, G_Z, f, k)$ näherungsweise zu bestimmen.

Kapitel 5

Ausblick

Die in den Kapiteln 1 bis 4 unter dem Begriff Stapelprobleme untersuchten Fragestellungen besitzen aus rein mathematischer Sicht gesehen recht unterschiedliche Charaktere. Dies zeigt sich bereits bei der Frage nach dem „Unbekannten“ für die einzelnen Problemstellungen: Beim klassischen Turm von Hanoi Problem interessierte uns die Anzahl der notwendigen Elementbewegungen und wir untersuchten die Struktur des Regelgraphen.

Für einfache Stapelprobleme lies sich die genaue Struktur von allen Stapeln bestimmen, die mit zwei Hilfsstapeln lösbar sind. Bei verzweigten Stapelproblemen ist bereits die Bestimmung der zur Lösung notwendigen Hilfsstapelplätze für gegebene Stapel NP-schwer.

In den vorangegangenen Kapiteln konnte nachgewiesen werden, daß Stapelprobleme – in für praktische Anwendungen relevanter Größe – algorithmisch gut handhabbar sind. Selbst für NP-schwere verzweigte Stapelprobleme ist für konkrete Beispiele oftmals die exakte Lösung in vernünftiger Zeit bestimmbar – außerdem liefert das vorgestellte heuristische Verfahren akzeptable Ergebnisse.

Für verzweigte Stapelprobleme, bei denen die Zuordnung von Elementen des Ausgangsstapels zu Positionen im Zielstapel nicht eindeutig ist, bleibt offen, ob sich ein Näherungsverfahren finden läßt, das die (NP-schwere) Bestimmung eines optimalen Matchings in vertretbarer Qualität löst.

Mit der vorliegenden Dissertation wurde erstmalig ein Überblick zur Thematik Stapelprobleme geschaffen. Allein die Anzahl variierbarer Parameter (z.B. die Zahl der Hilfsstapelplätze, die Struktur der Stapel und des Regelsystems) macht deutlich, daß dieser Überblick keineswegs vollständig sein kann. Die Untersuchungen für die hier betrachteten Problembereiche können insbesondere wie folgt vertieft werden:

Für einfache Stapelprobleme ist die interessanteste Fragestellung sicherlich die nach den Permutationen für die das Problem mit einer vorgegebenen Anzahl an Hilfsstapelplätzen lösbar ist.

Für verzweigte Stapelprobleme stellt sich die Frage nach einem einfachen Kriterium, mit dessen Hilfe entschieden werden kann, ob für gegebene Ausgangs-

und Zielstapel die Anzahl notwendiger Hilfsstapelplätze bzw. obere und untere Schranken in Polynomialzeit berechnet werden können oder nicht. Gleichzeitig ist es sinnvoll, mit Hilfe von weiteren Tests das Verhalten des heuristischen Verfahrens in Abhängigkeit von den bereits beschriebenen und weiteren Parametern zu untersuchen. Hierbei ist es sicher interessant, bereits bekannte Ergebnisse mit denen zu vergleichen, die sich ergeben, wenn andere Beispielgeneratoren benutzt werden.

Von besonderem Interesse sind die Stapelproblem mit nicht eindeutiger Elementezuordnung. Zur Bestimmung der zur Lösung notwendigen Anzahl an Hilfsstapelplätzen können hier zwei verschiedene Vorgehensweisen gewählt werden: Einerseits kann ein perfektes Matching bestimmt werden und danach das eindeutige Stapelproblem gelöst werden. Andererseits ist es denkbar, bereits bei der Auswahl der entsprechenden Matchingkanten auf eventuell notwendige Ausstapelungen Rücksicht zu nehmen und gegebenenfalls dabei die Anzahl der zur Lösung notwendigen Hilfsstapelplätze mitzubestimmen. Zum gegenwärtigen Zeitpunkt liegen jedoch keine Erkenntnisse über mögliche Vor- oder Nachteile des einen Verfahrens gegenüber dem anderen vor.

Es bleibt zu hoffen, daß diese Arbeit anderen Hilfe und Anregung bei der Bearbeitung dieses oder eines ähnlichen Themas ist.

Literaturverzeichnis

- [1] M.R. GAREY and D.S. JOHNSON: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco (1979)
- [2] M.C. GOLUMBIC: *Algorithmic graph theory and perfect graphs*. Academic Press New York (1980)
- [3] A.M. HINZ: *The Tower of Hanoi*. L'Enseignement Mathématique (2), **35** (1989), 289-321 Academic Press New York (1980)
- [4] J. HOPCROFT AND R.M. KARP: *An $n^{\frac{5}{2}}$ Algorithm for Maximum Matching in Bipartite Graphs*. SIAM J. Comp. **2** (1973), 225-231
- [5] D. JUNGNIKKEL: *Graphen, Netzwerke und Algorithmen*. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich 1990
- [6] L. KÄMMERER AND L. HEMPEL: *The Towers of Hanoi and Pileproblems*, in Operations Research Proceedings 1996; Springer-Verlag Berlin Heidelberg New York 1997
- [7] L. KÄMMERER AND L. HEMPEL: *Eine Heuristik zur Lösung von Stapelproblemen*. Beitrag zum IKM, Weimar, 1997
- [8] L. KÄMMERER AND L. HEMPEL: *A Heuristic Approach to Solve a Special Pileproblem*. Beitrag zum Symposium on Operations Research, Jena, 1997
- [9] A. A. K. MAJUMDAR: *The generalized p-peg Tower of Hanoi Problem*. Optimization (2), **32** (1995), 175-183
- [10] C.S. KLEIN AND S. MINSKER: *The super towers of Hanoi problem: large rings on small rings*. Discrete Mathematics (1-3) **114** (1993), 283-295
- [11] P. L'ECUYER: *Efficient and Portable Combined Atom Number Generators*. Communications of the ACM (v31 #6) (1988)
- [12] S. MINSKER: *The Towers of Antwerpen problem*. Information Processing Letters (2), **38**, (1991), 107-111
- [13] R. SCHMIEDEL: *persönliche Mitteilung*.
- [14] S. SCHULZ: *Algorithmen zur Auswahl eines günstigen Palettensortiments – ein Beitrag zur automatengestützten Palettisierung* Diplomarbeit, HAB Weimar, 1989
- [15] D. VASILKOV AND L. HEMPEL: *On the Computational Aspects of the Pile Problem*. Beitrag zum Symposium on Operations Research, Jena, 1997

- [16] H. WALTHER, G. NÄGLER: *Graphen Algorithmen Programme*. Fachbuchverlag Leipzig, 1987
- [17] J.-S. WU AND R.-J. CHEN: *The towers of hanoi problem with cyclic parallel moves*. Information Processing Letters (1), **46** (1993), 1-6
- [18] A.J. VAN ZANTEN: *An Iterative Optimal Algorithm for the Generalized Tower of Hanoi Problem*. Intern. J. Comput. Math. (3/4) **39** (1991), 163-168

Eidesstattliche Erklärung

Ich erkläre hiermit in Kenntniss der strafbaren Folgen einer eidesstattlichen Falschaussage an Eides Statt, daß ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten, Methoden und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Weimar, den 10. 09. 1997

Danksagung

An dieser Stelle danke ich Herrn Prof. Lorenz Hempel für die Betreuung meiner Arbeit und für die freundschaftliche Beratung und Unterstützung. Weiterhin danke ich Herrn Dr. Roland Schmiedel und Herrn Dr. Dmitri Vasilkov für ihre Ideen und Anregungen. Schließlich gilt mein Dank allen Kollegen der Bereiche Optimierung und Analysis für die gute Zusammenarbeit.

Lebenslauf

- Persönliche Daten: Lutz Kämmerer
Parkweg 2a
07639 Bad Klosterlausnitz
geboren am 10. 04. 1970 in Gera, ledig
- Schulbildung:
- 1976 – 1986 Besuch der Polytechnischen
Oberschule in Bad Klosterlausnitz
- 1986 – 1988 Besuch der Erweiterten Oberschule
in Stadtroda, Abitur im Juni 1988
- Hochschulbildung:
- 1988 – 1994 Studium der Mathematik an der
Friedrich-Schiller-Universität Jena
Schwerpunkte: Theoretische Informatik,
Algorithmtheorie, Kompliziertheits-
theorie, Formale Sprachen, Theoretische
Physik,
Diplomarbeit auf dem Gebiet der
Graphentheorie
- 1994 – 1997 Promotionsstudium an der
Bauhaus-Universität Weimar
- Praktikum:
- Feb. 1993 – Mai 1993 Praktikum bei der Dornier GmbH,
Friedrichshafen